www.ordix.de € 3,20

2|2016

ORDIX® news

einfach. besser. informiert.

Elasticsearch & Co

IT-Sicherheit für Projektmanager und IT-Leiter

Begegnen Sie neuen Bedrohungen und Sicherheitsanforderungen erfolgreich. Unser neues Seminar richtet sich an Projektleiter in großen Unternehmen, die IT-Sicherheitsmaßnahmen und Projekte planen und entsprechend der Kritikalität des Projektes verstehen und einschätzen müssen. Es richtet sich zudem an IT-Leiter aus kleinen und mittleren Unternehmen (KMU), die für ihr Unternehmen IT-Sicherheitsmaßnahmen prüfen und festlegen müssen.

Konditionen

Seminar-ID: IT-SEC-01
Dauer: 3 Tage

Preis pro Teilnehmer:
1.590,00 € (zzgl. MwSt.)

Frühbucherpreis:
1.431,00 € (zzgl. MwSt.)

Ziele und Nutzen des Seminars

- Sie kennen die Anforderungen des BSI Grundschutz und der ISO 27001 und können Sicherheitsrisiken identifizieren – sei es in Webanwendungen, in der Netzwerkkommunikation oder im Zugangs- und Berechtigungsmanagement, um nur einige Beispiele zu nennen.
- Sie wisssen mit welchen praxiserprobten Methoden und Arbeitstechniken eine erfolgreiche Einschätzung und Bewertung der Sicherheit Ihrer IT-Systeme möglich ist.

Weitere Informationen finden Sie unter: **seminare.ordix.de**

Nina Komo * 1996 - † 2016

Paderborn, Dezember 2016

Viel Stoff

Vor ein paar Tagen wurde ich von einem Kunden darauf angesprochen, dass es doch zurzeit eine Situation nach der anderen gibt, die für ein brauchbares Thema zu meinem Editorial reicht. Oh ja, er hatte leider recht. Viel Stoff über den ich schreiben kann, beängstigend viel.

Nehmen wir die jüngste Panne bei der Telekom: Egal ob jetzt Hacker-Angriff oder nicht, schlimm empfand ich es nicht. Nur seltsam, dass man versuchte, das erst mal totzuschweigen. Ganz putzig, wie unsere Mutti sich dem Problem in einer ihrer Reden am Tag danach widmete, da bekam ich so richtig Vertrauen in die Sicherheit in Deutschland.

Oder die neue Steigerungsform von Vollpfosten: AfD-Mitglied, Trump, Erdogan. Die Reihenfolge ist beliebig änderbar und trifft leider immer zu. Ach, bin ich froh, dass ich keine Kinder habe, denn was unseren Nachkommen neben keiner Rentenlösung sonst noch so droht, ist unbeschreiblich.

Europa und viele Teile der Welt leben nun seit mehr als siebzig Jahren in Frieden. Aber genau das scheint sehr viele Menschen mit Entscheidungsgewalt zu beunruhigen. Anders kann ich mir immer weiter gehende Rüstungsanstrengungen (u. a. EU-Verteidigungsfond), Hasstiraden (US-Wahlkampf), Rechtsruck in vielen europäischen Staaten, das Bestreben, die Bundeswehr in immer mehr Krisengebiete zu schicken, nicht erklären. Die weltweiten Rüstungsausgaben pro Jahr sind in den letzten 16 Jahren um mehr als 70 % gestiegen[1)] und sind damit ca. sechsmal höher als der deutsche Bundeshaushalt von 2015.

Dabei sollte doch inzwischen allen klar sein, dass trotz markiger Sprüche [2)] (Peter Struck) weder in Afghanistan, noch im Irak oder jetzt in Syrien irgendwas für Freiheit oder Unabhängigkeit erreicht wurde. Resultat dieser Konflikte ist eher das Gegenteil. Die Situation wird instabiler und Personen, wie der neue amerikanische Präsident, Erdogan oder einige rechte europäische Politiker, gießen mit ihren Parolen und Aktivitäten nur weiterhin Öl ins Feuer.

Überall scheinen Verbrecher unterschiedlichster Art weiterzukommen als normale Menschen. Und den Medien scheint es zu gefallen. Jüngstes Beispiel ist die Inszenierung der erneuten „Inthronisation“ des Uli Hoeneß in München unter den Augen und mit der ausdrücklichen Billigung der bayerischen Politik.

Da bleibt mir nur der Wunsch zu Weihnachten: „Liebes Fernsehen, zeige doch noch mal die Szene, wo der Uli Hoeneß über den Christoph Daum sagt, dass Kriminelle im Fußball nichts zu suchen haben.“

Ihnen wünsche ich, dass Sie nicht nur diese Zeilen lesen, sondern den einen oder anderen unserer Artikel zu Themen wie IT-Sicherheit (kein Hinweis auf die Telekom 😉), Big Data oder Storage interessant finden. Diese Themen zeigen im Übrigen ganz intensiv die Weiterentwicklung der ORDIX-Mannschaft jenseits von Datenbanken, Java, Application-Servern oder Betriebssystemen, obwohl Sie natürlich auch dazu wieder aktuell informiert werden.

Weiterentwicklung auch unseren Nachwuchs betreffend: Seit mittlerweile 20 Jahren fördern wir junge Menschen, die ihr duales Studium bei uns absolvieren. Inzwischen begleiten wir fast 30 Studenten auf diesem Weg, alle mit dem Ziel einer Festanstellung nach ihrem Studium.

Einer jungen Studentin widme ich diese Zeilen. Leider ist Nina Komo nur wenige Monate bei uns geblieben und ganz plötzlich verstorben. Möge sie hoffentlich jetzt in einer besseren Welt weilen.

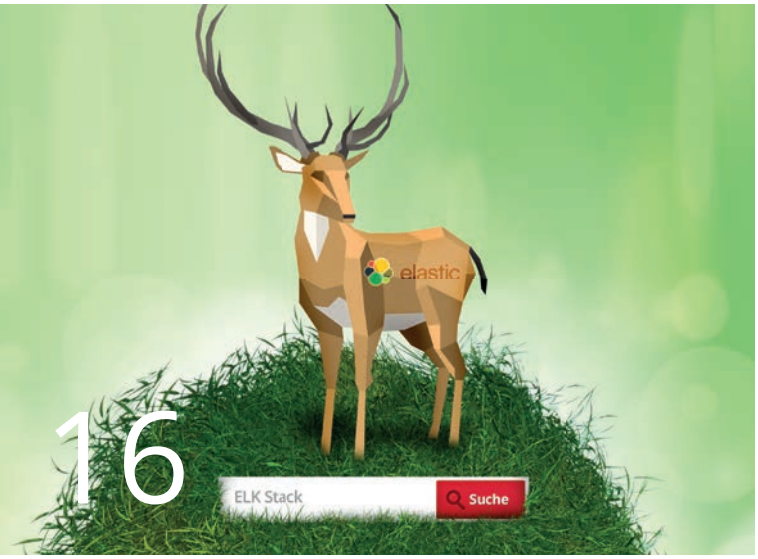
Ich wünsche Ihnen geruhsame Feiertage und einen guten Start in ein hoffentlich - trotz vieler Wahlen - besseres 2017.

Ihr

Wolfgang Kögler

[1)] Quelle: de.statista.com SIPRI

[2)] „Die Freiheit wird am Hindukusch verteidigt“



16

ELK-Stack: Volltextsuche und Logarchivierung

22

Moderne Webanwendungen mit JSF & PrimeFaces

IT-Security

Big Data

Web und Application-Server

Oracle

Storage

Betriebssysteme

IT-Security – Sichere Webanwendungen

Flashback – Reise in die Vergangenheit

Entwicklung

Microsoft

Aktuell

Impressum

Herausgeber:	ORDIX AG Aktiengesellschaft für Softwareentwicklung, Beratung, Schulung und Systemintegration, Paderborn
Redaktion/Layout:	Sascia Brinkmann, Jens Pothmann
V.i.S.d.P.:	Christoph Lafeld, Wolfgang Kögler
Anschrift der Redaktion:	ORDIX AG | Karl-Schurz-Straße 19a | 33100 Paderborn Tel.: 05251 1063-0 | Fax: 0180 1673490
Auflage:	7.000 Exemplare
Druck:	Druckerei Bösmann, Detmold
Bildnachweis:	© istockphoto.com | Ekaterina_Vichenko | Papier deer © freepik.com | Kjpargeter | Grassy Globe cloud © tempees.com | searchbar © freepik.com | Onlyyouqj | Hand-pressing-security... © istockphoto.com | Varijanta | Web-design... © commons.wikimedia.org | Oto Godfrey, Justin Morton | TeamTimeCar
Autoren:	Dr. Hubert Austermeier, Marius Dorlöchter, Winfried Gerhard, Klaus Grote, Christopher Herclik, Carsten Hummel, Andreas Jordan, Wolfgang Kögler, Philipp Loer, Sebastian Schäfers, Michael Skowasch, Dr. Dominik Stingl, Michael Thieme, Christian Wiesing
Copyright:	Alle Eigentums- und Nachdruckrechte, auch die der Übersetzung, der Vervielfältigung der Artikel oder von Teilen daraus, sind nur mit schriftlicher Zustimmung der ORDIX AG gestattet.
Warenzeichen:	Einige der aufgeführten Bezeichnungen sind eingetragene Warenzeichen ihrer jeweiligen Inhaber. ORDIX® ist eine registrierte Marke der ORDIX AG.
Haftung:	Eine Haftung für die Richtigkeit der Veröffentlichungen kann trotz sorgfältiger Prüfung durch die Redaktion vom Herausgeber nicht übernommen werden.

Sie können die Zusendung der ORDIX® news jederzeit ohne Angabe von Gründen schriftlich (z. B. Brief, Fax, E-Mail) abbestellen.

Big Data - Informationen neu gelebt (Teil V)

Daten mit Format – Lösungen zur Datenablage in Hadoop

Durch das verteilte Dateisystem HDFS ermöglicht Hadoop die zuverlässige Ablage großer Datenmengen sowie die effiziente Abfrage der Daten durch paralleles Auslesen. Damit Hadoop selbst und andere Applikationen aus dem Big-Data-Umfeld von diesen Vorteilen profitieren können, müssen die verwendeten Dateiformate bestimmte Anforderungen erfüllen. Im Rahmen des Artikels wird anhand vier prominenter Dateiformate beleuchtet, wie diese Anforderungen auf unterschiedliche Art und Weise umgesetzt werden und welche Vor- und Nachteile die jeweilige Lösung mit sich bringt.

Hadoops Anforderungen

Im Big-Data-Umfeld hat sich Apache Hadoop in den vergangenen Jahren als eine zentrale Komponente etabliert. Einer der Gründe für die hohe Bedeutung ist das in Hadoop enthaltene verteilte Dateisystem HDFS (Hadoop Distributed Filesystem) zur Ablage und Nutzung großer Datenmengen innerhalb eines Clusters von Servern (Knoten). Durch die Aufteilung der zu speichernden Daten in HDFS-Blöcke sowie deren Verteilung und Replikation im Cluster wird sowohl eine hohe Verfügbarkeit als auch ein effizienter Zugriff auf die Daten ermöglicht (siehe ORDIX® news Artikel 3/2015 [3]). Andere Applikationen aus dem Big-Data-Umfeld, die Daten zwecks Weiterverarbeitung aus dem HDFS auslesen oder es als Ablage für ihre erzeugten Daten verwenden, profitieren von der verteilten und redundanten Speicherung in zweierlei Hinsicht:

- Durch die Aufteilung der Daten in HDFS-Blöcke können diese parallel verarbeitet werden, wodurch eine schnellere Verarbeitung selbst größerer Datenmengen ermöglicht wird. Zusätzlich kann die resultierende Last auf mehrere Knoten innerhalb des Clusters verteilt werden.
- Tasks und Jobs einer Applikation können im Cluster auf die Knoten verteilt werden, auf denen die notwendigen Daten liegen. Durch diese standortbewusste (engl. locality-aware) Datenverarbeitung wird weistestgehend vermieden, dass größere Mengen an Daten über das Netzwerk ausgetauscht werden müssen.

Da Hadoop, analog zu einem nativen Dateisystem, keine Vorgaben hinsichtlich des Dateiformats macht, bleibt es dem Nutzer überlassen, welche Daten er in welchem Format speichert. Gerade bei der Speicherung von zu verarbeitenden Daten ist es daher unerlässlich, dass diese in einem Format abgelegt werden, welches eine Aufteilung der Daten für die parallele und lokale Verarbeitung ermöglicht.

Neben der Aufteilung der Daten spielt auch deren Komprimierung eine wichtige Rolle. Durch die Komprimierung der Daten wird der Datenaustausch über das Netzwerk innerhalb eines Clusters minimiert und die Dauer lesender und schreibender Festplattenzugriffe reduziert. Gerade bei großen Datenmengen erweisen sich diese Operationen als äußerst zeitintensiv [Q1]. Als positiver Nebeneffekt benötigen die Daten zusätzlich weniger Speicherplatz im Cluster.

Anhand der skizzierten Anforderungen sollte ein ideales Dateiformat daher eine Komprimierung der Daten und deren Aufteilung in mehrere HDFS-Blöcke ermöglichen. Da ein Großteil der von Hadoop unterstützten Komprimierungsalgorithmen keine Aufteilung der Daten zur verteilten Speicherung erlaubt bzw. nur unter erheblichem Aufwand ermöglicht [Q2], wurden neue Dateiformate entwickelt, um beiden Anforderungen Rechnung zu tragen. Im Folgenden wird daher ein Überblick über prominente Dateiformate im Big-Data-Umfeld mit dem Fokus auf Hadoop gegeben. Als Grundlage werden zunächst einfache Textdateien beschrieben. Darauf aufbauend werden SequenceFile-Dateien [Q3] als dedizierte Lösung für den Einsatz in Hadoop betrachtet. Abschließend werden mit Avro [Q4] und Parquet [Q5] stellvertretend zwei neuere Formate vorgestellt, welche die bisher betrachteten Ansätze beispielsweise hinsichtlich Einsatzmöglichkeiten und bereitgestellter Funktionalität übertreffen.

Textdateien

Textdateien stellen eine einfache Möglichkeit dar, um Daten in Hadoop abzulegen. Neben unstrukturierten und semistrukturierten Daten werden Textdateien häufig dazu verwendet, strukturierte Daten in Hadoop zu speichern. Dabei wird ein Datensatz innerhalb der Textdatei üblicherweise durch eine Zeile repräsentiert. Die Daten der vorhandenen

Spalten innerhalb einer Zeile werden wiederum durch Trennzeichen identifiziert, z. B. durch Kommas bei CSV-Dateien. Aufgrund der beschriebenen Anordnung der Daten spricht man bei Textdateien auch von einem zeilenorientierten Datenformat.

Ein großer Vorteil gerade von flachen Textdateien, bei denen auf eine Verschachtelung der Daten wie z. B. in JSON- oder XML-Dateien verzichtet wird, resultiert aus der Simplizität sowie dem einfachen Lesen und Schreiben von Daten. Aufgrund dessen können Textdateien in mehrere HDFS-Blöcke aufgeteilt und im Cluster für eine parallele Verarbeitung verteilt werden.

Die großen Nachteile des Formats treten jedoch bei der Kodierung und Komprimierung der Daten innerhalb einer Textdatei auf. Hinsichtlich der Kodierung müssen die als Text gespeicherten Daten zur weiteren Verarbeitung immer in ihre eigentlichen Datentypen konvertiert werden. Ebenso verbraucht die Darstellung als Text gerade bei größeren Zahlen mehr Speicherplatz als bei einer binären Darstellung. Für eine Komprimierung der Textdateien muss ein Verfahren gewählt werden, welches das Lesen der komprimierten Daten auch noch nach deren Unterteilung in HDFS-Blöcke ermöglicht. Als ein mögliches Komprimierungsverfahren bietet sich hier vor allem bzip2 an. Jedoch wird die Funktionalität des Lesens komprimierter und geteilter Daten durch eine rechenaufwendige und langsame Komprimierung sowie Dekomprimierung erkauft [Q1].

SequenceFile

SequenceFile [Q3] ist ein Dateiformat, welches speziell für die Nutzung mit MapReduce entwickelt wurde, um den oben beschriebenen Nachteilen von Textdateien entgegenzuwirken. Als eines der ältesten binären Dateiformate für Hadoop [Q2] erlauben SequenceFile-Dateien, die enthaltenen Daten zu komprimieren und gleichzeitig in mehrere HDFS-Blöcke aufzuteilen.

Innerhalb einer SequenceFile-Datei werden die Daten als binär kodierte Key-Value-Paare abgelegt. Zur Serialisierung der Datentypen und -strukturen wird auf die in Hadoop existierenden Verfahren zurückgegriffen. Neben der Speicherung von Daten als Key-Value-Paare kann eine SequenceFile-Datei auch als Container für kleinere Dateien (z. B. Textdateien) dienen. Diese Funktionalität ist für das Zusammenfassen mehrerer kleinerer Dateien zu einer größeren sinnvoll, da Hadoop vor allem für die Arbeit mit einer geringeren Anzahl an größeren Dateien ausgelegt wurde [Q2].

Eine SequenceFile-Datei selbst besteht aus einem Header und den eigentlichen Daten, die entweder als Records aus einzelnen Key-Value-Paaren oder als Blöcke bestehend aus mehreren Key-Value-Paaren gespeichert werden (siehe Abbildung 1). Um eine Mehrdeutigkeit des Begriffs Blocks zu vermeiden, wird ein Block im Kontext des betrachteten Dateiformats als Block bezeichnet. Für einen Block im HDFS wird weiterhin der Begriff HDFS-Block verwendet. Welches Verfahren zur Ablage der Key-Value-Paare innerhalb der Datei gewählt wird, richtet sich nach dem verwendeten Komprimierungsverfahren. Für den Fall, dass die Daten komprimiert werden sollen, kann zwischen der separaten Komprimierung eines jeden Records oder der Komprimierung eines kompletten Blocks gewählt werden.

Wie in Abbildung 1 dargestellt, werden zwischen einer gegebenen Anzahl von Records oder zwei Blöcken zusätzlich Sync Marker gesetzt. Durch die Verwendung von Sync Markern können die Daten zwischen zwei Sync Markern (entweder mehrere Records oder ein Block) unabhängig von den restlichen Records und Blöcken komprimiert werden, da Anfang und Ende der Daten gekennzeichnet sind. Parallel dazu wird durch diese Markierung eine Aufteilung einer SequenceFile-Datei in mehrere HDFS-Blöcke ermöglicht, da nun nicht mehr die gesamte Datei, sondern nur noch Abschnitte komprimiert werden.

Wie bereits erwähnt, wurden SequenceFile-Dateien mit einem Fokus auf MapReduce-Jobs entwickelt. Ebenso wird für die Serialisierung der Daten ausschließlich auf die in Hadoop vorhandenen Verfahren zurückgegriffen, wodurch der Einsatz vor allem auf MapReduce-Jobs sowie Hadoop beschränkt ist. Vor diesem Hintergrund wurde mit Avro ein unabhängiges Dateiformat entwickelt, welches auch außerhalb von Hadoop eingesetzt werden kann und darüber hinaus den Zugriff aus unterschiedlichen Programmiersprachen ermöglicht.

Abb. 1: SequenceFile-Datei mit Blöcken von Key-Value-Paaren

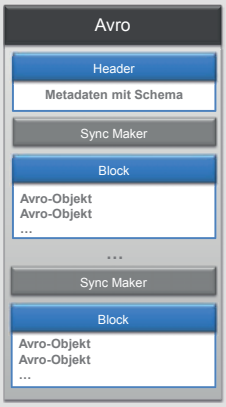


Abb. 2: Avro-Datei

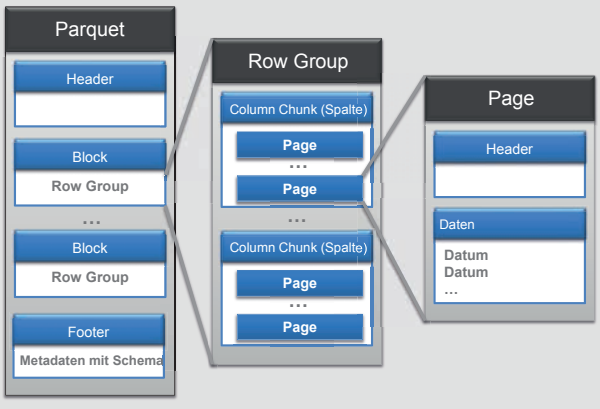


Abb. 3: Parquet-Datei

Avro

Apache Avro [Q4] ist ein Serialisierungssystem, welches auf einem sprachenunabhängigen Schema basiert. Dieses Schema wird dazu verwendet, Datentypen und -strukturen zu definieren, welche im Rahmen der Serialisierung und Deserialisierung in Byte-Streams übersetzt werden und umgekehrt. Durch die sprachenunabhängige Definition des Schemas kann diese Übersetzung in einer beliebigen Programmiersprache geschehen. Serialisierung und Deserialisierung spielen einerseits bei der Interprozesskommunikation im Rahmen sogenannter Remote Procedure Calls (RPCs), aber auch in verteilten Systemen bei der Übertragung von Daten über das Netzwerk eine wichtige Rolle. Während Avro für beide Anwendungsfälle eingesetzt werden kann, beschränkt sich dieser Artikel auf den Einsatz von Avro in verteilten Systemen und beschreibt die Nutzung als Dateiformat in Hadoop.

Das Schema von Avro zur Beschreibung der Daten ist üblicherweise in JSON beschrieben. Darüber hinaus gibt es die Möglichkeit, das Schema mit der eigenen Interface Description Language zu definieren. Eine große Stärke des dabei generierten Schemas ist neben der Sprachenneutralität vor allem die Eigenschaft der Schema-Evolution. Das bedeutet, dass nach der initialen Definition des Schemas neue Datenfelder hinzugefügt bzw. obsolete Datenfelder ignoriert werden können. In letzterem Fall spricht man auch von einer Projektion, da die Anzahl der Datenfelder reduziert wird (Projektion eines höher dimensionalen Raums auf einen Raum mit weniger Dimensionen). Damit ermöglicht Avro, dass sich die zu schreibenden und lesenden Daten über den Lauf der Zeit ändern können, während der Code zur Analyse der Daten nicht angepasst werden muss. Innerhalb der Avro-Datei wird das Schema immer im Header der Datei abgelegt (siehe Abbildung 2), wodurch die Datei selbsterklärend ist. Das heißt, dass ohne vorherige Kenntnisse über das Schema die Daten dennoch gelesen und verarbeitet werden können.

Die eigentlichen Daten werden in einer Avro-Datei als Sequenz von Datensätzen, sogenannten Avro-Objekten, gespeichert. Innerhalb eines Avro-Objekts sind die Daten zeilenorientiert angeordnet und binär kodiert. Die Avro-Objekte werden wiederum in einem Block zusammengefasst. Wie in Abbildung 2 dargestellt, besteht eine Avro-Datei in der Regel aus mehreren Blöcken, die durch eindeutige Sync Marker voneinander getrennt werden. Wie bei einer SequenceFile-Datei ist es daher möglich, die Daten innerhalb eines Blocks bei Bedarf zu komprimieren, während durch die Sync Marker sicher gestellt wird, dass eine Avro-Datei mit ihren Blöcken in mehrere HDFS-Blöcke unterteilt werden kann. Durch die Möglichkeit der Komprimierung der Daten bei gleichzeitigem Splitten der Datei ist Avro ebenfalls für den Einsatz in Hadoop geeignet, im Gegensatz zu SequenceFile-Dateien aber nicht darauf beschränkt. Aktuell unterstützt Avro als Komprimierungsverfahren Snappy [Q6] und Deflate [Q7].

Parquet

Apache Parquet [Q5] ist ein Dateiformat, welches sich von den bisher vorgestellten Formaten vor allem durch die Orientierung der zu speichernden Daten unterscheidet. Parquet legt seine Daten in einem spaltenorientierten Format ab, sodass immer ein Teil der Daten einer Spalte zusammen abgespeichert wird. Basierend auf der Annahme, dass sich die Daten innerhalb einer Spalte meistens geringfügig voneinander unterscheiden, kann die effektive Größe der Daten durch geeignete Kodierungsverfahren deutlich verringert werden. Ebenso ist es möglich, bei der Abfrage von Daten irrelevante Spalten zu überspringen, sodass weniger Daten gelesen und verarbeitet werden müssen. Spaltenbasierte Formate eignen sich daher vor allem für Operationen, die auf die Daten einer oder weniger Spalten zugreifen. Benötigen die auszuführenden Operationen hingegen Daten aus mehreren Spalten, kann ein zeilenorientiertes Format bei der Abfrage effizienter sein [Q1].

Parquet ist für ein breites Anwendungsspektrum entwickelt worden und nicht nur auf die Nutzung in Hadoop beschränkt. Dementsprechend werden die Metadaten zur Beschreibung des Dateiformats sowie der vorhandenen Daten wie bei Avro in einem sprachenneutralen Schema definiert. Als spaltenbasiertes Dateiformat ermöglicht Parquet auch die spaltenorientierte Darstellung verschachtelter Datenstrukturen, wie sie z. B. bei der Verwendung und Schachtelung von Maps entsteht. Zur Transformation in eine spaltenorientierte Darstellung greift Parquet auf die Algorithmen von Dremel zurück, welches ein von Google entwickeltes Framework zur Datenabfrage aus verschachtelten Datenstrukturen ist [Q8]. Durch diese Algorithmen können die Daten aller Datenfelder innerhalb einer geschachtelten Datenstruktur in einer eigenen Spalte abgespeichert werden. Somit ist anschließend beispielsweise das getrennte Einlesen von Keys und Values einer Map möglich.

Die eigentliche Parquet-Datei besteht aus einem Header, gefolgt von mehreren Blöcken mit den Daten und einem abschließenden Footer (siehe Abbildung 3). Durch die Speicherung des Schemas innerhalb der Datei sind die Daten wie bei Avro selbsterklärend, jedoch legt Parquet das Schema als Teil der allgemeinen Metadaten im Footer der Datei ab. Durch die Ablage im Footer wird sichergestellt, dass die Metadaten erst geschrieben werden, nachdem alle Daten innerhalb der Blöcke gespeichert wurden. Folglich sind Anfang und Ende eines jeden Blocks bekannt und werden in den Metadaten abgespeichert. Aufgrund dieser Informationen ist es im Gegensatz zu Avro und SequenceFile-Dateien nicht notwendig, Markierungen zwischen die einzelnen Blöcke zur Synchronisation zu schreiben. Da die Position von jedem Block aufgrund der Metadaten ermittelt werden kann, ist es möglich, eine Parquet-Datei auch ohne Markierungen in mehrere HDFS-Blöcke aufzuteilen. Zur Identifikation der einzelnen Blöcke muss daher zunächst immer das Ende einer Parquet-Datei gelesen werden, bevor auf die Daten zugegriffen werden kann.

Ein Block einer Parquet-Datei besteht aus einer einzigen sogenannten Row Group. Entsprechend des Namens umfasst eine Row Group eine variable Anzahl von Zeilen inklusive der darin enthaltenen Daten. Diese Daten sind, wie in Abbildung 3 dargestellt, pro Spalte in einem Column Chunk gruppiert, sodass die Anzahl an Column Chunks mit der Anzahl an Spalten übereinstimmt. Ein Column Chunk ist zusätzlich nochmal in Pages unterteilt, um geschachtelte Datenstrukturen bei Bedarf zu „glätten“ und spalten-

orientiert abzuspeichern (siehe Abbildung 3). Die flachen, spaltenorientierten Daten innerhalb einer Page werden von Parquet in einem zweistufigen Verfahren komprimiert. In einem ersten Schritt wählt Parquet eine geeignete Kodierung zur Repräsentation der Daten (siehe [Q9] für verfügbare Kodierungen). In einem zweiten Schritt können die kodierten Daten zusätzlich komprimiert werden. Als mögliche Verfahren stehen in Parquet Snappy [Q6], gzip [Q10] und LZO [Q11] zur Komprimierung zur Verfügung. Innerhalb eines Column Chunks können dabei pro Page unterschiedliche Kodierungs- und Komprimierungsverfahren verwendet werden.

Fazit

Im Rahmen dieses Artikels wurde neben gewöhnlichen Textdateien anhand drei prominenter Beispiele beleuchtet, wie sich die Dateiformate weiter entwickelt haben, um in Hadoop zum Einsatz zu kommen. Innerhalb der Betrachtung wurde deutlich, dass abgesehen von Textdateien alle drei Dateiformate die zu Beginn des Artikels identifizierten Anforderungen umsetzen. Durch die Teilbarkeit der Daten bei gleichzeitiger Komprimierung unterstützen die Formate die parallele und lokale Verarbeitung und reduzieren die Last auf das Netzwerk und die Festplatten des Clusters. Ebenso zeigt die Betrachtung, dass sich die Formate von dedizierten Lösungen zu generell einsetzbaren Dateiformaten entwickelt haben. Während bspw. SequenceFile-Dateien speziell für MapReduce-Jobs und Hadoop konzipiert wurden, zeichnen sich neuere Formate wie Avro oder Parquet durch ihre vielfältigen Einsatzmöglichkeiten sowie durch eine deutlich höhere Flexibilität und Funktionalität aus. Erwähnt sei hier der sprachenunabhängige Datenzugriff oder die Möglichkeit der Schema-Evolution, die eine Modifikation existierender Schemata über den Lauf der Zeit erlaubt.

Welches von den vorgestellten Dateiformaten letzten Endes das beste Format ist, lässt sich nicht pauschalisieren, da dies unter anderem von der Art der Analyse oder aber auch von der Unterstützung der genutzten Applikationen abhängt. So sollte beispielsweise bei der Wahl des Dateiformats geprüft werden, ob das Dateiformat mit der gewählten Applikation und der zugrundeliegenden Programmiersprache gelesen und geschrieben werden kann. Ebenso ist zu berücksichtigen, ob die geplanten Analysen auf den Daten einzelner oder vieler Spalten operieren. Wie im Abschnitt zu Parquet erwähnt, bieten sich für Operationen auf wenigen Spalten spaltenorientierte Formate an, während zeilenorientiere Formate bei Operation über viele Spalten hinweg vorzuziehen sind.

Dr. Dominik Stingl
(info@ordix.de)

Glossar

HDFS
Das Hadoop Distributed Filesystem (HDFS) ist ein hochverfügbares Dateisystem zur verteilten Speicherung sehr großer Datenmengen innerhalb eines Clusters von Servern (Knoten). Die Dateien werden hierbei in Datenblöcke mit konstanter Länge zerlegt und redundant auf die Knoten des Clusters verteilt, wodurch das HDFS vor allem das Persistieren sehr großer Dateien unterstützt.

Links

[1] ORDIX® news Artikel 1/2015 –
„Big Data – Informationen neu gelebt (Teil I) - Wie big ist Big Data?“:
http://ordix.de/ordix-news-archiv/1-2015.html

[2] ORDIX® news Artikel 2/2015 –
„Big Data – Informationen neu gelebt (Teil II) -Apache Cassandra“:
http://ordix.de/ordix-news-archiv/2-2015.html

[3] ORDIX® news Artikel 3/2015 – „Big Data – Informationen neu gelebt (Teil III) - Apache Hadoop – auf die elefantöse Art“:
http://ordix.de/ordix-news-archiv/3-2015.html

[4] ORDIX® news Artikel 1/2016 –
„Big Data – Informationen neu gelebt (Teil IV) - Apache Spark“:
http://ordix.de/ordix-news-archiv/1-2016.html

[5] Seminarempfehlung:
„Big Data: Informationen neu gelebt“ (DB-BIG-01)
https://seminare.ordix.de/seminare/big-data-und-data-warehouse

[6] Seminarempfehlung:
„Big Data: Apache Hadoop Grundlagen“ (DB-BIG-02)
https://seminare.ordix.de/seminare/big-data-und-data-warehouse

Quellen

[Q1] Mark Grover et al.: „Hadoop Application Architectures”;
1. Auflage; Sebastopol: O’Reilly Media; 2015

[Q2] Tom White: „Hadoop: The Definitive Guide“;
4. Auflage; Sebastopol: O'Reilly Media; 2015

[Q3] http://hadoop.apache.org/docs/current/api/index.html?org/apache/hadoop/io/SequenceFile.html

[Q4] https://avro.apache.org/

[Q5] https://parquet.apache.org/

[Q6] http://google.github.io/snappy/

[Q7]http://www.gzip.org/algorithm.txt

[Q8] Sergey Melnik et al.: „Dremel: Interactive Analysis of Web-Scale Datasets”
In: Proceedings of the VLDB Endowment. 3, 1-2, pp. 330-339

[Q9] https://github.com/Parquet/parquet-format/blob/master/Encodings.md

[Q10] http://www.gzip.org/

[Q11] https://de.wikipedia.org/wiki/Lempel-Ziv-Oberhumer

Entscheidungshilfe für die Suche nach der richtigen App

Die Antwort auf alle Fragen - Hybride Apps mit HTML5, CSS3 und JavaScript

In der Ausgabe 03/2015 haben wir über das Erstellen von Cross-Platform-Apps mithilfe des Oracle Mobile Application Frameworks berichtet. Doch ist dieses Framework das Allheilmittel für alle Zwecke? In diesem Artikel möchten wir noch vor der eigentlichen Entwicklung der App ansetzen und herausfinden, welches Framework das Richtige für Ihren Anwendungsfall ist.

Was sind native, hybride und Web Apps?

„Native Apps“ sind Anwendungen, die für eine spezifische Plattform entwickelt wurden. Dazu zählen zum Beispiel, die Entwicklung von Android Apps mit Java und dem Android SDK sowie das Schreiben von iOS Apps mit Objective-C oder Swift. Am Ende des Entwicklungsprozesses entsteht somit eine Installationsdatei, die anschließend durch den Apple App Store, Google Playstore und Co. verteilt werden kann.

„Web Apps“ hingegen beschränken sich nicht auf eine spezielle Plattform. Es handelt sich in der Regel um eine gewöhnliche Webseite, welche für mobile Geräte optimiert wurde. Diese Website kann nun mithilfe des Browsers eines Smartphones ausgeführt werden. Die Entwicklung erfolgt häufig mit modernster Web-Technologie wie HTML5, CSS3 und JavaScript.

„Hybride Apps“ stellen eine Kreuzung aus nativen und Web Apps dar (siehe Abbildung 1). Die Entwicklung wird regulär mit den genannten Web-Technologien realisiert, da die meisten Zielgeräte moderne Browser von Werk aus mitliefern. Die gängigen Frameworks für die Umsetzung von hybriden Apps, bauen anschließend Installationsdateien für die einzelnen Betriebssysteme, welche dann genau wie die nativen Apps im App Store vertrieben werden können.

Welcher Ansatz ist der richtige?

Viele Wege führen bekanntlich nach Rom. Aber welcher Ansatz eignet sich am besten für Ihre Anforderungen? Häufig stellt sich der richtige App-Typ sehr schnell durch ein paar einfache Fragen heraus. Wichtig ist: Machen Sie sich bereits im Vorfeld Gedanken zu den geplanten Features!

Ein gravierender Unterschied zwischen den App-Typen besteht in der Performance. Während die hybride App noch einigermaßen mit der herausragenden Leistung der nativen App mithalten kann, sollte bei der Web App mit Einbußen gerechnet werden. Dies gilt allerdings nur für anspruchsvolle Anwendungen. Triviale Aufgaben kann die Web App genauso gut erledigen.

Auch in der Offline-Verfügbarkeit muss bei der Web App mit Einschränkungen gerechnet werden. Während native und hybride Apps auf den Speicher des Smartphones zugreifen, ist die Web App in der Regel auf den Browsercache angewiesen und somit sehr eingeschränkt. Durch moderne Technologien wie dem Web Storage können Daten auf dem Client gespeichert und bis zu einem gewissen Grad offline genutzt werden (siehe [1]).

Native Funktionen der Geräte können ebenfalls nur sporadisch durch Web Apps genutzt werden. Um den vollen Funktionsumfang von iPhone und Co. auszunutzen, empfiehlt sich einer der anderen Typen. Beispiele für solche Funktionen sind: Kamera, Kontaktdaten und Benachrichtigungen.

Dass das Aufrufen einer Webseite einfacher ist als die Installation einer App aus dem App Store, ist nicht zu leugnen. Auch das Anlegen eines Lesezeichens mit Symbol auf dem Homescreen erschwert den Einrichtungsvorgang nicht sonderlich. Verwendet wird diese Funktion in der Realität allerdings eher selten (siehe [2]), weshalb die Web App zwar einfacher einzurichten, aber nicht zwangsläufig die bessere Wahl ist.

Auf den ersten Blick wirkt es, als sei die Erreichbarkeit ein weiterer Vorteil von Web Apps. Da Web Apps lediglich für Smartphones optimierte Webseiten sind, können Nutzer diese per Suchmaschine finden. Native und hybride Apps

hingegen sind nur im App Store erhältlich und erfordern eine gezielte Suche des Nutzers. Jedoch gibt es Anwendungsfälle, in denen der Nutzer gezielt nach Apps sucht, um das Aufrufen von Webseiten zu vermeiden; beispielsweise um anfallende Roaming-Gebühren im Ausland zu vermeiden.

Wirklich interessant wird es bei der Gegenüberstellung in Bezug auf die Entwicklungskosten. Wird die App nur für ein Betriebssystem entwickelt, so dürften sich die Kosten für alle Typen in etwa die Waage halten. Wird die Anwendung auf verschiedensten Zielgeräten (iOS, Android, etc.) ausgeführt, kann die native App nicht mehr mithalten. Es müsste dann für jede Zielplattform eine eigene native App geschrieben (und getestet!) werden. Da diesen unterschiedliche Programmiersprachen und SDKs zugrunde liegen, werden deutlich mehr Kenntnisse benötigt. Hier sollten Sie darauf achten, welches Know-how in Ihrem Unternehmen vorhanden ist.

Ein letzter Faktor sind die möglichen Inhalte Ihrer App. Hybride und native Apps werden bei der Veröffentlichung einer mehr (Apple) oder weniger (Google) strengen Kontrolle unterzogen. So wurde beispielsweise die App „I am rich“ nachträglich aus dem App Store entfernt, da sie laut Apple keinen Mehrwert für das iPhone bot [2]. Möchten Sie derartige Probleme umgehen, bietet sich hier die Web App als Mittel der Wahl an.

Da Sie nun die Vor- und Nachteile der jeweiligen App-Typen kennen, liegt es an Ihnen, sich für einen der drei Ansätze zu entscheiden. In Abbildung 2 finden Sie die genannten Kriterien noch einmal zusammengefasst. Grundsätzlich lässt sich sagen, dass die hybride App die Vorzüge der anderen beiden Varianten sehr gekonnt miteinander vereint. Sollten Sie sich nun für eine hybride App entschieden haben, dürfen bzw. müssen Sie sich noch für eines der zahllosen Frameworks entscheiden.

Hybride Apps, die Qual der Wahl

Frameworks für hybride Apps gibt es wie Sand am Meer. Doch welches passt zu den eigenen Anforderungen? Hier erhalten Sie eine Übersicht über einige der bekanntesten Ansätze und wertvolle Tipps zur Entscheidungshilfe.

Im Bereich von Enterprise Apps wird häufig das Kendo UI Framework verwendet. Die fertigen Apps können anschließend neben iOS, Android und WindowsPhone auch auf Blackberry-Geräten installiert werden. Unterstützt werden hier zusätzlich zu den Web-Technologien HTML, CSS und JavaScript auch PHP, Java, C# und zum Teil auch Ruby. Der Internetauftritt des Frameworks vermittelt den Eindruck, dass der Fokus auf der Reduzierung der Entwicklungszeit liegt. Wer die nativen Features des Smartphones nutzen möchte, wird vermutlich etwas enttäuscht sein: Das Framework bietet lediglich die Möglichkeit, auf Gesten und Multitouch zuzugreifen. Das Unternehmen Telerik, welches auf seiner Webseite mit Partnerschaften zu Microsoft und SAP wirbt, bietet sowohl ausreichend Support als auch Trainingskurse an. Der

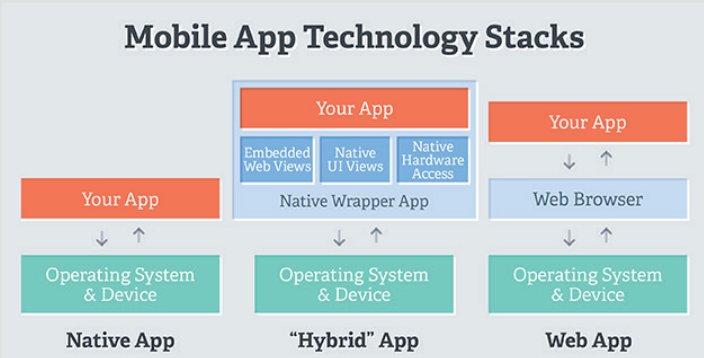


Abb. 1: App-Typen (Quelle: blog.meltmedia.com)

Kriterium	Native App	Hybride App	Web App
Performance	+	+	o
Offline-Verfügbarkeit	+	+	o
Native Features	+	+	-
Installation/ Einrichtung	-	-	+
Erreichbarkeit	+	+	+
Kosten	-	+	+
Inhalte	-	-	+

Abb. 2: Entscheidungsmatrix

Spaß hat allerdings seinen Preis: So beginnt die günstigste Lizenz ohne Rabatt bei 999 $ pro Entwickler.

Auch Sencha Touch wird als führendes Framework im Bereich der Entwicklung von plattformübergreifenden mobilen Web Apps angepriesen. Genau wie bei Kendo UI werden hier iOS, Android, WindowsPhone und Blackberry unterstützt. Entwickelt wird ausschließlich mit HTML, CSS und JavaScript. Dafür kann man mithilfe des Frameworks neben Gesten und Multitouch auch auf den Speicher und die GPS-Informationen des Geräts zugreifen. Die Lizenzen sind mit knapp 4.500 $ für 5 Entwickler im Durchschnitt ähnlich teuer, wie bei der Konkurrenz. Selbstverständlich kann auch Sencha mit großen Kunden werben, unter denen sich beispielsweise Airbus und Amazon befinden.

Zwei der bekanntesten Frameworks für hybride Apps stellen Apache Cordova und Adobe PhoneGap dar. Nicht ohne Grund werden die beiden Namen häufig synonym verwendet. Zu Beginn gab es lediglich das PhoneGap Framework, bevor dessen Quellcode 2011 an Apache gespendet wurde. Daraus entstand das Framework Cordova. PhoneGap ist auch weiterhin als Framework auf dem Markt bekannt, basiert jedoch auf Cordova. Die

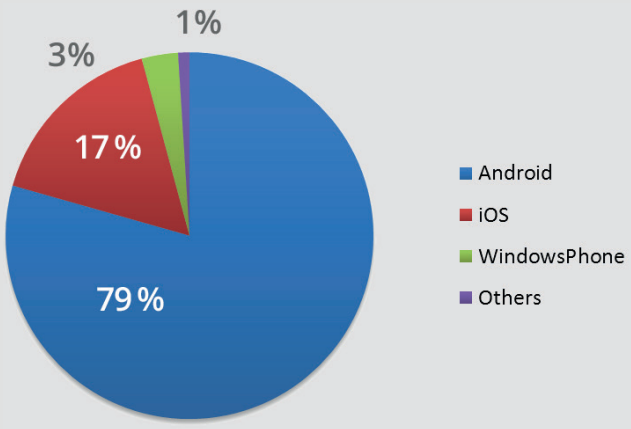


Abb. 3: Marktanteile Smartphones 2015

Quellen

[1] ORDIX news 2/2014
GWT und Web Storage
http://www.ordix.de/ordix-news-archiv/2-2014.html

[2] Native Apps vs. Web Apps vs. Hybride Apps
https://app3null.com/native-hybride-web-apps/

[3] Blog: There´s More Than One Way to Build Mobile Apps:
http://blog.meltmedia.com/2013/05/theres-more-than-one-way-to-build-mobile-apps/

[4] Hybride App-Entwicklung
http://t3n.de/news/hybride-app-entwicklung-frameworks-617199/

[5] Diskussionsforum von John Bristowe;
http://developer.telerik.com/featured/what-is-a-hybrid-mobile-app/#disqus_thread

[6] Angebot an mobilen Frameworks:
http://mobile-frameworks-comparison-chart.com/

[7] UI Library to Speed Up Your HTML/JS Development:
http://www.telerik.com/kendo-ui

[8] Sencha Touch:
https://www.sencha.com/products/touch/

[9] Apache Cordova:
https://cordova.apache.org/

beiden Frameworks unterstützen viele verschiedene Geräte, sowie einige von deren Hardware-Funktionen. Anleitungen, Bücher und mehr sind zahlreich im Internet zu finden, sodass eine Einarbeitung nicht allzu schwer fallen dürfte. Beide Frameworks werden unter der Apache Lizenz, Version 2.0 vertrieben und sind somit frei verfügbar.

Ein weiteres Framework, das auf jeden Fall genannt werden sollte, ist Ionic. Ionic ist im Vergleich zu den bereits genannten Konkurrenten eher als Newcomer anzusehen. Dennoch ist das Framework nicht zu unterschätzen. Ionic unterstützt eine Vielzahl von nativen Features, darunter Kamera- und Dateizugriff, sowie den Beschleunigungsmesser. Die Entwicklung der hybriden App findet dabei durch HTML, CSS und JavaScript statt. Anschließend wird, ähnlich wie bei PhoneGap und Apache Cordova, das Deployment genutzt. Die Nutzung des Ionic Frameworks ist völlig kostenlos. Wer mag, kann sich trotzdem für eine kostenpflichtige Enterprise-Variante entscheiden, um auf den Support zurückgreifen zu können. Einen kleinen Haken hat Ionic allerdings: Es werden zurzeit lediglich iOS und Android unterstützt. Eine Umsetzung für WindowsPhone wurde kürzlich für Ionic Version 2 angekündigt, während das Einbinden von FirefoxOS noch in Planung ist.

Man könnte die Liste der Frameworks selbstverständlich noch weiterführen, was die Auswahl allerdings nicht erleichtern würde. Daher sei gesagt, dass es ein paar Kriterien gibt, nach denen Sie einige Frameworks direkt ausschließen und sich Ihre Entscheidung so erleichtern können. Im Jahr 2015 hatten Android und iOS ca. 95% der Marktanteile (siehe Abbildung 3), weshalb es auch nicht verwunderlich ist, dass Sie diese beiden Plattformen mit den meisten Frameworks bedienen können. Möchten Sie zum Beispiel noch WindowsPhones unterstützen, werden Ihre Wahlmöglichkeiten eingeschränkt. Auch die Entwicklungssprachen sind ein Kriterium, nach dem gut gefiltert werden kann. Denn nicht alle Frameworks basieren ausschließlich auf den Web-Technologien. Die Unterstützung der nativen Features der Smartphones eignet sich sehr gut, um die Anzahl der möglichen Frameworks zu reduzieren. Zu guter Letzt bietet sich auch das Lizenzmodell als Entscheidungskriterium an. Online finden Sie unter http://mobile-frameworks-comparison-chart.com ein Tool, um die benannten Filter automatisch anzuwenden.

Ich hoffe, ich konnte Ihnen die Entscheidung bei der Wahl der richtigen App erleichtern. Ich freue mich, wenn Sie mir von Ihrer persönlichen Erfahrung berichten.

Sebastian Schäfers
(info@ordix.de)

Storage im Wandel der Zeit

Noch vor ein paar Jahren waren die Anforderungen übersichtlich, die Unternehmen an ein Storage stellten. Es sollte vor allem schnell, zuverlässig und relativ preisgünstig je Gigabyte sein. Heute unterliegen Storages ganz anderen Kriterien.

Geschichte der Speicher und Storages

Von Anfang an sind Computer auf externe Speicher angewiesen, auf denen die auszuführenden Programme, die zu verarbeitenden Daten und die Ergebnisse der Verarbeitung dauerhaft gespeichert werden. Von der Lochkarte von Hollerith 1887 bis zu heutigen modernen Speichersystemen war es ein weiter Weg.

Im Zuge der Entwicklung wurden die Geräte immer kleiner und schneller. Manch einer erinnert sich noch an die Winchester-Platten im Format einer Waschtrommel, deren Speicherkapazität bei einem Tausendstel einer modernen 2,5-Zoll-Festplatte lag. Auch die Leistung wurde permanent gesteigert, konnte jedoch mit denen von CPU und Arbeitsspeicher nicht mithalten. Diese Lücke konnte erst in den letzten Jahren durch den Einsatz von SSDs und Flash-Speichern weitgehend beseitigt werden.

Welche Speichertechnologien werden heute verwendet?

Nach dem Schwenk 2004 von Parallel-SCSI auf Serial-Attached-SCSI stiegen alle paar Jahre die Übertragungsgeschwindigkeiten, 2015/2016 haben fast alle Hersteller auf SAS III mit 12 Gbit/s umgestellt. Als Speichermedien kommen dabei zwei Produkte zum Einsatz: Flash-basierte Speicher in Form von Modulen oder SSDs und herkömmliche Festplatten als SAS-Disks und sog. NLSAS-Disks.

Flash-Module, auch Flash-EEPROM, gibt es in verschiedenen Bauformen, entweder als eigenständige Module oder als SSD mit Kapazitäten von 1-6 TB. Sie besitzen keine mechanischen Teile mehr und sind von der Stromaufnahme und Kühlung wesentlich günstiger als herkömmliche Festplatten. Flash-Module sind die zurzeit schnellsten Speichermedien. Sie werden als Cache-Erweiterung oder als Disk eingesetzt.

Als Festplatten kommen hauptsächlich SAS-Disks mit 10.000 und 15.000 UpM in 2,5-Zoll-Bauform vor. Diese Disks verfügen über eine SAS-III-Schnittstelle. Sie sind mit Kapazitäten von 300 GB bis zurzeit 1,6 TB erhältlich. Als NL-SAS werden die hochkapazitativen 3,5-Zoll-SATA-Disks mit bis zu 8 TB bezeichnet. Damit SATA-Disks in einem modernen Storage verwendbar sind, wurden sie mit einer SAS-Schnittstelle versehen, daher der Name NearLine-SAS. Sie dienen hauptsächlich Archivierungszwecken.

Anforderungen und Auswahl moderner Speichersysteme

An moderne Speichersysteme werden viele neue Anforderungen gestellt, seien es Big Data, Storage as a Service, Cloud, Hochverfügbarkeit oder Virtualisierung, um nur einige zu nennen. Den steigenden Ansprüchen stehen allerdings sinkende Budgets gegenüber. So wird die Auswahl und Zusammenstellung eines geeigneten Speichersystems zu einer immer komplexeren Aufgabe und der Druck auf die IT-Verantwortlichen wächst in zunehmendem Maße.

Aus der Vielzahl von Speichersystemen das für das Unternehmen passende zu finden, ist mittlerweile ein komplexer Prozess geworden. So gilt es neben dem geschätzten Datenwachstum der nächsten Jahre alle Anforderungen an das neue System zu sammeln und in ge-

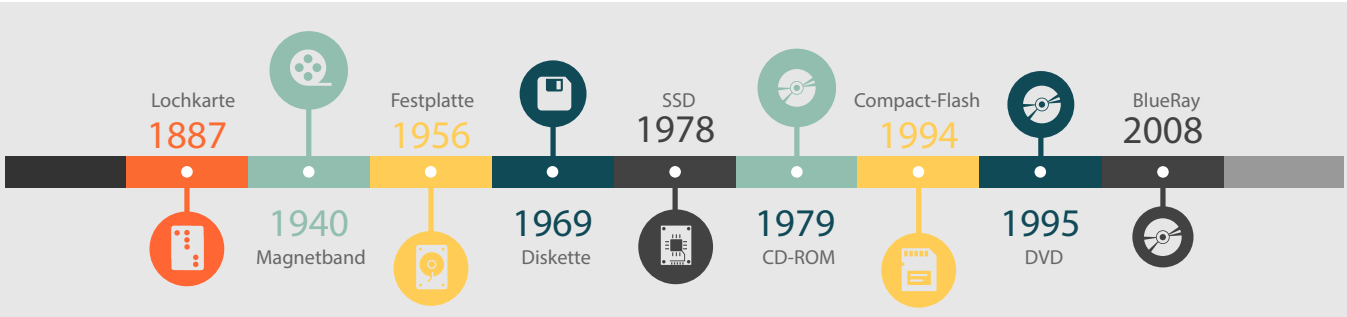


Abb. 1: Entwicklung der Speichermedien

eigneter Form auszuwerten und auszuwählen. Durch die Vielzahl der Funktionen und das fast unüberschaubare Angebot kommt es dabei immer wieder zu Fehlgriffen – das Unternehmen investiert in ein Speichersystem, das die geforderte Leistung nicht erbringt.

Hier eine Auswahl der zu berücksichtigender Faktoren:

- zu erwartende IO-Last
- Kapazität heute, in 1/2/3 Jahren
- Cache-Größen
- Hochverfügbarkeit
  (lokal, remote, synchron, asynchron)
- Funktionen
  (Snapshots, Tiering, Deduplikation, Kompression, etc.)
- physische Umgebung
  (Flächenbedarf, Stromverbrauch, Kühlung)
- Virtualisierung
- Administration

Durch die vielfältigen Funktionen der Speichersysteme ist alleine schon die Aufteilung der Hardware eine Herausforderung. SSDs und Platten müssen zu Einheiten gruppiert und konfiguriert werden, um jeder Applikation den schnellsten und dabei möglichst kostengünstigsten Speicher bereitstellen zu können.

Funktionen moderner Speichersysteme

Um die Anforderungen erfüllen zu können, haben die Hersteller nach und nach zahlreiche neue Funktionen in die Firmware integriert. Für die Datensicherheit und -sicherung werden Snapshots und Volume-Kopien lokal und remote verwendet. Damit lassen sich sehr schnell einfache und kostengünstige Sicherungen erstellen, die aber nicht das Backup oder die Archivierung ersetzen können.

Abb. 2: Flash-Module

Abb. 3: SAS-Disk/NLSAS-DISK

Mit Deduplizierung und Komprimierung soll der verwendete Speicherplatz besser genutzt werden. Das ist aber nur teilweise möglich. Mittels Deduplizierung kann ein Unternehmen eine spürbare Reduktion des benötigten Speicherplatzes erreichen, wenn es darum geht, eine Vielzahl von gleichartigen Daten wie z.B. VM-Images zu speichern. Hier wird dann nur noch ein Image vollständig gespeichert, bei allen anderen nur die Abweichungen von dem Urabbild.

Bei allen anderen Daten, strukturiert wie auch unstrukturiert, kann durch Deduplizierung kein Speicherplatz gewonnen werden; hier versucht man es mit Kompression. Bei strukturierten Daten, zumeist Datenbanken, und einfachen Dateien wie Mails, Dokumenten usw. kann man hier gute Ergebnisse erzielen. Die Masse der Multimedia-Daten dagegen lässt sich nicht mehr komprimieren. Diese Daten werden von Haus aus häufig schon in einem verdichteten Zustand gespeichert.

Tiering verbindet die deutlich teureren Flash-Speicher mit den herkömmlichen Festplatten, um die Leistungsfähigkeit des Systems insgesamt zu erhöhen. Das System erkennt dabei die am häufigsten angeforderten Daten und verschiebt sie automatisch auf die schnellsten teuren Speichermedien, die weniger genutzten Daten wandern nach und nach auf die langsameren günstigen Medien.

Zusätzlich findet immer mehr die Virtualisierung im Speicherbereich Einzug. Sie ist teilweise schon ein unverzichtbares Instrument, um der wachsenden Datenflut und deren Applikationen den immer passenden Speicher zur Verfügung zu stellen.

Immer größere Festplatten und wachsende Datenbestände bringen die bewährten RAID-Gruppen an ihre Grenzen, das Recovern einer TB-Platte nimmt je nach Größe nicht nur Stunden, sondern mittlerweile Tage in Anspruch. Die Gefahr von Datenverlusten steigt damit überproportional. Um dieses Manko zu umgehen, implementieren die Hersteller als neueste Funktion das Erasure Coding. Dabei werden die Daten mittels einer mathematischen Formel fragmentiert und diese Fragmente mit einer konfigurierbaren Anzahl von Kopien an unterschiedlichen Stellen in einem Speicher-Array abgelegt. Beim Recover-Prozess braucht daher nicht mehr die gesamte Platte wieder hergestellt zu werden, nur die verlorenen Teile einer Datei werden wieder aufgebaut.

Wie sehen die Speichersysteme von morgen aus?

Versuchen wir einen Blick in die Kristallkugel zu werfen. Der Markt für herkömmliche Festplatten wird weiter einbrechen, der Preisverfall bei den Flash-Speichern wird weiter zunehmen und sie werden in der nächsten Zeit die erste Wahl für schnelle Datentransferraten bleiben. So bieten heute alle Hersteller schon All-Flash-Arrays an. In den Laboren von IBM forscht man an Phase-Change-Memory-Bausteinen (PCM), sie sollen noch schneller als

Flash-Bausteine sein und dabei günstiger als DRAMs. IBM will in nächster Zeit die ersten Module auf den Markt bringen.

Cloud-Dienste und -Anbieter werden zunehmen und damit einhergehend wird auch die Menge der gespeicherten Daten exorbitant wachsen. Täglich werden heute schon mehr als 2,5 Trillionen Bytes gespeichert, 90 % aller Daten wurden erst in den letzten 2-3 Jahren auf digitalen Medien abgelegt. Derzeit sind etwa 10.000 Exabyte weltweit gespeichert, bis zum Jahr 2020, also in nur 4 Jahren, soll sich dieser Wert vervierfachen auf über 40.000 Exabyte.

Dadurch wird nicht nur eine zunehmende Konsolidierung von Rechenzentren stattfinden, auch Übernahmen zwischen den Herstellern zeichnen sich ab. Es bleibt abzuwarten, ob sich dadurch Standardisierungen bei den Speichersystemen ergeben. Die Virtualisierung wird sicherlich weiter ausgebaut, gleichzeitig wird dabei die Administration von Speichersystemen vereinfacht und teilweise automatisiert werden.

Die Auswahl und Zusammenstellung eines komplexen Speichersystems wird immer aufwendiger werden. Hier bildet sich in der letzten Zeit ein neuer Beratungszweig heraus, um Kunden herstellerneutral und fachübergreifend beraten zu können.

Backup/Restore/Archivierung

Gespeicherte Daten müssen kurz- bis mittelfristig gesichert und langfristig archiviert werden. Unter Backup versteht man die kurzfristige Sicherung der Daten auf anderen Medien. Hier kommen zumeist Datenbänder zum Tragen. Sie sind immer noch das günstigste Speichermedium je Gigabyte und in der Lage, Daten auch über längere Zeiträume sicher zu speichern. Die Speicherdichte auf den Bändern wird weiter zunehmen, auf den aktuellen LTO7-Bändern lassen sich bereits 6 TB unkomprimiert speichern und bei LTO8 soll dieser Wert verdoppelt werden.

Interessant bei diesen Datenmengen sind die Restore-Zeiten im Desaster-Fall. Das Zurückschreiben von den Datenbändern auf die Festplatten geht zwar deutlich schneller als das Schreiben auf Bänder, für die schelle Wiederherstellung der Daten muss aber auch die IT-Infrastruktur entsprechend aufgebaut sein. Hochkapazitative Leitungen und mehrfache Anbindung sind hier Pflicht.

Weitere Probleme stellen sich bei der Archivierung von Daten. Die zum Teil exorbitant langen Aufbewahrungsfristen stellen die Unternehmen immer wieder vor neue Probleme. So müssen Geschäftsunterlagen mind. 10 Jahre aufbewahrt werden. In medizinischen und anderen Branchen steigen diese Fristen auf 30 und mehr Jahre. Die Auswahl geeigneter Medien ist dabei problematisch: Wer kann denn heute noch Daten von Datenträgern lesen, die älter als 10 Jahre sind, egal ob Band oder Festplatte? Die Speicher-Technologie hat sich hier komplett gewandelt.

Ein weiteres Problem bei der Archivierung ist die Wiederauffindbarkeit. Die Speicherhersteller bieten auch hier

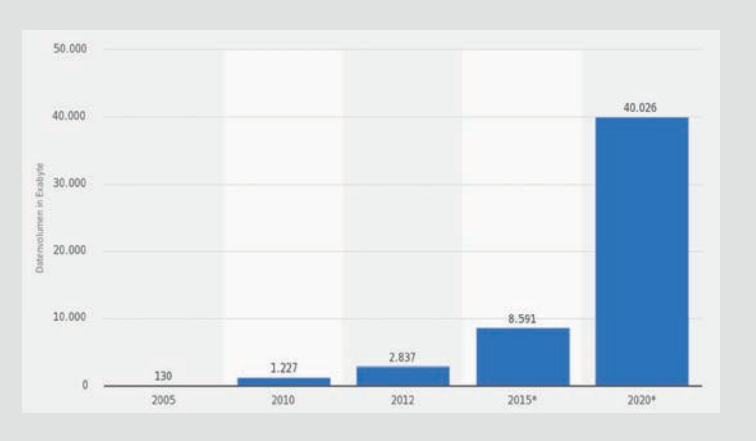


Abb. 4: Geschätztes Datenwachstum bis 2020 (Quelle: de.statista.com)

Lösungen an, die mit Katalogen, Indizes und Verschlagwortung Daten z. T. automatisch versehen und damit wiederauffindbar machen, aber leider gibt es keinen Standard. Und die verwendete Technologie setzt wieder auf den bekannten Speichersystemen mit den gleichen bekannten Problemen auf.

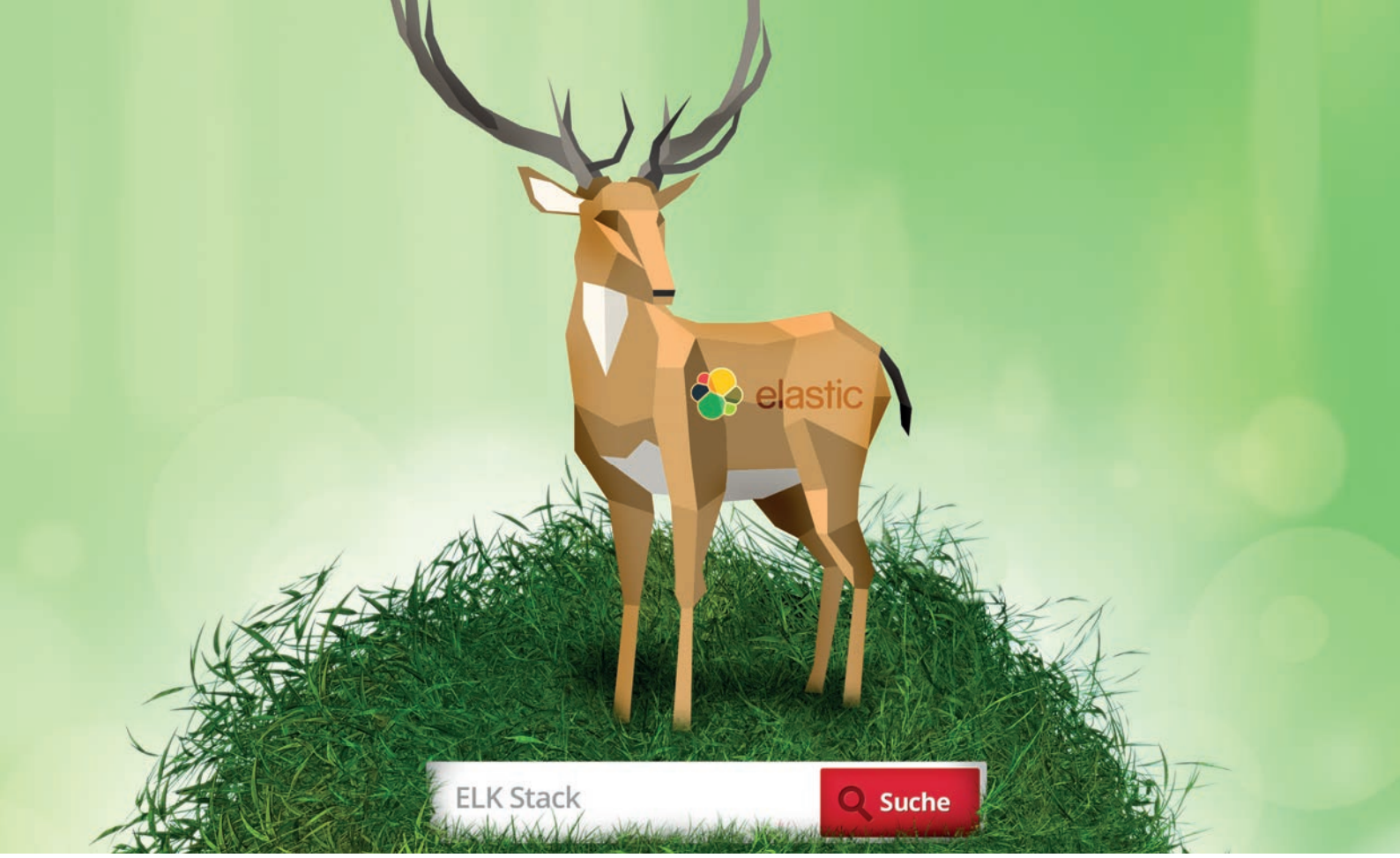
Fazit

Die ökologischen Folgen durch den Bau von immer mehr Rechenzentren und Speichersystemen und deren Stromversorgung und Kühlung sind nicht absehbar. Schon heute stoßen wir teilweise an die Grenzen des noch Machbaren. Allein die Erzeugung des notwendigen elektrischen Stroms stellt heute schon in einigen Ländern ein Problem dar.

Unternehmen und Behörden unterliegen gesetzlichen Vorgaben zur Speicherung und Aufbewahrung von Daten, die zum Teil unsinnig sind. Durch die zunehmende Vernetzung der Unternehmen untereinander und mit den Behörden reicht es aus, Daten eines Geschäftsvorfalls nur noch an einer Stelle zu sichern. Heute muss jedes Unternehmen für sich die Daten archivieren und die staatlichen Stellen sichern das Ganze dann noch einmal. Hier ist die Gesetzgebung dringend gefordert, die teilweise unsinnigen Fristen und Vorgaben entweder zu verkürzen oder abzuschaffen.

Und auch die allgemeine Sammelwut von Daten im privaten Bereich, hauptsächlich von unstrukturierten Daten wie Fotos, Filmen usw. bewirkt ein immer Mehr an Speicherplatz. Jeder muss sich daher selber fragen, welche Daten er wirklich speichern muss und ob nicht weniger manchmal doch mehr ist. Jedenfalls wird das Thema weiter spannend und interessant bleiben.

Klaus Grote
(info@ordix.de)



ELK Stack: Volltextsuche und Logarchivierung

Elasticsearch & Co

Die Volltextsuche in großen Datenmengen war bis vor einiger Zeit noch Aufgabe von relationalen Datenbanken (z. B. mithilfe von Oracle TEXT) oder dem auf Lucene basierenden Apache Solr. Seit einiger Zeit gewinnt jedoch der ELK Stack immer mehr an Bedeutung. Daher stellen wir in diesem Artikel die Suchmaschine Elasticsearch sowie weitere Komponenten des ELK Stacks Logstash und Kibana vor.

Elasticsearch

In jedem Unternehmen wächst die Zahl der produzierten Daten. Gerade Logdaten werden, aufgrund ihrer Menge oder ihrer geringen Werthaltigkeit, oft nur für kurze Zeit oder auch gar nicht aufbewahrt. Genau zu diesem Zweck wurde Elasticsearch von der Firma Elastic als Open-Source-Projekt entworfen. Es bietet eine vergleichsweise günstige Möglichkeit, Daten im JSON-Format zu speichern und in einer Volltextsuche für die verschiedensten Anwendungen zur Verfügung zu stellen.

Elasticsearch positioniert sich seit einigen Jahren erfolgreich als Alternative zu der ebenfalls auf Lucene basierenden Suchmaschine Apache Solr. Grund für den Erfolg von Elasticsearch dürften unter anderem folgende Eigenschaften sein: Elasticsearch bietet eine Hochverfügbarkeit durch ein relativ einfaches Clustering-Modell. Es kann, aufgrund seiner guten Lastverteilung und der Möglichkeit, jederzeit neue Knoten hinzuzufügen, sehr gut skalieren. Zudem ist es fast schemalos, dokumentenorientiert, bietet eine REST- sowie Java-API und kann in bestehende Umgebungen integriert werden. Auch unter hoher Änderungslast finden Dokumente schnellen Eingang in die Suche. Elasticsearch selbst spricht sogar von einer Echtzeitsuche. Darüber hinaus ist Elasticsearch sehr einfach in der Handhabung: Benötigt werden lediglich ein aktuelles JDK, cURL und ein Texteditor.

Speicherung

Elasticsearch verfolgt einen dokumentenzentrierten Ansatz, d. h. sämtliche Daten werden im verbreiteten JSON-Format übergeben und gespeichert. Die interne Ablage der Daten erfolgt in Indizes. Ein Index bildet dabei einen sowohl physischen als auch logischen Namensraum. Er kann hierbei mehrere unterschiedliche Dokumente aufnehmen und besteht selbst auch aus einem oder mehreren Doku-

mententypen. Die Unterteilung in mehrere Typen ermöglicht eine bessere Durchsuchbarkeit des Indexes (s.u.). Die physische Speicherung eines Indexes erfolgt in mehreren Primary Shards. Diese wiederum stellen einzelne Lucene-Instanzen dar. Die Aufteilung der Indexes in mehrere Shards dient der Lastverteilung: Zur physischen Lastverteilung können die einzelnen Shards auch auf mehrere Knoten eines Elasticsearch-Clusters verteilt werden. Aus den Primary Shards können darüber hinaus Replica Shards erzeugt werden. Diese sind eine Kopie ihres jeweiligen Primary Shards und befinden sich auf anderen Knoten des Clusters. Sie bieten Schutz vor Datenverlust, ermöglichen eine Hochverfügbarkeit und dienen zur Optimierung von Suchanfragen. Die Knoten eines Clusters haben als Konfigurationseinstellung immer denselben Clusternamen (Standard: „Elasticsearch“). Daher können mehrere Cluster nebeneinander im selben Netzwerk betrieben werden.

Cluster-Ansatz

Ein Elasticsearch-Cluster ist nach dem Master-Slave-Prinzip aufgebaut. Grundsätzlich kann jeder Knoten des Clusters beide Rollen annehmen. Beim Start führt jeder Knoten einen Discovery-Prozess durch und sucht nach anderen Knoten im gleichen Netzwerk. Die Knoten handeln den Master-Knoten selbst aus. Fällt dieser aus oder ist zeitweise nicht erreichbar, wählen sie aus ihrer Mitte einen neuen Master-Knoten.

Installation

Die Installation eines Elasticsearch-Knotens ist einfach. Das Installationspaket muss lediglich entpackt werden und der Knoten über den Befehl `./bin/elasticsearch -f` gestartet werden. Elasticsearch verfolgt einen Zero-Konfiguration-Ansatz. Alle möglichen Einstellungen sind bereits mit sinnvollen Standardwerten vorbelegt. Diese befinden sich in der im YAML-Format geschriebenen Konfigurationsdatei `config/elasticsearch.yml`.

Vor dem Start des Knotens empfiehlt es sich, den Cluster-Namen anzupassen, um den neuen Knoten nicht versehentlich in das falsche Cluster einzuhängen. Nach dem Start ist der Knoten über den Port 9200 erreichbar. Sollen dem Cluster weitere Knoten hinzugefügt werden, muss darauf geachtet werden, dass diese sich im selben Netzwerk und unter demselben Clusternamen befinden.

Dokumente laden und ändern

Wenn das Cluster läuft, können nun die ersten JSON-Dokumente übergeben und gespeichert werden. Die Übergabe der Dokumente kann entweder per HTTP-Put oder HTTP-Post erfolgen. Abbildung 1 zeigt die Übertragung von Adressinformationen der ORDIX AG in Form eines JSON-Dokumentes an den Elasticsearch-Knoten „`es1.ordix.de`“. Dieses Dokument soll dort, wie an der URL zu erkennen, in dem Index „`customers`“ als Dokumententyp „`company`“ abgelegt werden. Jeder Knoten des Elasticsearch-Clusters kann dazu über eine REST-konforme URL angesprochen werden. Es ist nicht nötig,

```
curl -X POST http://es1.ordix.de:9200/customers/compa-
ny -d '{
    "name": "ORDIX AG",
    "street": „Westernmauer 12-16",
    "city" : "Paderborn",
    "zipcode" : 33098,
    "country" : „Deutschland",
     "ceo" : "Wolfgang Kögler"
}’
{
  "ok" : true,
  "_index" : "customers",
  "_type" : "company",
  "_id" : "RGaH58ppD-Xj_gdkZ34tx3",
  "_version": 1
}
```

Abb. 1: Übertragung eines JSON-Dokuments mit entsprechender Antwort

```
//Überschreiben
curl -X PUT http://es1.ordix.de:9200/customers/compa-
ny/RGaH58ppD-Xj_gdkZ34tx3 -d '{
    "name": "ORDIX AG",
    "street": "Karl-Schurz-Str. 19A",
    "city" : "Paderborn",
    "zipcode" : 33100,
    "country" : "Deutschland",
     "ceo" : "Wolfgang Kögler"
}’
//lesen
curl -X GET http://localhost:9200/addresses/
german/1234?pretty
//löschen
curl -X DELETE http://localhost:9200/addresses/
german/1234?pretty
```

Abb. 2: Ändern, lesen und löschen eines JSON-Dokuments

```
curl -X GET http://es1.ordix.de:9200/customers/
company/_mapping?pretty
{
  "company" : {
    "properties" : {
    "name" : { "type" : "string" },
    "street" : { "type" : "string" },
    "city" : { "type" : "string" },
    "zipcode" : { "type" : "long" },
    "ceo" : { "type" : "string" }
      }
   }
}
```

Abb. 3: Abfrage von Schemainformationen

```
curl -X GET http://es1.ordix.de:9200/customers/compa-
ny/_search -d '{
  "query" : {
    "term" : {
      "city" : "Paderborn"
    }
   }'
 {
     "took" : 1,
     "timed_out" : false,
     "_shards" : {
       "total" : 1,
       "successful" : 1,
       "failed" : 0
     },
     "hits" : {
       "total" : 1,
       "max_score" : 0.30531824,
       "hits" : [ {
         "_index" : "customers",
         "_type" : "company",
         "_id" : "RGaH58ppD-Xj_gdkZ34tx3",
         "_score" : 0.30531824, "_source" : {
          "name": "ORDIX AG",
          "street": "Karl-Schurz-Str. 19A",
          "city" : "Paderborn",
          "zipcode" : 33100,
          "country" : „Deutschland",
          "ceo" : "Wolfgang Kögler"
         }
       } ]
     }
}
```

Abb. 4: Suche und Ergebnis

```
curl -X GET http://es1.ordix.de:9200/customers/compa-
ny/_search?pretty -d '{
  "fields" : ["name", "city"],
  "query" : {
    "match" : {
      "name" : "Odix"
    }
  }
}
{
  "took" : 2,
  "timed_out" : false,
  "_shards" : {
    "total" : 1,
    "successful" : 1,
    failed" : 0
  },
  "hits" : {
    "total" : 1,
    "max_score" : 0.002537901,
    "hits" : [ {
      "_index" : "customers",
      "_type" : "company",
      "_id" : "RGaH58ppD-Xj_gdkZ34tx3",
      "_score" : 0.002537901,
      "fields" : {
        "name" : "ORDIX AG",
        "city" : "Paderborn"
      }
    } ]
  }
}
```

Abb. 5: Volltextsuche mit unscharfem Suchbegriff.

den Master-Knoten anzusprechen. Elasticsearch quittiert jeden Aufruf mit einem JSON-Dokument. Dieses beinhaltet in diesem Fall eine Information über den Erfolg der Operation, den verwendeten Index sowie die ID und Version des betroffenen Dokuments.

In Abbildung 1 wurde ein Dokument ohne ID übermittelt. Daher generiert Elasticsearch selbst eine neue, eindeutige Dokumenten-ID. Unter Angabe der ID ist es möglich, ein Dokument zu überschreiben, zu lesen oder zu löschen. Dazu muss das Dokument allerdings per HTTP-Put und nicht, wie im vorigen Beispiel, per HTTP-Post übergeben werden.

Ist bereits ein Dokument mit dieser ID im Elasticsearch-Cluster vorhanden, wird das bestehende Dokument überschrieben und die Versionsnummer inkrementiert. Abbildung 2 zeigt, wie ein Dokument überschrieben, gelesen oder gelöscht werden kann.

Schemafreiheit

Auch wenn Elasticsearch von außen betrachtet schemafrei erscheint, ist es dies nicht wirklich. Bei der Übergabe von Dokumenten müssen keine Datentypen o. Ä. definiert werden. Diese werden automatisch aus den übergebenen Dokumenten abgeleitet: Ein aus Zeichen bestehendes Feld wird als String interpretiert, Zahlen deuten auf numerische Werte hin und für Datumsangaben stehen einige Grundformate zur Verfügung. Das Schema eines Dokumententyps kann, wie in Abbildung 3 gezeigt, auch abgefragt werden.

Suchen und finden

Hauptaufgabe einer Suchmaschine wie Elasticsearch ist das Suchen und Finden von Dokumenten. Diese bietet dazu eine eigene Abfragesprache: die Elasticsearch Query DSL. Diese Sprache bietet neben einfachen auch sehr komplexe Abfragen mit boolschen Operatoren, Gruppierungen oder Hierarchien.

Abbildung 4 zeigt die einfachste Form einer Abfrage: die Suche nach einem konkreten Term, der genau so in dem zu findenden Dokument vorhanden sein muss. Das dort dargestellte Ergebnis teilt sich in einen Header, der allgemeine Informationen zur Suchanfrage liefert, und die dazu gefundenen Dokumente.

Eine Volltextsuche macht jedoch nur Sinn, wenn eine Suchanfrage auch bei sehr unscharfen Anfragen gute Ergebnisse liefert. Dafür steht u. a. die `match`-Query zur Verfügung. Abbildung 5 zeigt eine Suchanfrage nach dem String „Ordix“. Das Ergebnis liefert zudem ein Dokument zu „ORDIX AG“. Daneben besitzt die Elasticsearch Query DSL noch viele weitere interessante Features, deren Erklärung den Rahmen dieses Artikels sprengen würde. Auf der Elasticsearch-Webseite befindet sich ein sehr gutes Tutorial zum Erlernen der Sprache. So ist es z. B. möglich, die Fähigkeiten durch eigene Skripte (Funktionen) zu erweitern oder den Dokumenten eine „Time to live“ (TTL) mitzugeben.

Logstash

Logstash ist ebenfalls ein von der Firma Elastic entwickeltes Werkzeug, um Logdaten in JSON-Dokumente umzuwandeln und in einem Elasticsearch-Cluster zur Verfügung zu stellen. In seiner Grundfunktion ist es eine netzwerkfähige Pipe mit verschiedenen Filtermöglichkeiten. Seine Aufgabe ist es, Meldungen aus verschiedenen Inputs (Dateien, log4j u.a.) zu empfangen, zu filtern und an einen oder mehrere Outputs weiterzuleiten.

Logstash selbst hat keine Möglichkeit, Daten zu speichern. Es ist dazu ausgelegt, die Daten in Elasticsearch abzulegen. Zur kurzfristigen Speicherung der Daten ist es jedoch möglich, diese in einer Redis-Datenbank zwischenzuspeichern, beispielsweise für den Fall einer kurzzeitigen Nichterreichbarkeit oder Überlastung des Elasticsearch-Clusters. So kann der Datenfluss entkoppelt und ein Datenverlust vermieden werden. Die Handhabung ist auch hier sehr einfach. Alle benötigten Informationen zu Quellen, Zielen und Informationen zum Parsen der Daten müssen in einer Konfigurationsdatei erfasst werden. Anschließend kann Logstash ebenso einfach wie Elasticsearch installiert und gestartet werden.

Logstash-Filter

Die Filterfunktionen in Logstash dienen in erster Linie dem Aufbau des JSON-Dokuments und nicht dazu, Daten herauszufiltern. Zum Beispiel kann die Zeile einer Logdatei mehrere Informationen beinhalten, wie den Rechnernamen, Zeitstempel, Messagetyp Error oder Debug. Durch die Verwendung von Filterfunktionen können diese Informationen extrahiert und in dem zu generierenden JSON-Dokument in eigenen Feldern abgelegt werden.

Zu den bekanntesten Filterfunktionen gehört der Grok-Filter. Mit diesem Filter werden Daten mithilfe regulärer Ausdrücke durchsucht, wohingegen der Multiline-Filter mehrere Zeilen einer Logdatei zu einem JSON-Dokument zusammenführt und so einen Stacktrace in einem JSON-Dokument abbildet. Daneben gibt es noch weitere Filter, wie z. B. Geo-IP. Dieser kann IP-Adressen um Geoinformationen ergänzen, um so die Herkunft der Besucher einer Webseite festzustellen. Es besteht auch die Möglichkeit, alle Filter hintereinander zu verwenden und somit zu kombinieren. Abbildung 6 zeigt das Beispiel einer Logstash-Konfigurationsdatei. Diese Datei teilt sich in drei Blöcke: Input, Filter und Output. Der Input-Block definiert die Datenquelle, in diesem Fall eine Datei. Im Filter-Block wird zunächst ein Multiline-Filter verwendet, um beispielhaft den mehrere Zeilen umfassenden Stacktrace eines Spring-Logs in einem JSON-Dokument zusammenzufassen.

Im sich anschließenden Grok-Filter werden nun aus diesen Informationen der Zeitstempel und die Prozess-ID extrahiert. Im Output-Block wird als Ziel das Elasticsearch-Cluster angegeben. Logstash erstellt standardmäßig jeden Tag einen neuen Index mit einem konfigurierbaren Prefix (standardmäßig "`logstash_`"), gefolgt von dem Erstellungsdatum.

```
input {
  file {
    path => ["/log/spring.log"]
    start_position => beginning
    type => „spring_log“
  }
}
 filter {
  multiline {
    pattern => "((^\s*)[a-z\$\.A-
Z\.]*Exception.+)|((^\s*)at .+)"
    what => "previous"
  }
  grok {
    match => [ "message",
      "^(?[0-9]{4}\-[0-9]{2}\-[0-9]{2})
       %{TIME:time}
       (?:\s*)
       (?[A-Z]+)
       %{NUMBER:pid}
       (?\[.*\])
       (?:\-\-\-)
       (?[0-9a-z\$A-Z\[\]/\.]*)
       (?:\s*:)
       (?(.|\s)+)"]
  }
  kv {}
}
 output {
  elasticsearch {
    host => es1.ordix.de
    index => "logs"
  }
}
```

Abb. 6: Logstash-Konfigurationsdatei

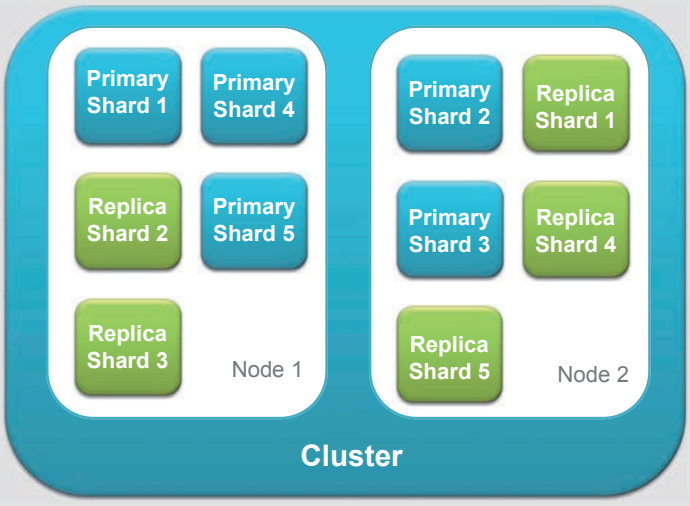


Abb. 7: Shard-Aufteilung in einem zwei Knoten Cluster

Glossar

DSL
Domain Specific Language (anwendungsspezifische Sprache)

YAML
Eine an XML angelehnte, vereinfachte Auszeichnungssprache

REST
Representational State Transfer, ein Programmierparadigma für verteilte Systeme

HTTP-Put
Request-Methode des Hyptertext Transfer Protokolls

HTTP-Get
Request-Methode des Hyptertext Transfer Protokolls

Redis
In-Memory NoSQL-Datenbank in der Form Key-Value Stores

TTL
Time to Live. Gültigkeitsdauer oder auch Aufbewahrungszeit von Daten

Links/Quellen

[1] Internetseite Elastic
https://www.elastic.co/de/

[Q1] Produktseite Elasticsearch
https://www.elastic.co/de/products/elasticsearch

Bildnachweis

Kibana

Kibana ist die dritte Komponente des ELK Stacks. Im Kern ist es eine Javascript-basierte Weboberfläche, welche auf ein Elasticsearch-Cluster per Webinterface zugreifen kann. Kibana dient der visuellen Darstellung von Abfragen des Elasticsearch-Clusters in Form von Tabellen, Säulen- oder Tortendiagrammen. Kibana bietet die Möglichkeit, die in Elasticsearch gespeicherten Daten anzuschauen und einfache Anfragen ohne Kenntnisse der Elasticsearch Query DSL „zusammenzuklicken“.

Fazit

Aus der Kombination von Elasticsearch, Logstash und Kibana wurde in einem Kundenprojekt der ORDIX AG ein Log Management System entwickelt. Dieses ermöglicht es, verschiedenste Data-Science-Anwendungen zur zentralen Analyse von Daten bisher nicht miteinander verbundener Systeme zu betreiben. Zudem ermöglicht dieses System, Entwicklern eine systemübergreifende, zentralisierte Fehleranalyse. Negativ fällt die hohe Änderungsrate der Elasticsearch Query DSL, ohne eine ausreichende Abwärtskompatibilität zu bieten, auf. Dies ist besonders ärgerlich, da auch der Aufwand bei der Erstellung der Abfragen nicht zu unterschätzen ist. Andererseits ist Elasticsearch durch seine Anpassungsfähigkeit und die konsequente Verwendung des verbreiteten JSON-Formats gerade in heterogenen Umgebungen sehr gut einsetzbar. Besonders überzeugen kann die mögliche Echtzeitsuche in Cloud-ähnlichen, dynamischen Umgebungen.

Philipp Loer
(info@ordix.de)

Neue Reihe: Enterprise Manager Oracle Cloud Control 13c

Schweben auf Wolke „13“

Nach Grid und der ersten Cloud ist Oracle, seit Dezember 2015, mit der „Cloud Control 13“ in der Wolke unterwegs. In der Artikel-Reihe zum Thema „Enterprise Manager Cloud Control 13“, stellen wir Ihnen neue Features und Möglichkeiten zur einfachen Anwendung in der Wolke vor. Seit Dezember 2015 ist die Version 13 des Enterprise Manager Oracle Cloud Control für alle gängigen Betriebssysteme verfügbar. Diese Version ist die Nachfolger-Version von Cloud Control 12 bzw. Grid Control Version 11.

Der Weg zur neuen Cloud

Wer aktuell noch ein Betreiber der alten Version Grid Control 10g/11g oder Cloud Control 12c ist, sollte sich Gedanken über ein Upgrade machen. Dazu gibt Oracle einen einfachen Weg vor – Neuinstallation! Selbstverständlich ist auch ein Upgrade von einer der genannten Versionen möglich. Jedoch stellt ein Upgrade die schwierigere Variante dar und ist nicht ohne Auszeiten von den überwachten Systemen durchzuführen. Gerade wenn Sie von Grid Control 10g/11g auf die neue Version wechseln, ist ein Upgrade sehr aufwendig. Für dieses Szenario empfiehlt sich daher eine Neuinstallation. Zusätzlich wird bei der Installation in der neusten Version der BI-Publisher mitinstalliert. Bei einem Upgrade muss er jedoch manuell nachinstalliert werden.

Alle denkbaren Möglichkeiten, sind in dem Cloud Control Upgrade Guide oder auch im Artikel „Das Gitternetz wird zur Wolke“, im dritten Teil unserer Artikelreihe Oracle Cloud Control 12c aus der Ausgabe 4/2012 nachzulesen.

Wieder eine neue Oberfläche?

Auch mit Oracle Cloud Control 13c hat sich an der Oberfläche wieder etwas getan. Logischerweise ist der Umstieg von 12c auf 13c nicht so gewaltig, wie er von den Grids war, aber er ist erwähnenswert. Für den gewohnten Cloud-Control-12c-Benutzer hat sich außer der Optik nicht viel verändert. In der 13c-Version setzt Oracle in der Oberfläche auf die aus 12c bekannten Drop-Down-Menüs. Der Grid-Control-Benutzer wird sich hier (wie schon beim Umstieg auf 12c) in einer deutlich strukturierteren Welt wiederfinden.

New Features

Wie bei Oracle üblich heißt es auch beim Enterprise Manager „New Version – New Features“. Die Version 13c des Enterprise Manager Cloud Control bietet eine Reihe von interessanten neuen Features.

Nachstehend finden sie eine kleine Übersicht:

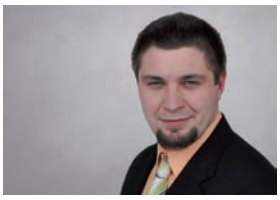
- „Always On“ Monitoring
- Monitoring & Incident Management
- Cloud Lifecycle Management
- BI Publisher Enterprise Reports
- Gold image based agent lifecycle management
- Database Consolidation workbench
- Exadata/Exalogic VM Provisioning
- Notification blackouts
- uvm.

Einige dieser Features werden wir in den kommenden Ausgaben thematisieren.

Fazit

Wir hoffen, wir haben Ihr Interesse auf die neue Artikelreihe „Enterprise Manager Oracle Cloud Control 13c” geweckt. Es erwarten Sie in weiteren Artikeln unter anderem Themen wie: BI-Publisher, Gold Agent Images, EM User Management, Monitoring und Incident Management. Auch mit dem Upgrade werden wir uns genauer befassen. Und wir werden sehen, ob die Version 13 ein gutes Omen ist.

Carsten Hummel
(info@ordix.de)



Michael Thieme
(info@ordix.de)

Moderne Webanwendungen mit JSF und PrimeFaces

Bereits im Jahre 2004 wurde die Spezifikation zu JavaServer Faces (JSF) in der Version 1.0 veröffentlicht [1]. Der Zeitraum von 2004 bis heute ist in der schnelllebigen Welt der IT eine halbe Ewigkeit. Nach und nach wurden immer mehr Webframeworks veröffentlicht – sogar so viele, dass es manchmal nicht einfach ist, überhaupt den Überblick zu behalten. Die aktuellsten Trends gehen weg von reinen Java-Webframeworks in Richtung reiner JavaScript-Framework (z. B. AngularJS, Node.js, usw.), um Anforderungen an moderne Webanwendungen besser bedienen zu können. In dem folgenden Artikel wollen wir überprüfen, ob JSF demnach zum alten Eisen gehört oder ob es im Zusammenspiel mit PrimeFaces auch noch heute eine gute Wahl für zeitgemäße Webanwendungen darstellt.

PrimeFaces

JSF ist ein komponentenorientiertes Webframework und wird sehr selten ohne eine zusätzliche Komponentenbibliothek verwendet. Über die Jahre hinweg sind viele Komponentenbibliotheken auf den Markt „gespült“ worden und viele davon im späteren Verlauf in der Menge untergegangen. PrimeFaces [2] ist eine dieser Komponentenbibliotheken, denen es jedoch gelungen ist, durch seine Vielzahl an Komponenten und guten Support nicht in der Masse unterzugehen. PrimeFaces mauserte sich zu der am häufigsten verwendeten und umfangreichsten Komponentenbibliothek für JSF. Daher ist es durchaus sinnvoll, dass wir in diesem Artikel überprüfen, ob JSF in Kombination mit PrimeFaces den Anforderungen an moderne Webanwendungen gewachsen ist.

Komponenten

PrimeFaces beinhaltet in erster Linie eine große Anzahl von Standardkomponenten, die bereits eine Vielzahl von

Anforderungen an moderne Webanwendungen erfüllen. Neben „einfachen“ Eingabefeldern werden auch Komponenten für komplexe Menüstrukturen, Charts und für das Einbinden von Multimedia angeboten. Auch Drag und Drop wird unterstützt und bietet dem Nutzer Funktionalitäten, die er bisher hauptsächlich von Desktop-Anwendungen kennt.

Das Einbinden dieser Komponenten in die eigene JSF-Anwendung ist denkbar einfach. Es muss lediglich die PrimeFaces-Bibliothek mitgeliefert und innerhalb einer View der zugehörigen Namespace „http://primefaces.org/ui“ eingebunden werden. Im Anschluss daran können die Komponenten, wie in JSF gewohnt, verwendet werden. In dem folgenden Beispiel wird die Kalenderkomponente von PrimeFaces aufgerufen:

```
<html xmlns="http://www.w3.org/1999/xhtml"
xmlns:f="http://xmlns.jcp.org/jsf/core"
xmlns:h="http://xmlns.jcp.org/jsf/html"
xmlns:p="http://primefaces.org/ui">
   <p:calendar value="#{person.geburtsdatum}" />
   […]
```

Wir können uns leider nicht alle Komponenten im Detail anschauen, sondern wollen uns auf die Funktionalitäten konzentrieren, welche einen zusätzlichen Mehrwert für moderne Webanwendungen bieten.

Clientseitige Validierung

JSF bietet standardmäßig nur eine serverseitige Validierung der Formulardaten an. Gerade in modernen Webanwendungen, sollen jedoch häufig die Antwortzeiten und die Kommunikation mit dem Webserver minimiert werden. Hierfür bietet PrimeFaces die Möglichkeit, Formatvalidierungen direkt auf dem Client auszuführen.

Um das innerhalb von PrimeFaces zu realisieren, wurde der CommandButton entsprechend erweitert. Dieser besitzt unter anderem zwei wichtige Attribute. Durch das Attribut `validateClient` wird die clientseitige Validierung aktiviert und innerhalb des Attributes `update` werden alle Komponenten angegeben, die während des Ajax-Request neu geladen werden sollen.

In dem Beispiel in Abbildung 1 werden verschiedene Möglichkeiten dargestellt, wie ein Formular clientseitig validiert werden kann. Die vier CommandButtons lösen jeweils eine andere Art der Validierung aus. Dabei ist es möglich, neben der reinen clientseitigen Validierung auch partielle Validierungen für einzelne Komponenten auszulösen. Das wird durch einen AJAX-Request realisiert. Natürlich ist es auch weiterhin möglich, die Validierung vollständig auf dem Server ausführen zu lassen.

```
<h:body>
   <h:form >
      <p:messages autoUpdate="true" />
      <h:panelGrid id="grid" columns="3" >
         <h:outputLabel for="einkommen"
value="Einkommen:" />
         <p:inputText id="einkommen"
value="#{validationBean.einkommen}" label="Einkommen">
            <f:validateDoubleRange minimum="4" maxi-
mum="6" />
         </p:inputText>
         <p:message for="einkommen" />
      </h:panelGrid>
      <p:commandButton value="Clientseitige Validation"
ajax="false" validateClient="true"  />
      <p:commandButton value="Ajax" update="grid"
validateClient="true" />
      <p:commandButton value="Partielle Validation"
update="grid" process="einkommen" alidateClient="true"
/>
      <p:commandButton value="Disabled" ajax="false"  />
   </h:form>
</h:body>
```

Abb. 1: Möglichkeiten der clientseitigen Validation von Formulardaten

```
@PushEndpoint("/infoMeldung")
public class MeldungResource{
   @OnMessage(encoders = {JSONEncoder.class})
   public String onMessage(String meldung) {
   return meldung;
   }
}
```

Abb. 2: Definition eines Endpoints für die Verarbeitung von Push-Nachrichten

```
<h:body>
      <h:form>
         <p:panelGrid columns="2">
            <h:outputText styleClass="ausgabe" />
            <p:commandButton value="Click"
actionListener="#{meldungView.doSomething}" />
         </p:panelGrid>
      </h:form>
      <p:socket onMessage="handleMessage" channel="/in-
foMeldung" />
      <script type="text/javascript">
         function handleMessage(data) {
            $('.ausgabe').html(data);
         }
      </script>
   </h:body>
```

Abb. 3: Verarbeitung und Ausgabe der Nachrichten, die als Push-Nachricht an den Client gesendet wurden

Links

[1] JSR 127 JavaServer Faces 1.0:
https://www.jcp.org/en/jsr/detail?id=127

[2] Webseite des Projekts PrimeFaces:
http://www.primefaces.org/

[3] Webseite des Projekts Atmosphere:
https://github.com/Atmosphere/

[4] PrimeFaces Dokumentation:
http://www.primefaces.org/documentation

[5] Beschreibung von Grid CSS:
http://blog.primefaces.org/?p=3248

[6] Übersicht aller kommerziellen und freien Themes und Layouts, die für PrimeFace angeboten werden:
http://www.primefaces.org/themes

Darüber hinaus besteht die Möglichkeit eigene clientseitige Validatoren zu erstellen. Jedoch müssen diese mit JavaScript implementiert werden und können demnach nicht in der Businesslogik wiederverwendet werden.

Support mobiler Endgeräte

PrimeFaces bietet unter dem Namen „PrimeFaces Mobile“ eine Vielzahl von Komponenten an, die speziell für mobile Geräte optimiert wurden. Auch diese Komponenten lassen sich leicht in eine bestehende Anwendung integrieren. Für diese Komponenten wurde ein zusätzlicher Namespace „http://primefaces.org/mobile“ reserviert, über den sich eine Vielzahl von Komponenten ansprechen lassen.

Neben dem neuen Namespace wird dem Framework noch mitgeteilt, dass für Seiten auf mobilen Endgeräten ein spezielles Renderkit mit dem Namen **`PRIMEFACES_MOBILE`** Verwendung findet. Das kann man entweder direkt in der View durch das View-Element angeben:

```
<f:view renderKitId="PRIMEFACES_MOBILE" />
```

Wahlweise kann das Renderkit auch direkt in der **`faces-config.xml`** angegeben werden:

```
<application>
   <default-render-kit-id>PRIMEFACES_MOBILE</default-render-kit-id>
 </application>
```

Nun können die Elemente wie gewohnt verwendet werden.

Bei den Komponenten wurde sehr darauf geachtet, möglichst viel Logik auf den Client zu verlagern, da gerade bei mobilen Anwendungen der Traffic zum Webserver sehr teuer ist. Jedoch wird der allgemeine Traffic bei PrimeFaces-Anwendungen immer etwas höher sein, als bei Webanwendungen, die mit einem reinen JavaScript-Framework erstellt worden sind. Dieses sollte bei der Auswahl des passenden Webframeworks natürlich nicht unberücksichtigt bleiben. Jedoch wird man bei kaum einem anderen Webframework so schnell zu einem guten Ergebnis kommen wie mit JSF und PrimeFaces.

Push-Benachrichtigungen

Die aktive Benachrichtigung eines Clients vom Server aus (auch Push-Benachrichtigung genannt), wird bei modernen Webanwendungen immer wichtiger, da ein kostenintensives Polling (zyklisches Abholen von Daten) gerade in mobilen Webanwendungen oder Seiten mit sehr hohen Zugriffszahlen weitestgehend vermieden werden soll.

Hierfür biete PrimeFaces eine eigene Lösung an, die auf dem Framework Atmosphere [3] basiert und bei der die Kommunikation über Websockets realisiert wird. Über die zugrundeliegende Implementierung muss sich jedoch wenig Gedanken gemacht werden, da PrimeFaces von dieser abstrahiert und eine einfache Bibliothek zur Verfügung stellt, um mit dieser Technologie zu arbeiten.

Für die Verwendung des Push-Service müssen zunächst, wie in der Dokumentation beschrieben [4], die zugehörigen Maven-Dependencies angegeben und das entsprechende Push-Servlet in der **`web.xml`** registriert werden. Im Anschluss kann die Funktionalität relativ einfach verwendet werden. Zunächst wird ein sogenannter Endpoint definiert. Dieser Endpoint kommuniziert mit der zugehörigen View und konvertiert die jeweiligen Nachrichten in das gewünschte Zielformat (z. B. JSON). In Abbildung 2 wird ein Endpoint mit dem Namen „infoMeldung“ erstellt, der die jeweiligen Daten als JSON-Nachricht an den Client überträgt. Innerhalb der Geschäftslogik wird nun wie folgt eine Nachricht an den zuvor erstellten Endpoint gesendet:

```
EventBus eventBus = EventBusFactory.getDefault().eventBus();
eventBus.publish("/infoMeldung", businessCode.getMeldung());
```

Nun fehlt lediglich noch die Logik in der Webseite, welche die Nachricht empfängt und verarbeitet. Hierfür muss die zugehörige JavaScript-Methode handleMessage implementiert werden, welche dafür sorgt, dass die Nachricht empfangen und an die richtige Stelle der Seite publiziert wird. In Abbildung 3 wurde diese Methode exemplarisch implementiert und die erhaltene Nachricht in das Feld mit der CSS-Klasse **`.ausgabe`** geschrieben.

Natürlich werden noch weitreichendere Möglichkeiten für das Versenden von Push-Nachrichten angeboten, die es dadurch zu einem mächtigen Werkzeug werden lassen.

Themes/Layouts

Responsive Design ist aus modernen Webanwendungen kaum noch wegzudenken. Gerade in einem komponentenorientierten Framework wie JSF ist es wichtig, dass die komplexen Komponenten ein responsives Design ermöglichen.

In Kombination mit Grid CSS [6] bietet PrimeFaces sogar ein leichtgewichtiges Layout-Werkzeug an, mit dem sich mit einfachen Mitteln ein einfaches responsives Grundlayout erstellen lässt.

Es werden aber auch eine Reihe kostenpflichtiger Designs und Layouts angeboten. Diese sollten immer in Erwägung gezogen werden, falls keine harten Vorgaben bezüglich des Corporate Designs einer Anwendung exisitieren und die Webanwendung ohne viel Aufwand über ein professionelles Layout und Design verfügen soll [5].

Über den Tellerrand

Selbst PrimeFaces verschließt sich nicht dem Trend der JavaScript-Frameworks und arbeitet parallel an einer eigenen Komponentenbibliothek für AngularJS2 mit dem Namen „PrimeNG“.

Auch eine reine JavaScript-Komponentenbibliothek auf Basis von jQuery gehört zu dem Portfolio. Wer also nicht zwingend JSF einsetzen kann, findet hier sicher einen guten Anlaufpunkt, um seine Anwendung mit fertigen Komponenten zu erweitern.

Bildnachweis

© istockphoto.com | Varijanta | Web-design-Entwicklung_Konzept

Fazit

Leider ist es in einem Artikel wie diesem nicht möglich, alle Funktionalitäten eines so mächtigen Frameworks aufzuzeigen. Dennoch sollte durch die vorgestellten Funktionalitäten deutlich geworden sein, dass sich auch mit JSF und PrimeFaces hochmoderne Webanwendungen entwickeln lassen. Dadruch stellt JSF mit PrimeFaces auch noch heute eine ernstzunehmende Option für die Entwicklung von Webanwendungen dar. Natürlich kann es immer Anforderungen geben, bei denen reine JavaScript-Frameworks zu bevorzugen sind. Jeder, dem die Auswahl der zur Verfügung gestellten Komponenten und Features ausreicht, findet mit JSF und PrimeFaces eine Kombination, die es sehr schnell ermöglicht, moderne Webanwendungen zu erstellen, ohne dabei Expertenwissen in JavaScript zu benötigen.

Auch wenn wir Ihnen in diesem Artikel nicht alle Funktionen von PrimeFaces vorstellen konnten, hoffen wir Ihr Interesse geweckt zu haben. Sollte das der Fall sein, können Sie innerhalb unseres Seminars „Rich Internet Applications mit JSF und PrimeFaces“ an praktischen Übungen erlernen, wie man mit JSF und PrimeFaces moderne Webanwendungen erstellt und die Vielzahl an Möglichkeiten effizient einsetzt.

Christian Wiesing
(info@ordix.de)

ORDIX® seminare

Seminarzentrum:
ORDIX AG
Kreuzberger Ring 13
65205 Wiesbaden
Tel.: 0611 77840-00

Online-Anmeldung, aktuelle Seminarinhalte und Termine unter:
https://seminare.ordix.de

Unser Seminarstandort ist Wiesbaden.

Die hier angegebenen Termine entsprechen dem ersten Seminartag.

Die Preise gelten pro Seminar pro Teilnehmer in Euro zzgl. ges. MwSt., Inhouse-Preise auf Anfrage.

Big Data und Data Warehouse

BIG Data				
DB-BIG-01	Big Data: Informationen neu gelebt	1 Tag	590,00 €	06.03. | 26.06. | 11.09. | 20.11.
DB-BIG-02	Big Data: Apache Hadoop Grundlagen	3 Tage	1.290,00 €	20.03. | 03.07. | 18.09. | 04.12.
Data Warehouse				
DB-DB-03	Data Warehouse Grundlagen	3 Tage	1.290,00 €	07.03. | 27.06. | 12.09. | 21.11.
DB-NSQL-01	Einführung in NoSQL-Datenbanken	2 Tage	1.090,00 €	23.03. | 06.07. | 21.09. | 07.12.

PostgreSQL

DB-PG-01	PostgreSQL Administration	5 Tag	2.150,00 €	23.01. | 20.03. | 15.05. | 28.08. | 20.11.

Oracle

Entwicklung				
DB-ORA-01	Oracle SQL	5 Tage	1.890,00 €	30.01. | 24.04. | 26.06. | 11.09. | 06.11.
DB-ORA-01A	Oracle SQL Power Workshop	3 Tage	1.290,00 €	23.01. | 03.05. | 21.08. | 27.11.
DB-ORA-02	Oracle Datenbankprogrammierung mit PL/SQL Grundlagen	5 Tage	1.890,00 €	13.02. | 08.05. | 10.07. | 25.09. | 13.11.
DB-ORA-34	Oracle Datenbankprogrammierung mit PL/SQL Aufbau	3 Tage	1.290,00 €	27.02. | 29.05. | 24.07. | 04.10.
DB-ORA-42	Oracle PL/SQL für Experten - Performance Analyse & Laufzeitopt.	3 Tage	1.290,00 €	30.01. | 06.06. | 09.10.
DB-ORA-53	Oracle Text	3 Tage	1.390,00 €	10.04. | 29.05. | 25.09. | 04.12.
DB-ORA-51	Oracle Spatial	3 Tage	1.290,00 €	06.03. | 29.05. | 16.10. | 04.12.
DB-ORA-46	Oracle APEX Anwendungsentwicklung Grundlagen	3 Tage	1.290,00 €	06.02. | 03.05. | 17.07. | 11.09. | 06.11.
DB-ORA-47	Oracle APEX Anwendungsentwicklung Aufbau	3 Tage	1.290,00 €	20.03. | 07.06. | 25.09. | 27.11.
Administration				
DB-ORA-03	Oracle Datenbankadministration Grundlagen	5 Tage	1.990,00 €	23.01. | 20.03. | 19.06. | 04.09. | 06.11.
DB-ORA-04	Oracle Datenbankadministration Aufbau	5 Tage	1.990,00 €	06.02. | 03.04. | 17.07. | 23.10.
DB-ORA-07	Oracle Tuning - Theorie und Interpretation von Reports	5 Tage	2.290,00 €	20.02. | 26.06. | 11.09. | 20.11.
DB-ORA-11	Oracle Troubleshooting Workshop	3 Tage	1.390,00 €	10.04. | 04.10.
DB-ORA-08	Oracle 12c Real Application Cluster (RAC) und Grid Infrastructure	5 Tage	2.290,00 €	06.03. | 15.05. | 31.07. | 09.10. | 11.12.
DB-ORA-49	Oracle 12c Neuheiten	5 Tage	2.090,00 €	13.03. | 03.07. | 25.09. | 04.12.
DB-ORA-52W	Oracle Lizenz Workshop Webinar	1 Tage	590,00 €	Termine auf Anfrage
DB-ORA-33	Oracle Security	3 Tage	1.290,00 €	16.01. | 18.04. | 14.08. | 16.10.
DB-ORA-35	Oracle Cloud Control	3 Tage	1.290,00 €	13.02. | 03.04. | 24.07. | 04.10. | 04.12.
DB-ORA-48	Oracle Golden Gate	3 Tage	1.290,00 €	27.02. | 03.05. | 31.07. | 23.10.
Backup und Recovery				
DB-ORA-32	Oracle Backup und Recovery mit RMAN	5 Tage	1.990,00 €	20.02. | 08.05. | 07.08. | 27.11.
DB-ORA-31	Oracle Data Guard	4 Tage	1.690,00 €	16.01. | 24.04. | 28.08. | 27.11.
MySQL				
DB-MY-01	MySQL Administration	3 Tage	1.290,00 €	30.01. | 27.03. | 17.07. | 25.09. | 27.11.

IBM Datenbanksysteme

Informix				
DB-INF-01	IBM Informix SQL	5 Tage	1.790,00 €	27.02. | 10.07. | 23.10.
DB-INF-02	IBM Informix Administration	5 Tage	1.990,00 €	13.03. | 24.07. | 13.11.
DB2				
DB-DB2-01	IBM DB2 für Linux/Unix/Windows SQL Grundlagen	5 Tage	1.890,00 €	27.02. | 15.05. | 31.07. | 16.10.
DB-DB2-02	IBM DB2 für Linux/Unix/Windows Administration	5 Tage	1.990,00 €	13.02. | 19.06. | 28.08. | 27.11.
DB-DB2-05	IBM DB2 für Linux/Unix/Windows Monitoring und Tuning	3 Tage	1.290,00 €	10.04. | 31.07. | 20.11.
DB-DB2-06	IBM DB2 für Linux/Unix/Windows Backup und Hochverfügbarkeit mit HADR	3 Tage	1.390,00 €	19.04. | 03.07. | 04.10.

Microsoft

Entwicklung				
MS-SQL-01	Querying Data with Transact-SQL	5 Tage	1.990,00 €	06.03. | 10.07. | 18.09. | 06.11.
MS-SQL-07	Updating Your Skills to Microsoft SQL Server 2016	5 Tage	1.990,00 €	13.03. | 19.06. | 21.08. | 23.10.
Administration				
MS-SQL-02	Administering a SQL Database Infrastructure	5 Tage	1.990,00 €	27.03. | 17.07. | 09.10. | 27.11.
MS-SQL-05	Implementing a SQL Data Warehouse	5 Tage	1.990,00 €	30.01. | 15.05. | 28.08. | 13.11.
MS-SQL-11	Microsoft SQL Server for Oracle DBAs	4 Tage	1.790,00 €	13.02. | 08.05. | 14.08. | 16.10.
MS-SQL-05W	Microsoft SQL Server 2016 Upgrade Webinar	1 Tag	99,00 €	13.01.

Storage

STORAGE01	Storage Grundlage	2 Tage	1.190,00 €	09.02. | 18.05. | 24.08. | 26.10.
PERFMESS01	Performance Messungen unter Unix und Windows	2 Tage	1.190,00 €	19.01. | 16.03. | 22.06. | 14.09. | 23.11.
PERFGUARD1	PerformanceGuard Grundlagen	3 Tage	1.950,00 €	03.04. | 18.09.
PERFGUARD2	PerformanceGuard Aufbau	2 Tage	1.590,00 €	06.04. | 21.09.
NETAPP01	Backup und Recovery Oracle on Netapp	5 Tage	1.950,00 €	06.02. | 24.04. | 14.08. | 23.10. | 04.12.

Web und Application-Server

INT-04	Apache HTTP Server Administration	3 Tage	1.190,00 €	06.02. | 29.05. | 07.08. | 04.10. | 11.12.
INT-07	Tomcat Konfiguration und Administration	3 Tage	1.290,00 €	13.02. | 03.05. | 24.07. | 18.09. | 20.11.
INT-08	WebSphere Application Server Installation und Administration	3 Tage	1.390,00 €	06.03. | 29.05. | 04.09. | 04.12.
INT-12	WildFly Application Server Administration	3 Tage	1.290,00 €	16.01. | 03.04. | 10.07. | 11.09. | 13.11.
INT-11-7	JBoss 7 Administration und Konfiguration	3 Tage	1.290,00 €	27.03. | 03.07. | 25.09. | 27.11.
DB-ORA-50	Oracle WebLogic Administration Grundlagen	3 Tage	1.390,00 €	23.01. | 10.04. | 17.07. | 16.10.

Projekt- und IT-Management

Klassisches Projektmanagement				
PM-01	IT-Projektmanagement - Methoden und Techniken	3 Tage	1.690,00 €	06.02. | 24.04. | 03.07. | 20.09. | 22.11.
PRINCE-01	PRINCE2® Foundation	3 Tage	1.225,00 €	13.02. | 24.04. | 26.06. | 21.08. | 16.10. | 11.12.
PRINCE-02	PRINCE2® Practitioner	3 Tage	1.560,00 €	15.02. | 26.04. | 28.06. | 23.08. | 18.10. | 13.12.
PRINCE-03	PRINCE2® kompakt	5 Tage	2.595,00 €	13.02. | 24.04. | 26.06. | 21.08. | 16.10. | 11.12.
PM-06	Projekte souverän führen - Systemisches Projektmanagement	4 Tage	1.850,00 €	13.02. | 08.05. | 04.09. | 06.11.
PM-05	Projektcontrolling in der IT	2 Tage	1.090,00 €	27.03. | 22.06. | 18.09. | 30.11.
PM-07	Krisenmanagement in Projekten - Projektkrisen meistern	2 Tage	1.100,00 €	02.02. | 10.08. | 26.10.
PM-14	Anforderungsmanagement in IT-Projekten	2 Tage	1.090,00 €	13.03. | 08.06. | 28.09. | 16.11.
PM-T-01	Testmanagement Grundlagen	2 Tage	1.190,00 €	15.03. | 06.06. | 17.08. | 26.10.
Agiles Projektmanagement				
PM-08	Agiles Projektmanagement mit Scrum	2 Tage	1.190,00 €	13.03. | 26.06. | 28.08. | 20.11.
PM-08-Z	Scrum Praxis und Zertifizierung	1 Tage	690,00 €	15.03. | 28.06. | 30.08. | 22.11.
PM-15	Leading Excellence for Scrum Master	3 Tage	1.340,00 €	06.03. | 06.06. | 01.11.
PM-24	KANBAN in der IT	2 Tage	1.190,00 €	16.03. | 29.06. | 31.08. | 23.11.
IT-Management, IT-Strategie und IT-Organisation				
IT-SEC-01	IT-Sicherheit für Projektmanager und IT-Leiter	5 Tage	2.700,00 €	20.02. | 15.05. | 21.08. | 23.10.
PM-29	Systemische Führung	3 Tage	1.650,00 €	15.05. | 13.11.
MGM-07	IT-Strategie - strategische IT-Planungen	3 Tage	1.650,00 €	23.01. | 07.08. | 23.10.
MGM-02	IT-Architekturen	3 Tage	1.590,00 €	02.08. | 30.10.
PM-10	IT-Controlling	3 Tage	1.590,00 €	20.02. | 05.04. | 14.08. | 11.12.
MGM-04	Geschäftsprozessmanagement (BPM)	3 Tage	1.590,00 €	01.03. | 19.06. | 25.09. | 04.12.
ITIL-01	ITIL® V3 Foundation	3 Tage	922,50 €	06.02. | 03.04. | 19.06. | 07.08. | 09.10. | 04.12.
ITIL-02	ITIL® V3 Practitioner	3 Tage	1.380,00 €	09.02. | 06.04. | 22.06. | 10.08. | 12.10. | 07.12.
ITIL-03	ITIL® V3 kompakt	3 Tage	2.250,00 €	06.02. | 03.04. | 19.06. | 07.08. | 09.10. | 04.12.
PM-28	IT-Organisation	3 Tage	1.650,00 €	30.01. | 10.07. | 16.10.
Kommunikation und Selbstmanagement				
PM-17	IT-Management Soft Skills	3 Tage	1.650,00 €	20.02. | 10.04. | 09.10.
PM-27	Social Skills für IT-Consultants	3 Tage	1.650,00 €	16.01. | 18.04. | 04.10.
PM-16	Kommunikation in Projekten - Power Workshop	2 Tage	1.090,00 €	09.02. | 04.05. | 07.09. | 16.11.
PM-11	Konfliktmanagement	2 Tage	1.100,00 €	19.01. | 13.07. | 19.10.
PM-19	Zeit- und Selbstmanagement	2 Tage	1.090,00 €	09.03. | 18.05. | 12.10.
PM-30	Verhandlungstechniken	2 Tage	1.100,00 €	27.04. | 07.12.

Betriebssysteme & Monitoring

Unix/Linux				
BS-01	Unix/Linux Grundlagen für Einsteiger	5 Tage	1.690,00 €	16.01. | 20.03. | 19.06. | 18.09. | 13.11.
BS-02	Linux Systemadministration	5 Tage	1.690,00 €	30.01. | 03.04. | 03.07. | 09.10. | 27.11.
BS-25	Unix Power Workshop für den Datenbank- & Applikationsbetrieb	5 Tage	1.890,00 €	23.01. | 08.05. | 28.08. | 13.11.
BS-27	Neuerungen SUSE Linux Enterprise Server 12	3 Tage	1.290,00 €	13.02. | 03.05. | 14.08. | 23.10.
BS-26	Einführung in die Administration von OpenStack	5 Tage	2.990,00 €	27.03. | 22.05. | 04.09. | 06.11.
BS-09	Linux Hochverfügbarkeits-Cluster	5 Tage	1.890,00 €	06.02. | 15.05. | 21.08. | 06.11.
Solaris				
BS-03-11	Solaris 11 Systemadministration Grundlagen	5 Tage	1.990,00 €	20.02. | 26.06. | 25.09. | 20.11.
BS-04-11	Solaris 11 Systemadministration Aufbau	5 Tage	1.990,00 €	06.03. | 17.07. | 11.12.
BS-06-11	Solaris 11 für erfahrene Unix/Linux-Umsteiger	5 Tage	1.990,00 €	27.02. | 24.07. | 11.09. | 27.11.
BS-24	Solaris 11 Administration Neuheiten	3 Tage	1.290,00 €	13.03. | 19.06. | 04.09. | 09.10.
BS-18	Solaris Virtualisierung mit ZFS und Container (Zonen)	5 Tage	1.990,00 €	24.04. | 07.08. | 04.12.
IBM AIX				
AIX-01	IBM AIX Systemadministration Grundlagen	5 Tage	1.990,00 €	13.02. | 26.06. | 11.09. | 06.11.
AIX-04	IBM AIX Systemadministration Power Workshop	3 Tage	1.290,00 €	10.04. | 10.07. | 04.10. | 11.12.
AIX-02	IBM AIX Installation, Backup und Recovery mit NIM	3 Tage	1.290,00 €	19.04. | 07.06. | 25.09. | 13.11.
Microsoft				
MS-W-10-01	Windows 10 für Administratoren	3 Tage	1.650,00 €	20.02. | 22.05. | 18.09. | 13.11.
MS-HV-03	Microsoft Hyper-V-Workshop unter Server 2016 (& frühere Versionen)	2 Tage	1.150,00 €	06.02. | 08.05. | 04.09. | 06.11.
MS-HV-01	Microsoft Hyper-V Deep-Dive unter Server 2016 (& frühere Versionen)	3 Tage	1.650,00 €	08.02. | 10.05. | 06.09. | 08.11.
Monitoring				
SM-NAG-01	Systemüberwachung mit Nagios - Workshop	3 Tage	1.190,00 €	30.01. | 03.05. | 28.08. | 23.10.

Entwicklung

Allgemeines				
OO-01	Einführung in die objektorientierte Programmierung und UML	3 Tage	1.190,00 €	16.01. | 03.04. | 03.07. | 18.09. | 20.11.
E-SWA-01	Softwarearchitekturen	5 Tage	1.890,00 €	23.01. | 24.04. | 24.07. | 13.11.
Script-Sprachen				
P-PERL-01	Perl Programmierung Grundlagen	5 Tage	1.690,00 €	20.03. | 19.06. | 18.09. | 20.11.
P-PERL-02	Perl Programmierung Aufbau	5 Tage	1.690,00 €	27.03. | 03.07. | 04.09. | 11.12.
P-UNIX-01	Shell, Awk und Sed	5 Tage	1.690,00 €	27.02. | 08.05. | 17.07. | 11.09. | 13.11.
XML				
P-XML-01	XML Grundlagen	3 Tage	1.190,00 €	27.03. | 29.05. | 07.08. | 06.11.
Java				
P-JAVA-01	Java Programmierung Grundlagen	5 Tage	1.690,00 €	30.01. | 08.05. | 17.07. | 09.10. | 04.12.
P-JAVA-03	Java Programmierung Aufbau	5 Tage	1.690,00 €	20.02. | 15.05. | 14.08. | 27.11.
P-JEE-08	Java Performance Tuning	3 Tage	1.290,00 €	03.04. | 10.07. | 11.09. | 06.11.
P-JAVA-11	Java 8 Neuheiten	2 Tage	990,00 €	09.02. | 04.05. | 03.08. | 26.10.
Java EE				
P-JAVA-12	Java EE Power Workshop	5 Tage	1.890,00 €	30.01. | 27.03. | 10.07. | 20.11.
J-HIB-01	Java Persistence API mit Hibernate	5 Tage	1.690,00 €	13.02. | 29.05. | 21.08. | 16.10.
INT-05	Java Web Services	3 Tage	1.190,00 €	27.02. | 29.05. | 18.09. | 27.11.
P-JEE-06	Spring Power Workshop	5 Tage	1.590,00 €	16.01. | 24.04. | 24.07. | 09.10.
Web- und GUI-Entwicklung				
P-JAVA-15	Einstieg in JavaFX	3 Tage	1.390,00 €	06.02. | 19.04. | 31.07. | 23.10.
P-PHP-01	PHP Programmierung	5 Tage	1.690,00 €	20.02. | 08.05. | 07.08. | 16.10.
P-JEE-05	Rich Internet Application mit JSF und Primefaces	5 Tage	1.590,00 €	06.03. | 26.06. | 28.08. | 13.11.
P-JEE-05A	Webanwendungen mit JavaServer Faces (JSF)	5 Tage	1.590,00 €	20.03. | 19.06. | 04.09. | 11.12.
Tools und Verfahren				
P-CI-01	Continuous Integration (CI) Workshop	3 Tage	1.190,00 €	23.01. | 19.04. | 03.07. | 25.09. | 11.12.

IT-Security – Worauf es ankommt (Teil II):

Sichere Webanwendungen

Dieser Artikel führt die Reihe von Artikeln zum Thema IT-Security fort, die mit der Ausgabe 3/2015 begann. Er konkretisiert die Anforderungen und Vorgehensalternativen für die Realisierung von sicheren Webanwendungen, basierend auf der Definition von Sicherheit in dem vorangegangenen Artikel. Es folgen Hinweise zur Verifikation des erreichten Sicherheitslevels einer Anwendung. Abschließend werden fortlaufende Aktivitäten beschrieben, um mögliche neu aufgedeckte Sicherheitslücken über Korrekturen in einer aktiven Anwendung zu beheben. Ein weiteres Ziel des Artikels ist es, auf die wichtigen Informationen im Internet zu verweisen und Anregungen zu geben, wie bereits mit einfachen Maßnahmen die schwerwiegendsten Sicherheitsrisiken eliminiert werden können.

Realisierung sicherer Webanwendungen

Aktueller Sicherheitsstatus von Webanwendungen

Der aktuelle Status weltweit – bezogen auf die Sicherheitslücken in Webanwendungen – ist durch einige gemeinnützige Organisationen sehr transparent. Sie bieten sehr detaillierte Informationen, wie:

- Hitlisten aktueller Lücken gewichtet nach Häufigkeit und Auswirkung
- technische Zusammenhänge dieser kriminellen Nutzungen
- Auswirkung und Gefährdungspotenzial
- konkrete Lösungen oder Alternativen

Hervorzuhebende Informationsquellen zu bekannten Sicherheitslücken sind:

- Bundesamt für Sicherheit [1]
- Common Weakness Enumeration, betrieben von der MITRE, gesponsert durch die Regierung der Vereinigten Staaten [3]
- Common Attack Pattern Enumeration and Classification, ebenfalls betrieben von der MITRE [6]
- OWASP: Open Web Application Security Project [4]

Alle Organisationen pflegen unabhängig voneinander gewichtete Hitlisten von den am häufigsten vorkommenden Sicherheitslücken unter Berücksichtigung ihres Gefährdungspotenzials. Inhaltlich unterscheiden sich diese Hitlisten der einzelnen Organisationen kaum.

Interessanterweise, ja sogar erschreckenderweise, gibt es in diesen Hitlisten über einen längeren Zeitraum kaum Veränderungen. Die Naivität, mit der weltweit viele Webanwendungen realisiert sind, ist beängstigend und gefährlich. Und das, obwohl das Wissen um die Existenz der Gefährdungen und der Vermeidungsmöglichkeit im Detail bekannt und jedem über das Internet verfügbar ist.

Einige exzellente Referenzen zur Realisierung von sicheren Webanwendungen gilt es hervorzuheben. Das Bundesamt für Sicherheit in der Informationstechnik [1] publizierte bereits 2006 einen allgemein zugänglichen Ratgeber mit dem Titel „Sicherheit von Webanwendungen; Maßnahmenkatalog und Best Practices“. Obwohl bereits 10 Jahre alt, ist es leider nach wie vor eine Pflichtlektüre für jeden, der an der Konzeption und Implementierung von Webanwendungen beteiligt ist. 2013 folgten die Publikationen „Leitfaden zur Entwicklung sicherer Webanwendungen“ – jeweils eine dedizierte Ausgabe für Anforderungen an die Auftragnehmer und für Auftraggeber aus der öffentlichen Verwaltung. Hier werden der komplette Entwicklungsprozess zur Realisierung und die Inbetriebnahme von sicheren Webanwendungen dargestellt. Hinzu kommen konkrete Hilfestellungen bei der Verwendung unterschiedlicher Technologien. Besonders hervorzuheben sind die Hinweise zu „Low Hanging Fruits“; Hinweise, wie man mit wenig Aufwand viele schwerwiegende Sicherheitslücken schließen oder vermeiden kann.

Eigentlich könnte der Artikel hier enden. Inhaltlich kann dieser nicht mit den BSI-Veröffentlichungen konkurrieren, die umfassend und leicht verständlich sind. Trotzdem versuchen wir mit diesem Artikel „bottom up“ das Verständnis zu wecken, wie über Beachtung einiger weniger Grundprinzipien oder Prämissen viele kritische Sicherheitslücken zu vermeiden sind.

Für jede Webanwendung gilt:

- Alle Eingabedaten an eine Webanwendung sind als gefährlich einzuschätzen und müssen explizit validiert werden, bevor sie zur Anwendung kommen.
- Alle Daten, die an den Browser eines Anwenders zurückgegeben werden, müssen explizit validiert werden.
- Kritische Informationen dürfen auch innerhalb der Webanwendung nur verschlüsselt genutzt werden.

Für den verantwortlichen Lieferanten der Anwendung gilt:

- Sicherheit ist keine statische Größe, sie ist ein bewegliches Ziel.
- Bisher nicht bekannte Sicherheitslücken können jederzeit aufgedeckt werden.
- Es bedarf der fortlaufenden Beobachtung der Publizierung von neu aufgedeckten Sicherheitslücken.
- Aufgedeckte, relevante Sicherheistlücken müssen schnell und sicher behoben werden.

Validierung von Eingabedaten

Prüfen vor der Übernahme als Operant oder Operator

Relevanz:

- Top 1: „SQL Injections“ [2]
- Top 2: „OS command injection“ [2]
- Top 13: „Pathname to a Restricted Directory ('Path Traversal')“ [2]
- Suche nach „sql injection“ aufgedeckt in den letzten 3 Monaten: 28 [16]
- Suche nach „command injection“ aufgedeckt in den letzten 3 Monaten: 2 [16]

Die spezifischen SQL-Begrenzungszeichen, wie das Anführungszeichen oder Apostroph, in Usernamen oder Passwörtern führen zu dem Problem. Für das Top-2-Problem sind es „..“, Schrägstrich und das Semikolon. Deswegen ist es zwingend notwendig, Eingabedaten auf sinnvolle und erlaubte Zeichen hin zu überprüfen.

Das BSI [1] empfiehlt ein positives Sicherheitsmodel, d. h. die Prüfung einer Eingabe über eine kontextspezifische „whitelist“. Der alternative Ansatz „blacklist“ prüft nur auf illegale Zeichen. Dieser Ansatz ist zwar einfacher zu realisieren, aber ein „vergessenes“ Zeichen öffnet eine Lücke. Inzwischen bieten Programmierumgebungen wie z. B. JAVA oder PHP Funktionen zur Validierung an. OWASP versucht über das Projekt ESAPI (Enterprise Security API) einen Standard zu schaffen, aber scheinbar stagniert der Fortschritt hier seit 2013.

Prüfen des Inhalts von Dateien nach „upload“

Relevanz:

- Top 9: „Unrestricted Upload of File with Dangerous Type“ [2]
- Suche nach „upload“ aufgedeckt in den letzten 3 Monaten: 14 [16]

Die Validierung von Eingabedaten schließt den Inhalt von Datei-Uploads mit ein. Die alleinige Prüfung auf den Dateityp über die Endung des Dateinamens reicht nicht. Ansonsten kann ein Anwender unter falscher Namensendung Programmlogik zum Beispiel als Bilddatei tarnen.

Prüfen der Länge

Relevanz

- Top 3: „Classic Buffer Overflow“ [2]
- Top 20: „Incorrect Calculation of Buffer Size“[2]
- Top 24: „Integer Overflow or Wraparound“ [2]
- Suche nach „buffer overflow“ aufgedeckt in den letzten 3 Monaten: 87 [16]
- Suche nach „integer overflow“ aufgedeckt in den letzten 3 Monaten: 38 [16]

Benutzereingaben müssen zusätzlich zu der Überprüfung auf sinnvollen Inhalt auch auf ihre maximal sinnvolle Länge hin überprüft werden. Es vermeidet Speicher-

überschreitungen (Buffer Overflows), wenn die Eingabe längenlimitiert ist. Geschieht dies über das Web unkontrolliert und wiederholt, reicht es für eine „DoS"-Attacke, da die Fehlerbehandlung erfahrungsgemäß eine höhere CPU-Last oder höheren Speicherbedarf durch die Webanwendung verursacht.

Erfolgen das Überschreiten der Puffergrenzen durch den Anwender gezielt unter Kenntnis der Umgebung der Webanwendung, kann er in den Programmablauf der Web-Applikation eingreifen und damit sogar gezielte Operationen wie „create executable file" z.B. über die Systembibliotheken, die ihn die Webanwendung eingebunden sind, ausführen.

Das Problem sind typische Anwendungen, realisiert in der Programmiersprache „C" oder „C++", in der viele Standard-Webkomponenten der Apache Foundation realisiert sind. Zur Vermeidung des Problems kann man verbesserte Libraries einsetzen, die bei Kopierfunktionen implizit gegen Puffergrenzen prüfen. Einige Compiler bieten optional die automatische Generierung von Prüfungen an (Beispiel: Microsoft Visual Studio/GS). Andere Lösungen versuchen durch „Address Space Layout Randomization" (ASLR) das Einbringen von Schad-Code zu bekannten, festen Speicheradressen zu verhindern. Je nach CPU-Typ des Servers kann über das Betriebssystem die Ausführung von Programmcode in Datenbereichen blockiert werden (NX-Bit - Data Execution Protection).

Zusätzlich zur Prüfung auf Puffergrenzen müssen Eingabedaten vor der Nutzung als Datentyp auf den Wertebereich hin geprüft werden, um ein „wraparound" zu vermeiden. Das Ergebnis nach einem „wraparound" wäre eine Zahl, die kleiner ist als notwendig. Tritt dies z. B. bei der Berechnung einer Puffergröße auf, erreicht der Angreifer trotz Längenprüfung in der Anwendung ein „Buffer Overflow".

Prüfen von Eingabetexten

Relevanz:

- Top 4: „XSS Cross-site Scripting" [2]
- Top 12: „CSRF Cross-Site Request Forgery" [2]
- Top 14: „Download of Code Without Integrity Check" [2]
- Top 16: „Inclusion of Functionality from Untrusted Control Sphere" [2]
- Top 22: „URL Redirection to Untrusted Site" [2]
- Suche nach „xss" aufgedeckt in den letzten 3 Monaten: 141 [16]
- Suche nach „forgery" aufgedeckt in den letzten 3 Monaten: 21 [16]

Mit diesen Attacken wird nicht eine Webanwendung selbst gefährdet, sondern sie wird missbraucht, um Schad-Code in die Systeme späterer Anwender einzuschleusen. Ein Angreifer versucht, Skript-Code in Texteingabefelder einzubringen in der Erwartung, dass dieser Text ungeprüft in dynamischen Webseiten wiedergegeben wird. Bei einem späteren Aufruf einer solchen Webseite wird versucht, den Skript-Code über den Browser eines Anwenders zur Ausführung zu bringen. Je nach lokaler Sicherheitseinstellung eines Anwendersystems ist es damit möglich, Nutzerdaten an Webseiten eines Angreifers zu senden oder von dort Schad-Code zu injizieren.

Aus diesem Grund ist es die Aufgabe der Webanwendung, die Verteilung von illegalem Skript-Code zu verhindern.

Die Angriffe sind in zwei Kategorien zu unterteilen:

- XSS: Hier versucht ein Angreifer, Skript-Code unterzubringen, der bei Abruf Schad-Code in das Zielsystem installiert.
- CSRF: Hier ist die Zielrichtung des Angriffs, die dynamischen Anwenderdaten aus „cookies" an eine andere kriminelle URL abzuliefern. Da in diesen „cookies" gewöhnlich dynamische Sitzungsdaten abgelegt werden, kann der Angreifer mit diesen Sitzungsdaten z. B. ohne Authorisierung über Namen und Passwort in die Sitzung eingreifen oder sie sogar übernehmen.

Verschlüsselung von kritischen Daten

Relevanz:

- Top 25: „Use of a One-Way Hash without a Salt" [2]
- Da ein unautorisierter Zugriff auf Daten der Webanwendung trotz vieler Maßnahmen nicht ausgeschlossen werden kann, ist es notwendig, kritische Informationen wie z. B. Passwörter, Kreditkartennummern usw. verschlüsselt und nicht im Klartext zu nutzen oder in der Datenbank zu speichern. Der Schutz der Verschlüsselung muss ausreichend hoch sein, entsprechend einem möglichen Schaden bei lesbarem Zugriff.
- Einfache kryptografische „hashes" sind nicht mehr ausreichend. Im Internet gibt es sogenannte „rainbow tables", die eine schnelle Suche nach dem Originalwert unterstützen. Durch Hinzunahmen eines zweiten Hashparameters, dem sogenannten "salt", erhöht sich der Aufwand für eine Entschlüsselung drastisch. „rainbow tables" für „hash" mit „salt" werden extrem lang und damit nicht mehr handhabbar. Je nach „hash"-Algorithmus ist der Schutz mehr oder weniger sicher bzw. es ist mehr oder weniger aufwendig, ein Passwort zu restaurieren. Stand der Technik ist zur Zeit der SHA-2-Algorythmus, da „md5" einfach zu entschlüsseln und SHA-0 kaum sicherer ist. SHA-1 bietet etwas Schutz.
- Da die gelisteten Algorithmen sehr schnell sind, kann man in einer kurzen Zeit auch mit vielen Versuchen zum Entschlüsseln rechnen. Durch eine Verlangsamung des Algorithmus erhöht man diese Zeit. Gerade bei Passwortprüfung in Anwenderdialogen sind langsamere Algorithmen wie die Funktion „bcrypt" kaum spürbar, erhöhen aber signifkant die Zeit beim automatisierten Entschlüsseln und damit die Sicherheit von verschlüsselten Passwörtern.

Verifizierung einer Webanwendung

Die Überprüfung der Qualitätsziele bezogen auf die Robustheit der Webanwendung gegenüber Sicherheitsattacken muss als fester Teil eines Qualitätszyklus etabliert

werden. Es gibt einige verfügbare Tools oder Hilfsmittel, die nutzbar sein könnten. Sicherheitstests werden fachsprachlich Penetrationstests genannt. Da es aber keine absolute Sicherheit gibt, muss der Aufwand kommerziell zu rechtfertigen sein. Es erscheint sinnvoll, den Aufwand und die Kosten in Relation zu einem möglichen Schaden zu stellen.

Die folgende Liste fasst wichtige Informationsquellen zusammen.

- BSI bietet Testchecklisten. [1]
- OWASP bietet Testdaten und Hinweise für die Durchführung von Tests. [12]
- OWASP stellt Test-Guidelines zur Verfügung. [13]
- BSI bietet einen Praxisleitfaden für die Durchführung von Penetrationstests. [15]
- „Kali“ stellt eine Linux-basierte „Penetration Test Umgebung“ zur Verfügung. [14]

Neben der reinen Testabdeckung ist es wichtig, in eine automatisierte Verifikation einer Webanwendung zu investieren. Nur so ist es realistisch, neu aufgedeckte Sicherheitslücken kurzfristig für Anwendungen operativ zu beheben.

Sicherheitsradar

Eine Anwendung ist so lange als „relativ sicher“ zu betrachten, bis eine neue, bisher ungekannte Sicherheitslücke aufgedeckt wird. Dabei sind nicht nur Sicherheitslücken in der eigenen Anwendung relevant, sondern auch die in den eingesetzten Technologien oder Plattformen. Deswegen ist es notwendig, regelmäßig aktuelle Informationen zu verfolgen, die

- neue Sicherheitslücken veröffentlichen,
- Korrektur-Patches für die verwendeten Technologien und Plattformen ankündigen.

Fazit

Webanwendungen signifikant sicherer zu machen, ist mit vertretbarem Aufwand durch die Vermeidung der typischen und häufigen Fehler möglich, die dieser Artikel skizziert. Je nach Kritikalität einer Anwendung reicht dies allerdings nicht. Hier helfen die gelisteten Internetreferenzen als Informationsquelle, um weitere Sicherheitslücken in dem eingesetzten Technology Stack der eigenen Webanwendung zu identifizieren und zu beseitigen. Zudem liefern diese Portale auch Anleitungen zur Behebung oder Umgehung von Sicherheitsproblemen. Regelmäßiges Prüfen auf neue entdeckte Sicherheitslücken ist notwendig, um einen erreichten Sicherheitslevel auch über die Zeit zu halten.

Der Artikel behandelt nicht die Aspekte einer sicheren Authentifizierung von Anwendern und der Sicherstellung der Datenintegrität und Geheimhaltung bei der Übertragung über das Internet. Dies bleibt einem nachfolgenden Artikel vorbehalten.

Links/Quellen

[1] [Q1 Bundesamtes für Sicherheit in der Informationstechnik: https://www.bsi.bund.de

[2] Common Weakness Enumeration Plattform US Home Land Security https://cwe.mitre.org/top25/index.html

[3] Common Attack Pattern Enumeration and Classification Plattform US Home Land Security http://capec.mitre.org/

[4] OWASP: Open Web Application Security Project https://www.owasp.org

[5] OWASP Top 10 – 2013: Die 10 häufigsten Sicherheitsrisiken für Webanwendungen; Deutsche Übersetzung Version 1.0 https://www.owasp.org/images/4/42/OWASP_Top_10_2013_DE_Version_1_0.pdf

[6] Übersicht über die aktuellen Cyberangriffe auf DTAG-Sensoren der Deutschen Telekom http://sicherheitstacho.eu/

[7] Öffentliches Forum/Info-Austausch aktuellster Sicherheitslücken http://seclists.org/fulldisclosure/

[8] CWE/SANS TOP 25 Most Dangerous Software Errors https://www.sans.org/top25-software-errors

[9] Angriffstechniken https://www.corelan.be/index.php/articles/

[10] CIS Critical Security Controls https://www.sans.org/critical-security-controls

[11] Lernumgebung zum Erkennen und Verifizieren bekannter Sicherheitslücken über eine beispielhafte unsichere JEE-basierte Webanwendung https://www.owasp.org/index.php/Category:OWASP_WebGoat_Project

[12] OWASP Cheat Sheets (Testdaten zur Verifikation gegenüber bekannten Lücken) https://www.owasp.org/index.php/Cheat_Sheets

[13] OWASP Testing Guide https://www.owasp.org/index.php/Category:OWASP_Testing_Project

[14] Penetration Testing Distribution https://www.kali.org/

[15] Ein Praxis-Leitfaden für IS-Penetrationstests https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Sicherheitsberatung/Pentest_Webcheck/Leitfaden_IS_Penetrationstest.pdf?__blob=publicationFile&v=1

[16] National Vulnerability Database: Alle entdeckten Sicherheitslücken mit Abfragen nach Type und Zeitraum, mit technischen Hintergrundinformationen mit möglicher Umgehungen oder Behebung https://web.nvd.nist.gov/view/vuln/search?search_type=all&cves=on

Bildnachweis

Winfried Gerhard
(info@ordix.de)

Docker Security Internals

Sind meine Container sicher und wenn ja warum?

In der ORDIX® news Ausgabe 2/2015 wurde Docker mit seinen Möglichkeiten vorgestellt und zum Thema Security von Containern kurz Stellung bezogen. Technisch bedingt erfordern Container vielfältige Maßnahmen, um sie voneinander abzugrenzen. Was das Thema Security im Umfeld von Docker bedeutet und welche Funktionen zur Verfügung stehen, damit die betriebenen Container sicher sind, wird in dem folgenden Artikel erläutert.

Technologische Probleme

Im Gegensatz zur Voll- bzw. Paravirtualisierung (z. B. VMware ESXi oder KVM), bei denen jeder Gast einen eigenen Kernel benötigt, nutzen Container den Kernel des Hostsystems. Dies ermöglicht eine bessere Performance, da der Hypervisor und der Kernel des Gastsystems als Zwischenschicht entfallen (siehe Abbildung 1). In diesem Umstand ist jedoch auch der größte Nachteil der Container-Virtualisierung begründet. Aufgrund der geringen Trennung sind vielfältige Maßnahmen notwendig, um die Container voneinander und vom Host-System abzugrenzen und deren Rechte einzuschränken.

Security auf verschiedenen Ebenen

Die Sicherheitsmaßnahmen, die von Docker genutzt werden, werden auf unterschiedlichen Ebenen realisiert. Auf der untersten Ebene implementiert der Kernel u. a. Namespaces und Control Groups, die die grundlegenden Funktionalitäten für die Container-Virtualisierung unter Linux bereitstellen. Die zweite Ebene bilden z. B. die Mandatory Access Control (MAC) Lösungen AppArmor und SELinux, die den Kernel um zusätzliche Sicherheitsfeatures erweitern. Darüber hinaus kann Docker selbst abgesichert werden. Dies umfasst Einschränkungen beim Zugriff auf den Daemon, Tools für die Analyse der Images auf Sicherheitsprobleme und den Betrieb einer eigenen Docker Registry. Die verschiedenen Möglichkeiten auf den einzelnen Ebenen werden im Folgenden erläutert.

Namespaces

Docker nutzt die Kernel-Namespaces, um die Container voneinander abzugrenzen. Sie ermöglichen es, Container isoliert voneinander und vom Host-System zu betreiben, indem sie eine Abstraktionsschicht um die globalen Ressourcen bilden. Für die Prozesse in einem Namespace scheint es, als hätten sie alleinigen Zugriff auf die jeweiligen Ressourcen. Die Namespaces bilden somit die grundlegendste und wichtigste Security-Ebene. Kurz gesagt: Ohne Namespaces keine Docker-Container. Der Kernel implementiert die Namespaces IPC, Mount, Network, PID, User und UTS, die in den folgenden Abschnitten erläutert werden. Zudem wird aufgezeigt, wie diese Namespaces von Docker genutzt werden können.

Der IPC-Namespace stellt den Containern einen isolierten Zugriff auf System V IPC Objects (Message Queues, Semaphoren und Shared Memory Segmente), sowie POSIX Message Queues bereit. Jeder IPC-Namespace hat eigene System V IPC Identifier und ein eigenes POSIX Message Queue Filesystem. Damit sind die Objekte nur für die Prozesse in dem Namespace sichtbar. Wenn ein IPC-Namespace zerstört wird, werden automatisch alle IPC-Objekte, die zu diesem Namespace gehören, gelöscht. Bei Docker haben alle Container standardmäßig einen eigenen IPC-Namensraum. Beispielsweise ist es jedoch mit der Option `--ipc="container:test"` möglich, einen Container zu starten, der den IPC-Namespace mit dem Container „test“ teilt.

Mithilfe des Mount-Namespace kann Prozessen eine separate Sicht auf die Dateisystem-Hierarchie ermöglicht werden. Er spielt somit für Docker eine zentrale Rolle. Jedem Container wird ein eigenes `root`-Filesystem auf Basis eines konfigurierbaren Image zur Verfügung gestellt. Darüber hinaus bietet Docker vielfältige Möglichkeiten, Daten über Volumes innerhalb von Containern bereitzustellen. Dies liegt nicht im Fokus dieses Artikels, kann aber in dem Docker-Artikel in den ORDIX® news 2/2015 nachvollzogen werden.

Die Isolation von Netzwerk-Ressourcen (u. a. Netzwerk-Devices, Protokoll-Stacks und Firewalls) erfolgt über den Network-Namespace. Docker-Container verfügen ohne

spezielle Konfiguration über ein eigenes Loopback und ein eigenes virtuelles Interface, welches mit einer Bridge verbunden ist und die Kommunikation nach außen ermöglicht. Dies entspricht der Option `--net=bridge`. Um einen Container vollständig von der Außenwelt abzuschotten, kann er mit der Option `--net=none` gestartet werden. Ähnlich wie beim Mount-Namespace bietet Docker auch im Bereich Networking viele weitere Optionen. So kann ein Container den Network-Namespace etwa mit dem Host-System oder einem anderen Container teilen.

Für den Betrieb von Containern ist es wichtig, dass diese einen eigenen PID-Namespace besitzen, damit sie keine Kenntnis von den Prozessen außerhalb ihres Namensraums haben. Durch die hierarchische Anordnung der PID-Namespaces ist es dem Parent Namespace möglich, die Prozesse seiner Children zu sehen und zu beeinflussen. Dies führt dazu, dass ein Prozess in einem Child Namespace mehrere PIDs besitzt – eine in seinem Namensraum und eine in dem Parent Namespace (Host-System). Für Docker Container wird automatisch ein eigener PID-Namespace generiert. Bei besonderen Anforderungen kann der Container mittels `--pid=host` auch den Namensraum mit dem Host-System teilen.

Ein wichtiger Schritt für die Sicherheit von Containern ist die Unterstützung des User-Namespace durch die Docker Engine ab Version 1.10. Seitdem kann ein Container eine eigenständige UID- und GID-Range besitzen, die außerhalb des Containers auf einen unprivilegierten Benutzer gemappt wird. Demnach können Prozesse innerhalb eines Containers als Benutzer `root` laufen und damit in dem Namespace über entsprechende Rechte verfügen, auf dem Host-System agieren sie aber „nur“ als normaler Benutzer. Im Gegensatz zu den anderen Namespaces wird der User-Namespace von Docker nicht per Default verwendet, sondern muss erst aktiviert werden. Dies lässt sich damit begründen, dass bei der Verwendung des Namespace eine Koordination mit anderen Komponenten erforderlich ist. Zum Beispiel beim Mapping von Volumes vom Host in den Container, müssen im Vorfeld die Dateirechte angepasst werden, damit der Container auf diese zugreifen kann.

Über die Option `--userns-remap=default` erfolgt die Aktivierung global, die dem Docker Daemon beim Start mitgegeben werden kann. In Folge dessen wird für alle Container der User-Namespace automatisch aktiviert, der jedoch über die Option `--userns=host` beim Erstellen eines Containers explizit wieder deaktiviert werden kann. Der Parameter default für die Option `userns-remap` bewirkt, dass Docker einen Benutzer und eine Gruppe mit dem Namen dockremap erstellt, falls diese noch nicht existieren. Darüber hinaus wird eine 65536 lange, zusammenhängende Range in der `/etc/subuid` bzw. `/etc/subgid` reserviert, die auf den Benutzer bzw. die Gruppe dockremap gemapped wird.

Docker Container werden standardmäßig mit einem eigenen UTS-Namespace gestartet. Sie verfügen somit über einen eigenen Host- und Domänenamen. Um den UTS-Namespaces des Hosts auch im Container zu nutzen, kann dieser mit der Option `--uts=host` erstellt bzw. gestartet werden. Aus Security-Sicht ist von einer solchen Konfiguration jedoch abzuraten, da aus dem Container der Hostname des Host-Systems geändert werden könnte.

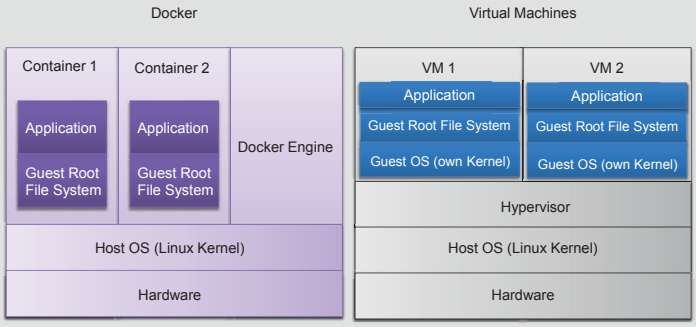


Abb. 1: Docker im Vergleich zu VMs

Eigenschaft	Option
Relative Gewichtung der CPU-Zeit zwischen den Containern	`--cpu-shares`
Beschränkungen eines Containers auf spezielle CPUs	`--cpuset-cpus`
Limitierung der CPU-Ressourcen	`--cpu-period` `--cpu-quota`
Limitierung der Memory-Nutzung	`-m / --memory`
Relative Block-I/O-Gewichtung zwischen den Container	`--blkio-weight`
Limitierung des I/Os pro Device	`--device-readwrite-bps` `--device-write-bps`

Abb. 2: Docker-Optionen zur Verwendung von Cgroups

Capability	Capability
CAP_CHOWN	CAP_SETUID
CAP_DAC_OVERRIDE	CAP_SETFCAP
CAP_FSETID	CAP_SETPCA
CAP_FOWNER	CAP_NET_BIND_SERVICE
CAP_MKNOD	CAP_SYS_CHROOT
CAP_NET_RAW	CAP_KILL
CAP_SETGID	CAP_AUDIT_WRITE

Abb. 3: Default-Capabilities für Container

Control Groups

Control Groups oder kurz Cgroups sind die zweite Kernkomponente der Container-Virtualisierung unter Linux. Sie wurden in der ORDIX® news 3/2010 bereits ausführlich besprochen und dort können die im Folgenden erwähnten Eigenschaften im Detail nachvollzogen werden. Docker nutzt die Cgroups mit Blick auf das Thema Security, um die Ressourcennutzung von Containern zu beschränken, so dass sie sich untereinander und das Host-System nicht beeinflussen können. Im Wesentlichen können die in Abbildung 2 aufgeführten Eigenschaften über die angegebene Option beim Erstellen bzw. Starten des Containers gesetzt werden. Seit der im April 2016 veröffentlichten Docker Version 1.11 kann zusätzlich der PIDs Resource Controller verwendet werden, der mit dem Kernel 4.3 eingeführt wurde. Dieser ermöglicht die Limitierung der Anzahl der Prozesse in einer Control Group und kann damit z. B. verhindern, dass Fork-Bomben in einem Container das ganze System beeinträchtigen.

Capabilities

Die Capabilities sind ein Standard-Feature des Linux-Kernels und stehen schon mehrere Jahre zur Verfügung. Sie werden allerdings verhältnismäßig selten eingesetzt, da ihre Konfiguration tiefe Kenntnisse über die verwendeten Applikationen erfordern. Von Docker werden sie jedoch standardmäßig eingesetzt, da sie die Sicherheit von Containern erhöhen. Im Allgemeinen wird mithilfe von Capabilities das starre **`root/non-root`** Rechtesystem von Linux durch eine granulare Rechteverwaltung abgelöst. Ein **`root`**-User in einem Container benötigt nicht die vollständigen **`root`**-Rechte, da ein Großteil der Verwaltung auf dem Host-System erfolgt und im Container nur die eigentliche Anwendung betrieben wird.

Docker entzieht den Containern viele Rechte (Capabilities), so dass auch der **`root`**-User in einem Container bestimmte Tasks nicht ausführen kann. Zu diesen gehören z. B. alle Mount-Operationen und das Laden von Kernel-Modulen. Dabei wurde ein Whitelist-Ansatz gewählt und dem Container werden nur die explizit definierten Rechte gewährt. Eine vollständige Liste ist in Abbildung 3 aufgeführt. Dies erschwert Angreifern, die Zugriff zu einem Container erlangt haben, den Ausbruch erheblich. Der Administrator hat die Möglichkeit, Capabilities über den Standard hinaus anzupassen, indem er über die Optionen **`--cap-add`** bzw. **`--cap-drop`** weitere Capabilities hinzufügt oder entfernt.

Seccomp-Profile

Ein weiteres Feature des Linux-Kernels ist der Secure Computing Mode (Seccomp). In Verbindung mit Docker wird verhindert, dass Applikationen in einem Container definierte Systemcalls ausführen können. Dies überschneidet sich zum Teil mit den beschriebenen Capabilities, bietet aber einen detaillierteren Ansatz. Docker unterstützt Seccomp seit der Version 1.10 und ermöglicht die Konfiguration der erlaubten bzw. verbotenen Systemcalls über JSON-Profile. Wenn der Kernel Seccomp unterstützt, werden alle Container mit einem Default-Profil gestartet, das aktuell den Aufruf von 44 Systemcalls verbietet. Das Default-Profil kann entweder mit der Option **`--security-opt seccomp=unconfined`** deaktiviert oder durch die Angabe eines selbst erstellen Profils überschrieben werden.

AppArmor, SELinux & Co

Mandatory Access Control (MAC) Systeme ermöglichen es über die üblichen Dateisystemrechte hinaus, den Zugriff auf Objekte für einzelne Prozesse zu regulieren. Dies kann genutzt werden, um die Rechte von Containern einzuschränken und damit zur Sicherheit beizutragen. Für Linux stehen mehrere Lösungen zur Verfügung, u. a. AppArmor, Grsecurity, SELinux, Smack und TOMOYO. Da die wichtigsten Distributionen entweder AppArmor oder SELinux einsetzen, werden nur diese beiden Lösungen im Folgenden vorgestellt. Beide nutzen das Linux Security Modules (LSM) Framework des Kernels und bieten somit einen ähnlichen Funktionsumfang.

AppArmor ist seit der Version 2.6.36 im Kernel integriert und wird von den Suse-Distributionen und Ubuntu eingesetzt. Die Konfiguration von AppArmor für einzelne Anwendungen erfolgt über sogenannte Profile, die entweder von dem Distributor mitgeliefert oder händisch erstellt werden müssen. Der Vorteil von AppArmor liegt in der guten Lesbarkeit der Profile. Dies erleichtert den Einstieg und macht die Nutzung auch für weniger versierte Administratoren möglich. Für Docker werden bereits Profile von Suse (**`/etc/apparmor.d/docker`**) bzw. Ubuntu (**`/etc/apparmor.d/docker-default`**) mitgeliefert. Sie regulieren u. a. den Zugriff auf **`/proc`** und **`/sys`** und werden, wenn AppArmor aktiviert ist, automatisch genutzt.

SELinux wurde mit dem 2.6er-Kernel eingeführt und ursprünglich von der NSA entwickelt. Es ist die Default-MAC-Lösung der Red Hat Distributionen, die aktuell einen Großteil der Entwicklung tragen. Im Vergleich zu AppArmor bietet SELinux eine detailliertere Konfiguration, die jedoch deutlich komplexer ist. Alle Dateien, Verzeichnisse, Devices usw. werden mit einem Label (Security-Context) versehen. Dieses besteht aus den Informationen User, Role, Type und Level, die eine Kombination aus Role-Based Access Control (RBAC), Type Enforcement (TE) und optional Multi-Category Security (MCS) und Multi-Level Security (MLS) ermöglichen. Für Docker unter Red Hat wird eine Policy, in der verschiedene Rules zusammengefasst sind, mitgeliefert. Sie sichert mittels Type Enforcement das Host-System ab. Zusätzlich werden die einzelnen Container voneinander abgegrenzt, indem jedem Container ein eindeutiges Level zugewiesen wird (MCS).

Absicherung des Daemons

Neben dem Einsatz der verschiedenen Kernelfeatures und einer der MAC-Lösungen, sollte der Docker Daemon

(Engine) abgesichert werden. Da der Daemon als **root** läuft, hat jeder Benutzer, der Container starten kann, im Prinzip **root**-Rechte auf dem System. Er könnte z. B. **/etc** als Volume in einem Container zur Verfügung stellen und damit die Passwort-Hashes in der **/etc/shadow** lesen und diese evtl. entschlüsseln. Dies wird zwar in der Standardkonfiguration von AppArmor und SELinux unterbunden, häufig wird Docker aber auch ohne eine MAC-Lösung eingesetzt bzw. deren Konfiguration könnte fehlerhaft sein.

Dementsprechend sollte es nur definierten Benutzern möglich sein, Container zu starten. Die Engine kann Anfragen über drei verschiedene Sockets (Unix, TCP und FD) empfangen. Es ist notwendig, die Möglichkeiten des Zugriffs für die verschiedenen Arten zu kennen, um diese ggf. einzuschränken. Der Socket (**/var/run/docker.sock**), der der Default ist, sollte mit möglichst geringen Rechten versehen werden. Abhängig von der genutzten Distribution unterscheiden sich die Standardrechte und Docker ist entweder nur als **root** bzw. auch für Mitglieder der Gruppe **docker** verwendbar.

Die Option **-H fd://** für den Docker-Daemon führt dazu, dass der beschriebene Socket nicht vom Docker-Daemon selbst, sondern über Socket-Aktivierung von **systemd** angelegt wird und dieser die Anfragen an Docker weiterleitet. In diesem Fall müssen die Rechte - falls erforderlich - in der **docker.socket Unit** angepasst werden. In einem professionellen Umfeld wird Docker häufig aus der Ferne administriert. Um den Zugriff zu ermöglichen, muss Docker mit der Option **-H <IP-Adresse>:<Port>** gestartet werden. Diese Verbindung ist jedoch ohne weitere Konfiguration unverschlüsselt und unautorisiert. Über die Option **--tlsverify** und mit der Angabe eines Zertifikats, dessen Keys und dem Zertifikat der CA kann die Verbindung zum Client verschlüsselt werden. Zusätzlich akzeptiert der Daemon nur noch Verbindungen von Clients, die über ein entsprechendes Client-Zertifikat verfügen.

Authorization Plugins

Die beschriebenen Möglichkeiten zur Absicherung des Zugriffs auf den Docker-Daemon sind eine „Alles oder Nichts"-Entscheidung. Ein User, der sich mit der Engine verbinden kann, hat immer vollen Zugriff. Seit der Version 1.10 können Authorization Plugins genutzt werden, um eine verbesserte Zugriffskontrolle zu ermöglichen. Von Docker selbst wird noch kein Plugin mitgeliefert, mit dessen Hilfe eine dedizierte Autorisierung ermöglicht werden kann. Am vielversprechendsten ist im Moment der Twistlock AuthZ Broker, der als eigener Container betrieben wird und von der Engine mit der Option **--authorization-plugin=authz-broker** genutzt werden kann.

Beim Twistlock AuthZ Broker erfolgt die Authentifizierung über Docker, indem der Benutzername aus dem Client-Zertifikat genutzt wird. Auf Basis dieses Namens kann konfiguriert werden, welche Operationen der Nutzer durchführen darf. Die Möglichkeiten reichen von der Freigabe aller Befehle bis hin zu einem **readonly**-Zugriff auf die Logfiles eines bestimmten Containers. Somit kann Docker auch in Umgebungen eingesetzt werden, in denen unterschiedliche Parteien Zugriff auf eine Engine haben.

```
# analyze-local-images 2fa927b5cdd3
Saving 2fa927b5cdd3 to local disk (this may take some time)
Retrieving image history
Analyzing 5 layers...
Analyzing b0a8...db81
...
Analyzing 45ef...7aee
Retrieving image's vulnerabilities
Clair report for image 2fa927b5cdd3 (2016-06-25
07:02:16.201427678 +0000 UTC)
Success! No vulnerabilities were detected in your image
```

Abb. 4: Analyse von lokalen Images mit Clair

Quellen

[Q1] Schürmann, Tim, „Systemsicherheit erhöhen mit Grsecurity".
In: ADMIN, IT Praxis & Strategie,
http://www.admin-magazin.de/Das-Heft/2012/04/Systemsicherheit-erhoehen-mit-Grsecurity

[Q2] Docker Security:
https://sreeninet.wordpress.com/2016/03/06/docker-security-part-1overview/

[Q3] Frazelle, Jessi, „Docker Engine 1.10 Security Improvements",
https://blog.docker.com/2016/02/docker-engine-1-10-security/

Images im Fokus

Die Absicherung des Hosts und die Trennung der Container untereinander wurden in den bisherigen Punkten erläutert. Letztendlich werden jedoch Anwendungen in Containern betrieben, die ebenfalls möglichst sicher sein sollten. Docker hat Ende 2015 mit Nautilus ein Projekt vorgestellt, mit dem die Images im offiziellen Repository auf Schwachstellen überprüft werden. Nautilus ist jedoch, trotz anderer Versprechen, noch nicht für die Öffentlichkeit freigegeben.

Mit Clair steht jedoch eine Alternative bereit, die von CoreOS entwickelt wird. Clair führt statische Analysen durch und scannt damit Images und nicht laufende Container. Dabei wird jede Schicht des Images separat analysiert. Für die Erkennung von Schwachstellen greift Clair auf die Common Vulnerabilities and Exposures (CVE) Datenbanken von Debian, Red Hat und Ubuntu zurück. Der einfachste Weg, Clair einzusetzen, ist das analyze-local-images Tool. Mit ihm können eigene oder aus dem Docker Hub bezogene lokale Images auf Sicherheitsprobleme untersucht werden (siehe Abbildung 4). Wird eine eigene Docker Registry betrieben, kann Clair in die Registry integriert werden, um die Analyse zu automatisieren.

Kein root im Container

Eine weitere Maßnahme zur Absicherung der Container ist, Applikationen innerhalb des Containers, wie auch auf einem normalen System, möglichst ohne `root`-Rechte zu betreiben. Je weniger Rechte eine Applikation hat, umso schwieriger ist es für einen Angreifer, über die Applikation hinaus Rechte zu erlangen. Dieser Grundsatz gilt auch für Docker-Umgebungen und ebenfalls wenn der User-Namespace eingesetzt wird. Es sollte somit schon im Dockerfile über die Option `USER <non-root User>` sichergestellt werden, dass die Applikation als unprivilegierter Benutzer läuft.

Private Registry

Das öffentliche Verzeichnis für Docker Images ist das Docker Hub. Es bietet Zugriff auf tausende Images der Community und aus offiziellen Quellen. Dies erlaubt einen schnellen und unkomplizierten Betrieb von Docker, aber es gibt trotzdem mehrere Gründe, eine eigene Registry zu betreiben. Dafür sprechen z. B. die volle Kontrolle über den Speicherort der Images, der fehlende Zwang für eine Verbindung ins Internet und aus Security-Sicht die Möglichkeit, die verfügbaren Images zu kontrollieren. Für den Betrieb einer privaten Registry kann entweder die unter der Apache-Lizenz veröffentlichte und frei verfügbare Docker Registry oder die Trusted Registry, für die die Docker Inc. Support anbietet, verwendet werden. Im einfachsten Fall wird die freie Registry in einem Container betrieben. Dafür steht das offizielle Image registry bereit.

Fazit

Dieser Artikel hat gezeigt, dass vielfältige Maßnahmen auf den unterschiedlichen Ebenen zur Verfügung stehen, um Docker Container sicher betreiben zu können. Insbesondere in den letzten Releases (seit Anfang 2016) sind viele Features hinzugekommen, die zur Sicherheit von Containern beitragen. Die Betrachtung der einzelnen Features auf unterschiedlichen Ebenen vereinfacht deren Einordnung und bringt Ordnung in das Chaos. Dies ist wichtig, da nur eine Kombination aus den Kernel-Features, den MAC-Lösungen (AppArmor und SELinux), der Absicherung der Engine und die sichere Gestaltung der Images einen sicheren Betrieb von Docker ermöglichen.

Christopher Herclik
(info@ordix.de)

Marius Dorlöchter
(info@ordix.de)

Newton: OpenStack endeckt die Gravitation

Spätestens mit dem aktuellen 14. Release, das unter dem Namen „Newton“ veröffentlicht wurde, sollte OpenStack auf dem Plan jedes IT-Entscheiders stehen. Um die Möglichkeiten von OpenStack Newton einschätzen zu können, bietet dieser Artikel einen Überblick über die Architektur von OpenStack und erläutert das Aufgabengebiet der einzelnen Komponenten.

Was ist OpenStack?

OpenStack ist die derzeit innovativste Cloud-Infrastrukurlösung auf dem Markt. Dies ist zu einem großen Teil auf die OpenStack Foundation zurückzuführen, die die Steuerung des Projekts übernimmt und in der sich mehr als 37.000 Mitglieder zusammengeschlossen haben. Außergewöhnlich ist dabei die Zusammensetzung aus Technologieanbietern, großen Unternehmen, innovativen Cloud-Startups und individuellen Entwicklern.

Bei OpenStack handelt es sich um ein Open-Source-Projekt, das unter der Apache-2.0-Lizenz veröffentlicht wird. Mithilfe von OpenStack lassen sich komplexe Cloud Computing Infrastrukturen aufbauen. Zudem unterstützt es IT-Architekten bei der Orchestrierung und dem Management der Cloud-Umgebung. Entstanden ist das Projekt aus einer Kooperation zwischen der US-amerikanischen Weltraumbehörde NASA und der Firma Rackspace, die jeweils eine Komponente (Computing und Object Storage) entwickelten und diese zu einem Projekt zusammenschlossen.

Architektur

Die OpenStack-Architektur kann in die drei Bereiche Komponenten (Services), Treiber und APIs unterteilt werden. Die wichtigste Aufgabe übernehmen die einzelnen Komponenten, die nur in Kombination eine vollständige Cloud-Infrastruktur-Lösung ergeben und in separaten Teilprojekten entwickelt werden. Die Verwaltung der einzelnen Komponenten und deren Kommunikation untereinander, erfolgt über RESTful APIs. Die wichtigsten Komponenten und deren Zusammenhänge (siehe Abbildung 1) werden im folgenden Abschnitt erläutert.

Einige der OpenStack-Komponenten implementieren eine Treiberschicht. Diese hat die Aufgabe, die Integration und Kommunikation mit externen hardware- und softwarebasierten Infrastrukturkomponenten sicherzustellen. Die Treiber werden entweder direkt von den Herstellern passend zu ihren Lösungen oder von der OpenStack Community entwickelt und bereitgestellt. Die meisten Treiber betreffen die Bereiche Compute, Networking und Storage. Die

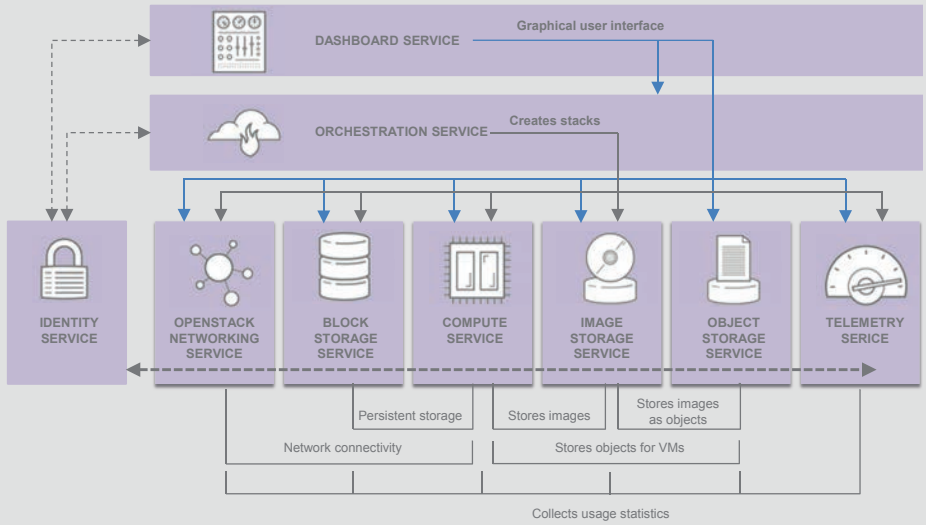


Abb. 1: Die wichtigsten Komponenten und ihre Beziehungen (Quelle: https://access.redhat.com/documentation/en/red-hat-openstack-platform/9/single/architecture-guide/)

Codename	Beschreibung
Horizon	Dashboard – Webbasiertes User-Interface für viele der anderen OpenStack-Komponenten
Ceilometer	Telemetry - Sammlung von Daten der Kernkomponenten, die für die Abrechnung weiterverwendet oder auf Basis von denen Alarme definiert werden können
Heat	Orchestration - Orchestrierung von zusammengesetzten Cloud-Applikationen
Trove	Database as a Service - Automatisierung der Administration von relationalen Datenbanken
Sahara	Elastic Map Reduce - Möglichkeiten zur Bereitstellung von Data-Processing-Frameworks (z. B. Hadoop oder Spark)
Ironic	Bare-Metal Provisioning - Vergleichbar mit Nova, nur für die Bereitstellung von physikalischen Servern
Zaqar	Messaging Service - Cloud Messaging und Notification Service, der von Entwicklern von Web oder mobilen Applika- tionen genutzt werden kann
Manila	Shared Filesystems - Stellt Dateisysteme über NFS und SMB für Instanzen oder extern bereit
Designate	DNS Service - Einfache Verwaltung für DNS, unterstützt Power-DNS und Bind9
Barbican	Key Management - Sichere Speicherung, Verteilung und Manage-ment von Secrets (Passwörter, Keys und Zertifikate)
Magnum	Containers - Bietet einen OpenStack-kompatiblen Zugriff auf die Container-Engines Docker und Kubernetes
Murano	Application Catalog - Zugriff auf und Veröffentlichung von vorge-fertigte Cloud-Applikationen
Congress	Governance - Ermöglicht die Definition von Policies für verschie-dene Cloud-Services
Aodh	Alarmierung – Basierend auf den Daten von Ceilometer und defi-nierten Schwellwerten können Aktionen getriggert werden
CloudKitty	Rating - Stellt eine Möglichkeit bereit, die von Ceilometer gesammelten Daten zu bewerten, um sie im Anschluss für die Abrechnung nutzen zu können
Freezer	Backup - Backup- und Desaster-Recovery-Plattform
Mistral	Workflow - Definition von einzelnen Tasks in einem Wokflow, die zusammenhängend ausgeführt werden
Monasca	Monitoring - Überwachung der OpenStack Services und Bereit-stellung von Metriken
Searchlight	Index and Search - Indexiert die OpenStack Ressourcen und ermöglicht eine schnelle und einfache Suche
Senlin	Cluster - Eine generische Cluster-Lösung für OpenStack Services
Solum	Platform as a Service - Automatisierte Bereitstellung von Applika-tionen in Docker-Containern
Tacker	Virtual Network Function - Management und Orchestrierung von virtuellen Netzwerken
Watcher	Resource Optimization - Optimierung der Ressourcennutzung z. B. durch das Verschieben von VMs
Panko	Event Storage - Erweiterung für Ceilometer um die Indexierung von Metadaten und die Speicherung von Events
Vitrage	Root Cause Analysis - Erweiterung, Organisation und Analyse von Alarmen und Events

Abb. 2: Optionale Komponenten im Newton-Release

Treiber für den Bereich Compute dienen in der Regel der Unterstützung von Virtualisierungstechnologien, wie z. B. Microsofts Hyper-V, KVM oder VMware vSphere. Der Bereich Networking deckt, wie der Name schon andeutet, den Support für Netzwerk-Hardware und -Software (u. a. Cisco, Open vSwitch oder VMware NSX) ab. Ebenfalls werden die wichtigsten Speicherlösungen von EMC, Hitachi, NetApp und vielen anderen unterstützt.

Die meisten Services können über drei Wege angesprochen werden. Die benutzerfreundlichste Variante ist das Graphical-User-Interface (GUI), das ein Teil des Dashboards ist. Zudem existiert ein Command-Line-Interface, welches von Administratoren direkt oder in Skripten verwendet werden kann. Im Hintergrund verwenden beide Varianten die API der entsprechenden Komponente. Diese kann auch direkt z. B. von 3rd-party-Applikationen aufgerufen werden und bildet das primäre Interface für die Kommunikation der Komponenten untereinander. Die API ist RESTful und dementsprechend HTTP-basiert und unterstützt JSON- sowie XML-Daten.

Komponenten

Die Aufteilung in einzelne Komponenten, auf Basis der zu erfüllenden Aufgaben, ermöglicht eine flexible Gestaltung der OpenStack-Infrastruktur. Nicht jeder Service wird in jeder OpenStack-Installation benötigt. Die Core-Komponenten, die in den meisten Installationen vorhanden sind, werden in den nächsten Punkten vorgestellt. Intern werden die Services mit einem Codenamen versehen, der sich z. B. in den Kommandozeilen-Tools und den Konfigurationsdateien wiederfindet. Die Komponenten können alle auf einem System installiert sein. Dies macht Testinstallationen einfach. In Produktionsumgebungen werden die einzelnen Komponenten in der Regel auf mehreren physikalischen Systemen verteilt.

Compute - „Nova“

Nova bildet den Kern einer OpenStack-Infrastruktur, wenn diese auf den Betrieb von virtuellen Maschinen ausgelegt ist. Sie ist die komplexeste Komponente und beinhaltet alle Dienste, die für die Verwaltung von Cloud-Instanzen zuständig sind. Die eigentliche Virtualisierung der Instanzen erfolgt durch die Hypervisoren auf physikalischen Systemen, den sogenannten Compute Nodes. Nova unterstützt mithilfe von Treibern unterschiedliche Virtualisierungstechnologien, die aufgrund der Abstraktionsschicht gleichzeitig genutzt werden können.

Zusätzlich bietet Nova noch weitere instanzspezifische Funktionen. Es arbeitet eng mit anderen Komponenten zusammen und übernimmt deren Koordination. Die einzelnen Instanzen generieren sich aus statischen Images. Sie enthalten Betriebssysteme sowie alle weiteren Programme und werden vom Image Service (Glance) verwaltet. Für die Bereitstellung der Netzwerkressourcen, für eine Instanz greift Nova auf Neutron zu. Volumes werden der Instanz über dem Block Storage Service Cinder bereitgestellt.

Object Storage – „Swift“

Swift ist mit Nova eines der beiden initialen Projekte von OpenStack und ein verteiltes, skalierbares und objektbasiertes Speichersystem. In einem Object Storage werden die Daten nicht wie in einem File Storage hierarchisch, sondern innerhalb eines Containers auf nur einer Ebene abgelegt. Jedes Objekt (BLOB) wird mit einem eindeutigen Identifikator versehen, sodass es beim Zugriff nicht notwendig ist, den physikalischen Standort des Objektes zu kennen. Die Daten in Swift können nach dem Erstellen nur gelesen und gelöscht, jedoch nicht verändert werden.

Für die Nutzung in einer Cloud-Architektur ist ein Object Storage ideal, da der Unique Identifier für jedes Objekt dafür sorgt, dass der automatisierte Zugriff erleichtert wird. Im Falle von OpenStack kann Swift für die Speicherung von Images, Snapshots und Cloud-Instanzen genutzt werden. Darüber hinaus bietet die Architektur von Swift viele Eigenschaften, die insbesondere für die Anwendung als Cloud-Speicher wichtig sind. Swift ist jederzeit durch weitere Speichereinheiten erweiterbar und somit horizontal hoch skalierbar. Durch das Fehlen einer zentralen Organisationseinheit und der Möglichkeit, alle Komponenten häufiger redundant auszulegen, kann ein Single-Point-of-Failure vermieden werden.

Block Storage - „Cinder“

Cinder ist der Block Storage Service von OpenStack und im Vergleich zu Swift weniger komplex. Es bietet keine automatische Verteilung und Replikation der Daten. Cinder stellt den Instanzen blockbasierten Speicher als Volumes bereit. Volumes können den virtuellen Maschinen vor dem Starten und im laufenden Betrieb zugewiesen werden. Sie eignen sich vor allem für performancesensitive Anwendungen und solche, die einen direkten Zugriff auf Block-Devices benötigen. Als Storage-Backend können diverse Storage-Lösungen (z. B. LVM, NetApp, EMC und Hitachi) dienen.

Image - „Glance“

Im zeitlichen Ablauf ist Glance der Beginn des Compute-Workflows. Der Image Service ist für die Registrierung, Auflistung und das Abrufen von Images für die Instanzen zuständig. Es übernimmt die Verwaltung der Images, aber nicht deren Speicherung. Glance stellt lediglich eine Abstraktionsschicht für die verschiedenen Storage-Technologien dar. Unterstützt werden die Storage-Backends Cinder, Swift, Amazon S3 und lokale Filesysteme.

Das zentrale OpenStack Repository ist der Image Service und beinhaltet die Images: Metadaten und Statusinformationen. Dabei werden die wichtigsten Image-Formate (z. B. AMI, ISO, QCOW2, VDI und VMDK) unterstützt. User und Services können private oder öffentliche Images speichern, die zum Starten von Instanzen verwendet werden.

Networking - „Neutron“

Für die Kommunikation der Instanzen untereinander sowie mit der Außenwelt ist Neutron verantwortlich. Es ermöglicht das Management von Netzwerken und bietet dabei Funktionen wie z. B. VLANs, DHCP, Firewall as a Service, Quality of Service und ACLs. Neutron ist modular aufgebaut und über einen Plugin-Mechanismus anpassbar. Dieser erleichtert nicht nur die Integration in bestehende Umgebungen, sondern auch die Erweiterung um Funktionen. Es stehen Plugins für die wichtigsten Netzwerk-Komponenten und -Software zur Verfügung.

Identity -„Keystone“

Eine der Kernkomponenten von OpenStack ist Keystone. Es bildet den zentralen Dienst für die Authentifizierung der Benutzer und das Rechtemanagement für die anderen Komponenten. Dabei implementiert Keystone keine eigenen Funktionen, um Benutzer zu speichern, sondern bildet eine Abstraktionsschicht für verschiedene Backends (z. B. SQL, PAM, LDAP oder AD). Damit kann es die bestehende Benutzerverwaltung für die OpenStack-Infrastruktur verwenden.

Bevor ein Benutzer eine Aktion mit einer der OpenStack-Komponenten ausführen kann, muss er sich gegenüber Keystone authentifizieren. Zusätzlich zu der Überprüfung von Benutzername und Passwort bzw. Zertifikaten können mithilfe von Rollen fein konfigurierbare Berechtigungen realisiert werden. Darüber hinaus kann die OpenStack-Installation auf Basis von Projekten getrennt werden. Dies ist nicht nur auf Ebene der Benutzer, sondern auch auf der Netzwerk- und Speicherebene möglich.

Die Entdeckung der Gravitation

Die beschriebenen Kernkomponenten sind seit Längerem ein Teil der offiziellen OpenStack Releases. Ihnen werden eine Reihe von optionalen Services zur Seite gestellt, deren Umfang von Version zu Version anwächst. Die Entwicklung in unabhängigen Projekten ist ein Grund für die hohe Dynamik von OpenStack insgesamt. Die offiziellen Releases, die in einem halbjährlichen Rhythmus veröffentlicht werden, enthalten einen definierten Stand der Komponenten. Für jedes Release werden im Vorfeld Ziele für die Entwicklung definiert. Der Fokus lag bei Newton auf Verbesserungen der Skalierbarkeit und der Elastizität sowie einer verbesserten Integration von virtuellen und physikalischen Servern und Containern. Eine Liste der im aktuellen Release enthaltenen, optionalen Komponenten, die nach ihrer Relevanz sortiert sind, ist in Abbildung 2 aufgeführt.

Horizon, Ceilometer und Heat

Die Services Horizon, Ceilometer und Heat sind Teil der meisten OpenStack-Distributionen und aus diesem Grund auch in der Abbildung 1 dargestellt. Das Dashboard

Horizon ist in der Regel der erste Berührungspunkt mit OpenStack und bietet zumindest für die detailliert vorgestellten Komponenten eine grafische Möglichkeit zur Administration. Darüber hinaus kann der Endanwender bzw. Kunde über Horizon im Rahmen seiner Berechtigungen seine zugewiesenen Ressourcen verwalten.

Der Telemetry-Service Ceilometer ist für die Ermittlung und Sammlung des gesamten Nutzungs- und Performance-Verhaltens aller OpenStack-Komponenten zuständig. Die Daten werden pro Benutzer bzw. Kunde ermittelt und ermöglichen z. B. eine spätere Bewertung und Abrechnung der genutzten Ressourcen. Das Rating und die Abrechnung sind dabei kein Bestandteil von Ceilometer. Die gesammelten Daten werden lediglich über eine API zur Verfügung gestellt. Diese können von Drittanwendungen angesprochen werden, die die Folgeverarbeitung übernehmen.

Quellen

[1] John Rhoton, Jan De Clercq und Franz Novak:
„OpenStack Cloud Computing“;
1. Auflage; Tunbridge Wells: Recursive Press; 2014

[2] Carlo Velten, Rene Büst und Max Hille:
„OpenStack im Unternehmenseinsatz“;
Crisp Research AG; 2014

[3] Tilman Beitter, Thomas Kärgel, André Nähring,
Andreas Steil und Sebastian Zielenski:
„IaaS mit OpenStack. Cloud Computing in der Praxis“;
1. Auflage; Heidelberg: dpunk.verlag; 2014

[4] Tom Fifield, Diane Fleming, Anne Gentle, Lorin Hochstein,
Jonathan Proulx, Everett Toews und Joe Topjian:
„OpenStack Operations Guide“;
1. Auflage; Sebastopol: O’Reilly; 2014

Heat ist das OpenStack-Projekt, hinter dem sich die Orchestrierungsfunktionalität verbirgt. Heat ermöglicht die Beschreibung von komplexen Cloud-Applikationen in Templates. Die Sammlung der Objekte und ihre Verknüpfung untereinander werden als Stack bezeichnet. Die Templates werden von der Heat-Engine eingelesen und ausgeführt. Die Engine triggert, wie in Abbildung 1 zu sehen ist, die weiteren Komponenten wie Nova und Cinder, die die eigentlichen Ressourcen bereitstellen.

Zusammenfassung

Die Kernkomponenten von OpenStack stehen seit mehreren Jahren zur Verfügung. Diese bilden die Grundlagen für den Aufbau und die Verwaltung einer IT-Infrastruktur mit OpenStack. Deren Funktionsumfang kann durch eine wachsende Anzahl von optionalen Komponenten erweitert werden. Durch die unabhängige Entwicklung in eigenständigen Projekten ist eine hohe Dynamik gewährleistet, die wesentlich zum Erfolg von OpenStack beiträgt. Dazu kommt die offene Architektur, die auf standardisierten Schnittstellen zwischen den Komponenten beruht und eine Erweiterung vereinfacht. Mit OpenStack steht dementsprechend eine Lösung bereit, die einen großen Schritt in Richtung Standardisierung von IT-Infrastrukturen ermöglicht und bei der Automatisierung helfen kann.

Marius Dorlöchter
(info@ordix.de)

Neuheiten WebLogic Server 12.2.1

Liebling Kreuzberg: die Mandanten warten

Manfred Krug († Oktober 2016) als Anwalt Liebling wusste es schon: Mandanten müssen unabhängig voneinander betreut werden und man sollte sie nicht zu lange warten lassen. Multitenant (Mandantenfähigkeit) ist der entsprechende einschlägige Begriff in der IT und dieser soll im vorliegenden Beitrag zur neuen WebLogic Server Version 12.2.1 die Hauptrolle spielen. Gerade aus der Cloud-Thematik bekannte Dienste wie PaaS (Platform as a Service) oder SaaS (Software as a Service) benötigen auf Anbieterseite die Kernanforderung der Abgrenzung. Soll heißen, Services einer Cloud müssen einen strikt abgeschirmten Bereich für die User zur Verfügung stellen. Es darf zu keinerlei Störungen oder Seiteneffekten kommen, die etwa durch gemeinsam genutzte Speicherbereiche oder durch schlecht abgegrenzte Sicherheitszonen entstehen könnten. Mit der Partitionierung steht ab sofort ein neues, mächtiges Werkzeug im WebLogic Server zur Verfügung, um hier Unterstützung anbieten zu können. Und das wollen wir uns nun genauer anschauen.

Multitenant – ein passendes Szenario

Wir wollen uns ein Szenario vorstellen mit einer brandneuen, fiktiven Webanwendung, die äußerst innovativ und ansprechend daherkommt und somit auf zahlreiche pozentielle User im Lande attraktiv wirkt. Viele könnten sich gut vorstellen, diese Anwendung zunächst einmal ausgiebig zu testen und sie dann auch intensiv zu nutzen. Allerdings würden bei einem Kauf hohe Kosten anfallen. Des Weiteren könnte dieses Programm einem hochfrequenten Update-Zyklus unterliegen, sodass es sinnvoll erscheint, die Anwendung vom Hersteller zu mieten.

Der Anbieter könnte sie auf seiner Infrastruktur zur Verfügung stellen, für regelmäßige Wartung und Updates sorgen und gewährleisten, dass unterschiedliche User sich bei gleichzeitiger Nutzung nicht ins Gehege kommen. Dieses Szenario wird Ihnen vielleicht bekannt vorkommen, denn es firmiert unter dem Stichwort PaaS (Program as a Service) im Cloud-Kontext und war eines der Hype-Themen der vergangenen Jahre. Ob Hype oder nicht, es ist und bleibt eine plausible und gerechtfertigte Anforderung an moderne Unternehmens-IT-Systeme, solche Anwendungen mit den beschriebenen Anforderungen zur Verfügung zu stellen.

Was hat Oracle im Middelware-Segment anzubieten?

Oracle, als Big Player im IT-Sektor, setzt als Middleware-Komponente bekanntermaßen sehr stark auf WebLogic Server (WLS), welcher Ende 2015 einen eher klein wirkenden Versionssprung von Version 12.1.3 auf 12.2.1 vollzogen hat. Dahinter verbirgt sich jedoch neben zahlreichen Technologieanpassungen auf neue Standards und der Eliminierung von alten Schnittstellen ein gänzlich neues Konzept im WLS Kontext, nämlich die Thematik „Partitionierung von Domains“.

Bevor wir die Technik der Partitionierung besprechen, soll zunächst der Bogen zu PaaS gespannt werden. Hilfreich bei dieser Betrachtung ist die Frage: „Wie war PaaS mit den ‚alten‘ Mitteln des WLS zu bewerkstelligen?“ Nun, es kommt darauf an, eine Anwendung als vielfache Kopien betreiben zu können, die sich untereinander nicht beeinflussen können. Bislang konnte man mehrere Domains definieren, in jeder Domain war die Anwendung zu deployen und so konnten mehrere Instanzen derselben Anwendung laufen. Als Beispiel für ein PaaS-Szenario können unterschiedliche Unternehmensbereiche wie Entwicklung, Test, Integrationstest, Performanztest, Produktion angesehen werden.

Alles hatte die Domain zu leisten!

Eine Domain stellte bislang die umfassende Organisationseinheit in Bezug auf Server, Applikationen, Sicherheitsbereiche etc. dar. Alles, was in WebLogic-Umgebungen eingerichtet, deployt, administriert, gemanagt wird, befindet sich also in solch einer Domain. Es ist damit der allumfassende Container, welcher zentral administrierbar

ist und dem dadurch eine starke Bedeutung zukommt, insbesondere in großen, kritischen Systemlandschaften.

Partitionen ebnen einen eleganten Weg zu Multitenant

Ab der neuen WLS Version 12.2.1 ist eine Domain nicht mehr die einzig mögliche Gruppierungsart, in der zusammengehörige Ressourcen gebündelt werden können. Innerhalb einer Domain können eine oder mehrere Partitionen angelegt werden. Dabei handelt es sich dann um eine Art Unterbereiche, die für sich alleine Applikationen und Ressourcen beinhalten können.

Partitionen sind eigene Teilbereiche einer Domain, die auch als slices bezeichnet werden. Und so kann man sich das Prinzip auch vorstellen – als eine Segmentierung oder Aufteilung einer Domain. Partitionen stellen eine eigenständige Laufzeitumgebung mit isolierter, unabhängiger Administration dar. Auch das Bild als eine Art „Mikro-Container“ passt als Anschauung für das neue Konzept ganz gut; Mikro-Container, von denen sich eine unbegrenzte Anzahl innerhalb einer Domain befinden kann (siehe Abbildung 1). Die Eigenständigkeit und die Isolation der Partitionen resultiert aus dem Sachverhalt,

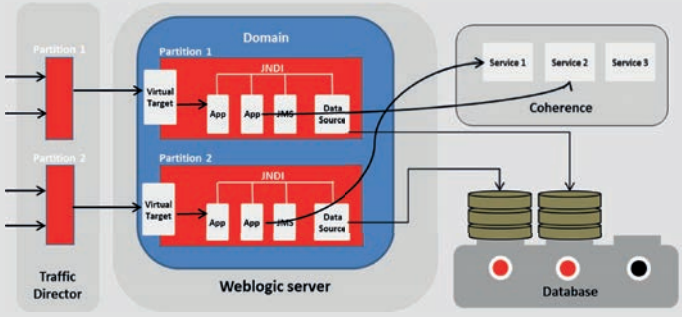


Abb. 1: Typische Aufteilung einer umfassenden WLS 12.2.1 Systemlandschaft. Gut zu erkennen sind im mittleren Bereich die zwei Partitionen innerhalb der Domain, auf die über sog. Virtual Targets zugegriffen werden kann. Die strikte Abschirmung der Partitionen untereinander kommt zum Ausdruck im Vorhandensein der separaten JNDI-Verzeichnisse.

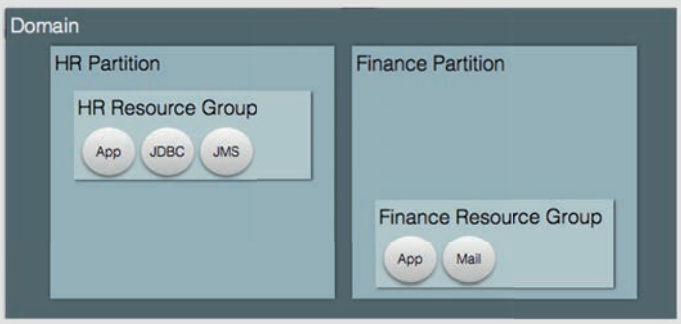


Abb. 2: Gruppierung von deploybaren, zusammengehörigen Artefakten (App, Treiber etc.) zu sogenannten Resource Groups.

dass jeder Partition sowohl eine eigene Sicherheitszone (realm) als auch ein eigener Namensraum (JNDI) zugeordnet ist.

Anwendungen brauchen ihr Revier

Wir betrachten nun eine Anforderung, die darin besteht, eine Anwendungsgruppe „Finanzen“ und eine „Personal“ den entsprechenden Bereichen eines Unternehmens als Dienst anzubieten. In WLS Version 12.1.3 und früher sah die typische Reaktion auf diese Anforderung so aus, dass für jeden Klienten (Unternehmensbereiche, die Dienste beanspruchen möchten) eine eigene Domäne zu spendieren war. Dadurch ähneln sich solche Domains sehr stark und adäquate Ansätze, um doppelte Administrationsarbeiten zu erleichtern und zu automatisieren konnten z. B. mittels Templates erreicht werden. Durch diese Schablonen lässt sich das Anlegen und Betreiben solcher gleichartigen Domains stark vereinfachen und vor allem weniger fehleranfällig gestalten.

Das resultiert in unserer Anwendung mit HR- und Finanzanteilen in separaten Domains für die unterschiedlichen Klienten, was zum Einsatz von Templates führt, die sich für HR- und Finanzanwendungen eigenen. Das könnte allerdings nicht ausreichend sein für solche Klienten, die sich an eine Domain anmelden und sowohl HR- als auch Finanzdienste nutzen wollen. Dafür müsste mit dem starren Konzept der Domains eine dritte Art geschaffen werden, was ein drittes Template erforderte.

Resource Groups und Resource Group Templates

Wir kommen nun zu wichtigen Konzepten, mittels derer sich die zuvor genannte Bereitstellung von Anwendungsteilen für bestimmte Zielgruppen in Partitionen bewerkstelligen lassen. Ein neues Konzept im Zusammenhang mit Multitenant ist der der Resource Group (RG). In einer RG werden „deploybare“ Dinge zusammengefasst, wie Anwendungen, passende Treiber (JDBC, JMS), spezielle Bibliotheken o. Ä. Eine RG lässt sich einer Partition zuordnen, was auch die Frage zum Teil beantwortet, wie sich Partitionen nutzen lassen. Der große Vorteil einer RG besteht in der Gruppierung von zusammengehörigen Bestandteilen einer Anwendung (z. B. Finanzanwendung, siehe Abbildung 2).

Und wie können wir nun elegant lösen, was das beschriebene Szenario verlangt? Dass nämlich ein Klient sowohl HR- als auch Finanzdienste nutzen möchte? Für die Antwort benötigen wir noch ein weiteres Konstrukt, und zwar das der Resource Group Template (RGT, siehe Abbildung 3). Zur Vermeidung von Redundanzen bei der Verwendung von Ressourcen, wie z. B. Anwendungsteile oder Treiber, können diese in RGTs definiert werden. Eine Resource Group, die diese Bestandteile nutzen möchte, braucht lediglich eine Referenzierung auf das passende RGT vorzunehmen. Somit müssen keine doppelten Ressourcen vorgehalten werden (also zweimal die HR-Bestandteile als Kopien), sondern es ist nur zu gewährleisten, dass die Referenzierung in der RG korrekt aus-

geführt wird, dass z. B. korrekte URLs für JDBC-Verbindungen eingetragen sind.

Virtual Target – Ziele wünschenswert und hochwillkommen

Es fehlt aber noch etwas Entscheidendes … das spürt man förmlich. Der aufmerksame Leser könnte sich fragen: „Wie adressiere ich denn nun eine Partition? Gibt es eine spezifische URL? Oder wie wird es gemacht?“ Genau da setzt das Konstrukt Virtual Target (VT) ein, das uns mit diesen Möglichkeiten versorgt. Ein VT hat als Bestandteile (teilweise optional):

- Host-Name und Port
- optional URI
- Network access point und Channel
- protokollspezifische Konfigurationen
- Target cluster und Managed Servers

Das sind allesamt bekannte Begriffe im WLS-Kontext und sie erlauben es, Partitionen gebrauchsfertig zu gestalten. So gelangen Requests von Webanwendungen gemäß der Angaben Host-Name, Port, URI zum dedizierten Ziel, nämlich z. B. zu einem speziellen Servlet. Das steckt innerhalb einer deployten EAR, welche ihrerseits einer RG angehört, die einer Partition zugeordnet ist. Zugegeben etwas verwirrend, aber es erfüllt seinen Zweck: Requests von spezifischen Mandanten zielsicher auf spezifische Partitionen zu leiten.

Scope – Ich sehe nichts. Siehst du etwas?

Mit diesen neuen Dingen wie RGs und RGTs ergeben sich neue Fragestellungen, wie z. B. die nach dem des Scope von deployten Artefakten; also: Wo sind diese sichtbar und damit nutzbar? Beim Deployment bekommen wir vier Möglichkeiten an Zielen, die gewählt werden können:

- Global: Die Ressource ist in einer nicht partitionierten Domain nutzbar.
- RGT: Die Ressource steht allen RGs zur Verfügung, die auf die RGT referenzieren.
- RG (Partition): Die Ressource kann nur innerhalb der Partition mit der RG genutzt werden.
- RG auf Domainlevel: innerhalb der RG nutzbar, allerdings auf Domainlevel.

Das heißt auch, dass Applikationen oder Bibliotheken nicht auf mehrere Partitionen verteilt sein können – sie sind isoliert in den klaren Grenzen der Partition, was aber genau zu dem gewünschten Verhalten führt, vielen Mandanten unabhängige und sichere Dienste und Anwendung zur Verfügung zu stellen. Und das heißt Multitenancy.

Fazit

Liebling Kreuzberg beherrschte die „menschliche“ Mandantenfähigkeit perfekt. In der IT beherrschen wir diese nun dank der Partitionen im WebLogic Server 12.2.1 wesentlich eleganter als bisher. Sie haben nun einen Überblick bekommen, wie die Partitionen aufgebaut sind und wie man zu unabhängigen, lauffähigen Einheiten kommen kann (durch Virtual Targets, denen Resource Groups zugeordnet sind), die uns die Fähigkeit verleihen, mehrere Mandanten unabhängig und sicher mit Anwendungen und Rechnerkapazität zu versorgen. Dabei profitieren Sie zugleich von einer größeren Dichte und besseren Ressourcennutzung, weil es nicht länger nötig ist, ganze Domains zu definieren, um isolierte Umgebungen im WLS zu erhalten.

In unserem Seminar „WebLogic Server Administration“ bieten wir Ihnen eine Einführung in diese spannende, neue Thematik an und Sie können lernen, wie Sie WebLogic selbst einsetzen können. Für Fragen und Unterstützung zu diesem Thema stehen wir Ihnen selbstverständlich gerne zur Verfügung.

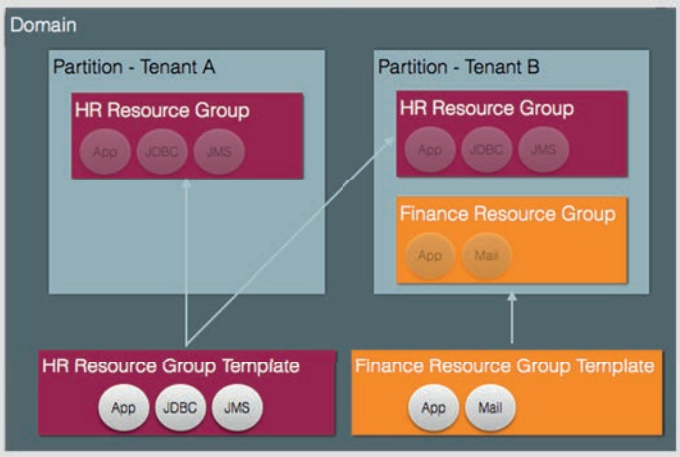


Abb. 3: Resource Group Templates (RGT) beinhalten die eigentlichen Ressourcen wie Applikationen, JDBC-Treiber o. Ä. Eine Resource Group in einer Partition referenziert eine solche RGT.

Links

[1] Oracle Weblogic Server Multitenant
https://docs.oracle.com/middleware/1221/wls/WLSMT/concepts.htm

[2] Oracle Fuion Middleware
https://docs.oracle.com/middleware/1221/wls/

Glossar

Multitenant
Mandantenfähigkeit; eine Anwendung oder ein System mit unterschiedlichen Klienten (Mandanten) so zur Verfügung stellen, dass diese unabhängig voneinander sind, sich nicht gegenseitig stören oder beeinflussen.

Dr. Hubert Austermeier
(info@ordix.de)

Flashback – Reise in die Vergangenheit:

Warum Oracle Zeitreisen anbieten kann, der Microsoft SQL Server aber nicht

Warum bietet Oracle die Funktionalität „Flashback-Query“ und der MS SQL Server nicht? Wie sich die beiden Datenbankmanagementsysteme in ihrer Arbeitsweise und Funktionalität unterscheiden, werden wir näher erläutern. Dabei werden auch die Begriffe Lesekonsistenz und Isolationslevel behandelt.

Wie sahen die Daten gestern aus?

Der Blick zurück wird bei Oracle durch die Erweiterung der Abfrage um den „AS OF“-Zusatz realisiert:

```
SELECT * FROM mitarbeiter AS OF TIMESTAMP
SYSDATE - 1;
```

Mit dieser Abfrage wird der Stand der Tabelle wiedergegeben, wie er 24 Stunden vor Beginn der Abfrage gültig war. Bereits an dieser Stelle sei erwähnt: Das klappt natürlich nicht in jedem Fall. Es müssen bestimmte Voraussetzungen erfüllt sein, die im Folgenden näher betrachtet werden.

Ablauf einer Transaktion

Bei einer Transaktion gilt das „Alles oder Nichts“-Prinzip. Es bedeutet, dass alle im Rahmen einer Transaktion durchgeführten Änderungen entweder vollständig umgesetzt oder komplett rückgängig gemacht werden. Die beiden wichtigsten Aspekte beim Ablauf einer Transaktion sind: Den Weg zurück sichern und die Änderung garantieren.

Um den Weg zurück zu sichern, wird zu jeder Datenänderung die entgegengesetzte Datenänderung generiert und gesichert. Bei Oracle spricht man vielfach vom „Before Image“, generell von Undo-Informationen. Um die Änderungen nach der Bestätigung des Commit zu

garantieren, wird vor dem Versenden dieser Bestätigung sichergestellt, dass alle Datenänderungen selbst gesichert wurden. Bei Oracle spricht man auch vom „After Image“, generell von Redo-Informationen.

Von der Verarbeitung dieser Informationen völlig unabhängig erfolgt die Übertragung der geänderten Daten aus dem Buffer Pool in die Datendatei zu einem anderen (meist später liegenden) Zeitpunkt. Diese Verarbeitung ist für die folgenden Überlegungen nicht relevant und wird daher nicht weiter betrachtet.

Technische Umsetzung

Beim MS SQL Server werden die Informationen zunächst im Hauptspeicher, genauer gesagt im Log Buffer gehalten. Dieser relativ kleine Bereich, mit wenigen MB, enthält die Redo- und die Undo-Informationen in Form von Statements. Er wird in kurzen Intervallen und auf jeden Fall beim Commit auf die Festplatte geschrieben. Dort landen die Informationen im Transaktionsprotokoll, einer Datei, die mitunter relativ groß werden kann. Denn je nach Konfiguration der Datenbank werden die einzelnen Bereiche dieser Datei (Virtual Log Files) erst dann zur Wiederverwendung freigegeben, wenn die Informationen durch ein Backup (genauer gesagt durch eine Transaktionsprotokollsicherung) verarbeitet wurden.

Auch Oracle besitzt einen Log Buffer mit der gleichen Funktionalität wie der MS SQL Server. Allerdings werden die Undo- und Redo-Informationen im Folgenden nicht gemeinsam verarbeitet. So besitzt Oracle eine zusätzliche Komponente: Den Undo Tablespace. Dieser enthält die Daten-Werte aus den Undo-Informationen, also die „Before Images“. Da es sich um einen Tablespace handelt, liegen die Informationen sowohl im Hauptspeicher als auch auf der Festplatte vor, alle Änderungen werden ebenfalls durch Redo-Informationen gesichert. Damit liegen diese Informationen auch nach dem Ende der Transaktion und sogar nach einem Neustart der Instanz weiterhin vor.

Die Redo-Informationen werden bei Oracle, ähnlich wie beim MS SQL Server, in Redo Log Files geschrieben. Oracle verwendet dabei mindestens zwei Dateien mit einer festen Größe von mindestens 4 MB. Ist die aktive Datei voll, wird diese geschlossen und die andere Datei zur aktiven Datei. Die geschlossenen Dateien werden vom Archiver in die Archive Destination kopiert und stehen dort für die Wiederherstellung der Datenbank zur Verfügung.

Wichtigster Unterschied zwischen den beiden Systemen und der Schlüssel zu Flashback-Query sind die längerfristig zugänglichen Undo-Informationen bei Oracle.

Lesekonsistenz

Diese Undo-Informationen werden bei Oracle dazu genutzt, lesenden Abfragen einen konsistenten Stand der Informationen zum Zeitpunkt des Beginns der Abfrage

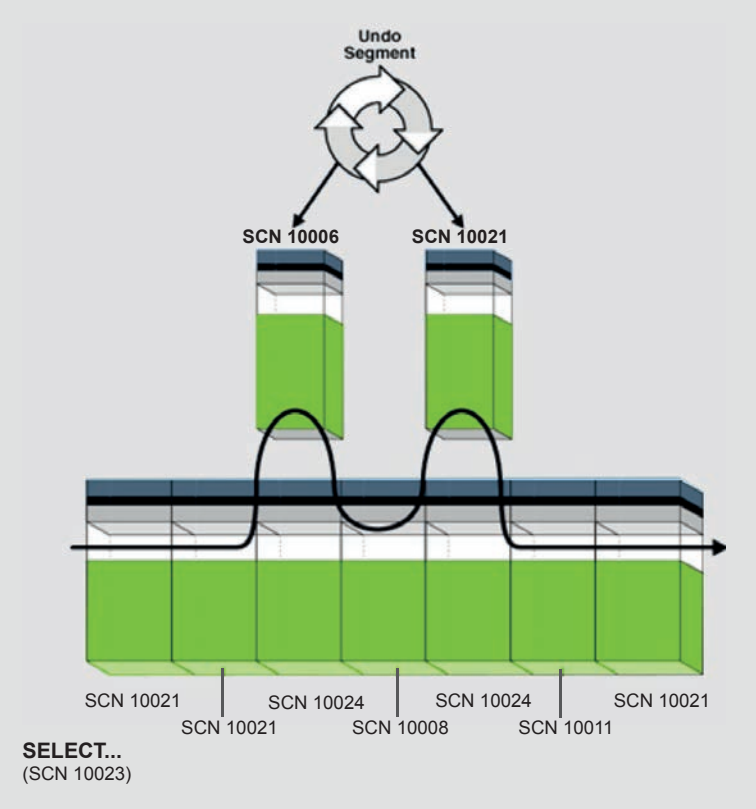


Abb. 1: Mit den Undo-Informationen können ältere Stände der Daten rekonstruiert werden.

zur Verfügung zu stellen. Dabei werden diese nicht von ändernden Zugriffen blockiert, da ältere Stände aus den Undo-Informationen rekonstruiert werden (Abbildung 1) – jedenfalls so lange, wie der Undo Tablespace groß genug ist, um alle benötigten Informationen aufzunehmen.

Beim MS SQL Server richtet sich das Verhalten nach dem verwendeten Isolationslevel. Beim Standard „Read Committed“ erzeugen ändernde Abfragen Sperren auf den betroffenen Datensätzen. Die lesenden Zugriffe werden dadurch blockiert und müssen auf das Ende der schreibenden Transaktion warten.

Die alternativen Isolationslevel „Snapshot“ und „Read Committed Snapshot“ verändern das Verhalten und ermöglichen, wie bei Oracle, einen lesekonsistenten gleichzeitigen Zugriff. Erreicht wird dies durch den Aufbau von Zeilen-Versionen in der tempdb. Die Aktivierung dieser Isolationslevel ändert jedoch den Aufbau der Daten in den Dateien, da in jeden Datensatz ein zusätzlicher 14 Bytes großer Zeiger zu den Zeilen-Versionen eingefügt wird.

Ein entscheidender Unterschied zu Oracle bleibt: Die Zeilen-Versionen sind nur bis zum Neustart der Instanz verfügbar. Ein Bereinigungsprozess entfernt bereits im Vorfeld die nicht mehr benötigten Zeilen-Versionen von laufenden Transaktionen.

Glossar

Isolation
Die Trennung von Transaktionen auf eine Weise, dass eine laufende Transaktion nicht von einer parallel ablaufenden Transaktion durch Änderung der benutzten Daten in einen undefinierten Zustand gebracht werden kann.

Isolations-Level
Der Isolation-Level bestimmt, wann das Datenbanksystem welche Sperren setzt. Damit wird festgelegt, welche Abfragen parallel aufgeführt werden können und welche Abfragen auf das Ende von anderen Abfragen warten müssen.

Redo Log-Dateien
Redo Log-Dateien speichern Informationen über alle Datenänderungen in den Tablespace-Dateien. Sie können im Falle eines Speichermedium-Fehlers zur Rekonstruktion der Daten herangezogen werden. Es müssen mindestens zwei Redo Log-Dateien (mindestens zwei Gruppen mit jeweils mindestens einer Datei) für jede Database definiert sein.

Quellen

[Q1] Oracle Database concepts:
http://docs.oracle.com/database/121/CNCPT/consist.htm#GUID-00A3688F-1219-423C-A5ED-4B8F25BEEAFB__BABFDBAJ

Bildnachweis

Andreas Jordan
(info@ordix.de)

Flashback-Query

Warum ist die Dauer der Speicherung von Undo-Informationen so wichtig? Weil Flashback-Query nichts anderes ist als eine konsequente Erweiterung der Lesekonsistenz in die Vergangenheit.

Mithilfe der Undo-Informationen kann der Zustand der Daten nicht nur zum Zeitpunkt des Beginns der Abfrage, sondern auch zu anderen Zeitpunkten in der Vergangenheit bereitgestellt werden.Voraussetzung ist natürlich, dass diese Informationen noch im Undo Tablespace vorliegen.

Konfigurationsparameter

Zwei Konfigurationsparameter haben direkten Einfluss auf die Verfügbarkeit von Flashback-Query. Zum einen die Option `UNDO_RETENTION` für das System. Diese Angabe in Sekunden bestimmt, wie lange die Undo-Daten nach Ende einer Transaktion vorgehalten werden. Dieser Wert gilt nur für automatisch vergrößernde Undo Tablespaces. Des Weiteren darf die `MAXSIZE` nicht erreicht werden.

Zum anderen kann die Option `RETENTION GUARANTEE` für den Undo Tablespace die unbedingte Einhaltung der `UNDO_RETENTION` garantieren. Dies kann jedoch zu Abbrüchen von schreibenden Transaktionen führen, wenn nicht ausreichend Speicher vorhanden ist.

Fazit

Die getrennte und persistente Speicherung der Undo-Daten ist der Schlüssel zur Zeitreise. Anhand dieses Features zeigt sich auch, welche Auswirkung die unterschiedliche Architektur der Datenbankmanagementsysteme auf die Verarbeitung von Abfragen hat.

Noch ein abschließender Hinweis: In Produktionsumgebungen wird Flashback-Query sicherlich sehr selten eingesetzt. Für Entwicklungen und Tests kann das Feature jedoch durchaus sinnvoll eingesetzt werden.

Oracle TEXT (Teil II)

New Features Oracle TEXT 12c

In der letzten Ausgabe der ORDIX® news wurde die Volltextsuche mit Oracle TEXT vorgestellt. In diesem Artikel werden neue Performance Features von Oracle TEXT in der Oracle-Version 12c gezeigt.

Performance beim Index anlegen

Das Anlegen eines Oracle TEXT Index kann unter Umständen sehr lange dauern. Ein Grund dafür kann die Default-Memory-Einstellung sein, die lediglich bei 64 MB liegt. Mit einer Vergrößerung der Default Memory-Einstellung kann die Performance beim Anlegen des Oracle TEXT Index verbessert werden.

In Oracle 11g lag die maximal wählbare Memory-Einstellung **MAX_INDEX_MEMORY** bei 2 GB. Mit Oracle 12c wurde **MAX_INDEX_MEMORY** auf 256 GB erweitert. Abgefragt werden können die Memory-Einstellungen über die Tabelle **CTX_PARAMETERS** (siehe Abbildung 1).

Die Memory-Größe kann im **CREATE INDEX** Statement oder über die Änderung der Parameter mit dem Package **CTX_ADM** eingestellt werden. Das Beispiel in Abbildung 2 zeigt die Nutzung beim Anlegen des Index und die Änderung der Default-Einstellungen über das Package **CTX_ADM**. Der Memory-Parameter sollte - unter Berücksichtigung des dem System zur Verfügung stehenden Arbeitsspeichers - so hoch wie möglich gesetzt werden.

Near Real Time Index

Im Gegensatz zu einem normalen Index ist ein Oracle TEXT Index bei Änderungen auf der Basistabelle nicht automatisch aktuell. Immer häufiger gibt es die Anforderung, den Oracle TEXT Index auch bei großer Änderungsrate auf der Basistabelle aktuell zu halten. Dies kann man mit einem manuellen **SYNC**-Befehl erreichen. Die Aktualität des Index ist dann abhängig von der Häufigkeit der manuellen Synchronisation. Mit der Option **SYNC ON COMMIT** kann beim Erstellen des Oracle TEXT Index dafür gesorgt werden, dass die Änderungen auf der Basistabelle sofort in den Oracle TEXT Index eingepflegt werden. Der Nachteil dieser beiden Möglichkeiten ist jedoch, dass die neuen Änderungen in den Index einfach nur in der **$I**-Token-Tabelle angefügt und nicht in der richtigen Reihenfolge sortiert abgelegt werden. Dies hat eine starke Fragmentierung des Index zur Folge und führt zu einer schleichenden Performance-Verschlechterung.

Mit dem neuen Feature „Near Real Time Index“ führt Oracle in 12c einen zweistufigen neuen Oracle TEXT Index ein. Neben der Haupt-Token-Tabelle **$I** werden geänderte Daten in der Token-Tabelle **$G** abgelegt. Beim Suchen über den Oracle TEXT Index werden beide Token-Tabellen **$I** und **$G** durchsucht. Die **$G**-Tabelle ist in der Regel sehr klein und liegt daher in der SGA und kann auch im Keep-Pool gehalten werden. Für das Erstellen eines „Near Real Time Index“ muss die neue Storage-Option **STAGE_ITAB** verwendet werden (siehe Abbildung 3).

In einem Maintenance-Job müssen irgendwann die Token aus der **$G**-Tabelle in den Oracle TEXT Index (**$I**-Tabelle) eingepflegt werden. Dies geschieht mit dem **Merge**-Befehl. Danach sollte mit einem Optimize-Job die $I-Tabelle „rebuild“ werden (siehe Abbildung 4).

BIG_IO-Option

Zur Erinnerung: In der Token-Tabelle **DR$<Indexname>$I** des Oracle TEXT Index stehen die Tokens/Wörter und Informationen über deren Vorkommen in den Dokumenten bzw. Tabellen. Diese Informationen werden in der Spalte **Token_Info** als **BLOB** abgespeichert. In dieser Spalte werden standardmäßig maximal 4000 Bytes gespeichert. Häufig vorkommende Wörter werden daher mehrfach in die Tabelle **DR$<Indexname>$I** abgespeichert.

```
select * from ctx_parameters where PAR_NAME like '%MEMORY%';

PAR_NAME                       PAR_VALUE
------------------------------ ------------------------------
DEFAULT_INDEX_MEMORY           67108864
MAX_INDEX_MEMORY               274877906944
```

Abb. 1: Memory-Einstellungen

```
Index Memory beim Create Index:

CREATE INDEX idx_tab ON tab(text)
  INDEXTYPE IS CTXSYS.CONTEXT
  PARAMETERS ('memory 3072M');

Änderung der Default Memory - Einstellung:

begin
ctxsys.ctx_adm.set_parameter('DEFAULT_INDEX_MEMORY','3072M');
end;
/
```

Abb. 2: Memory-Syntax

```
Storage Option „Near Real Time Index“:

begin
ctx_ddl.drop_preference ('mystorage');
ctx_ddl.create_preference ('mystorage', 'BASIC_STORAGE');
ctx_ddl.set_attribute ('mystorage', 'STAGE_ITAB', 'true');
end;
/

Create Index „Near Real Time Index“:

CREATE INDEX idx_tab ON tab(text)
  INDEXTYPE IS CTXSYS.CONTEXT
  PARAMETERS ('storage mystorage sync (on commit)');

Token-Tabelle/Index $G in den Keep-Pool legen:

exec ctx_ddl.set_attribute ('mystorage','G_TABLE_CLAUSE',
   'storage (buffer_pool keep) )';
exec ctx_ddl.set_attribute ('mystorage','G_INDEX_CLAUSE',
   'storage (buffer_pool keep) )';
```

Abb. 3: Create Near Real Time Index

```
begin
ctx_ddl.optimize_index('idx_tab','MERGE');
ctx_ddl.optimize_index('idx_tab','REBUILD');
end;
/
```

Abb. 4: Maintenance Near Real Time Index

```
begin
ctx_ddl.create_preference( 'mystorage', 'BASIC_STORAGE' );
ctx_ddl.set_attribute ( 'mystorage', 'BIG_IO', 'true' );
end;
/
```

Abb. 5: BIG_IO-Option

```
Begin
ctx_ddl.create_preference('mystorage', 'basic_storage');
ctx_ddl.set_attribute('mystorage', 'query_filter_cache_size',
'50M');
end;
/
```

Abb. 6: Query Filter Cache

```
Syntax: ctxfiltercache((query_text) [,save_score][,topN])
Beispiel:
select * from tab where contains(dokument, 'ctxfiltercache((-
Meier), true, true)') > 0 ;
```

Abb. 7: Query-Filter-Cache-Abfrage

Links

[1] ORDIX® news Artikel 1/2016
„Suchen Sie noch oder finden Sie schon? - Die Volltextsuche mit Oracle TEXT“
http://ordix.de/ordix-news-archiv/1-2016.html

Mit der **BIG_IO**-Option wird in 12c, unter Verwendung von Securefiles mit großen Chunks, die Grenze von 4000 Bytes aufgehoben. Es werden nun gleiche Wörter nicht mehrfach, sondern weniger Einträge pro Token in der **$I**-Tabelle gespeichert. Die Byte-Größe pro Eintrag in der Token-Tabelle wird zwar größer, aber die Anzahl der Einträge wird deutlich reduziert. Der Oracle TEXT Index wird dadurch kleiner und beim Lesezugriff müssen weniger Indexsegmente gelesen werden. Die **BIG_IO**-Option wird als Storage-Attribut konfiguriert wie in Abbildung 5 zu sehen ist.

Query Filter Cache

Der Query Filter Cache ist ein Ergebnis-Cache für Oracle TEXT Abfragen. Die Ergebnisse dieser Abfragen werden in einen separaten Speicherbereich abgelegt und können wiederverwendet werden. Mit einer Storage Preference kann der Query Filter Cache aktiviert werden. Der Index wird dann mit dieser Storage Preference angelegt (siehe Abbildung 6).

Der Query Filter Cache kann jedoch nur für einfache Abfragen eingesetzt werden. Eine Progressive-Relaxion-Abfrage (mehrere Abfragen auf einmal) mit dem Query Filter Cache zu kombinieren ist nicht möglich. In der SQL-Abfrage muss der Query Filter Cache explizit angesprochen werden. Mit den zusätzlichen Optionen **save_score** oder nur **topN by score** kann der Score ebenfalls im Cache berücksichtigt werden (Abbildung 7). Ob der Query Filter Cache verwendet wurde, kann in der Tabelle **CTX_FILTER_CACHE_STATISTICS** abgefragt werden.

Fazit

Oracle TEXT ist eine vollwertige Volltextsuchmaschine, die kostenfrei in jeder Oracle Edition verfügbar ist. Mit den neuen Performance Features in der Oracle-Version 12c wurden sinnvolle Verbesserungen implementiert.

Weitere Oracle TEXT 12c Features lernen Sie in unserem Oracle TEXT Seminar kennen. Gerne unterstützen wir Sie auch bei Ihrem Oracle TEXT Projekt.

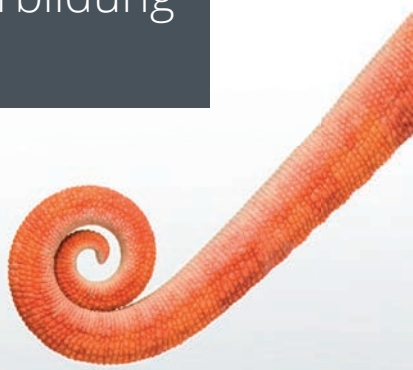
Michael Skowasch
(info@ordix.de)

ORDIX® seminare

einfach. gut. geschult.

Neues Seminarprogramm 2017

Haben Sie sich schon Gedanken über Ihre Weiterbildung im kommenden Jahr gemacht?



Wir haben in unserem neuen Seminarprogramm einige neue Seminare für Sie zusammengestellt. Mit Sicherheit ist auch etwas für Sie dabei!

Neu im Programm:

- IT-Sicherheit für Projektmanager und IT-Leiter
- IT-Organisation
- Social Skills für IT-Consultants
- Verhandlungstechniken
- Einführung in NoSQL-Datenbanken
- PostgreSQL-Administration
- Einstieg in JavaFX (Seite 99)
- Continuous Integration (CI) Workshop
- Windows 10 für Administratoren
- Microsoft Hyper-V-Workshop
- Microsoft Hyper-V-Deep-Dive
- PerformanceGuard
- Backup und Recovery Oracle on Netapp

Besuchen Sie unsere Internetseite seminare.ordix.de

Wir gestalten Zukunft in der IT

Duales Studium in Kooperation mit vier Fachhochschulen

Entlang der Thematik „Der Mensch als Erfolgsfaktor“ bilden wir jährlich mehr als zwölf Studenten in Kooperation mit vier Fachhochschulen aus. Wissen zu teilen und der offene Dialog untereinander sind bei uns gelebte Unternehmenskultur. Denn getreu dem Motto „Wissen vermehrt sich, indem man es teilt“, geben wir unser Fachwissen und unsere Praxiserfahrungen weiter. An Studenten, Praktikanten und natürlich an Absolventen.

Mit unterschiedlichen dualen Studiengängen investieren wir in die Ausbildung junger Nachwuchskräfte, die eine Alternative zum klassischen Hochschulstudium suchen. Die Vorteile liegen auf der Hand: Schon während des gesamten Studiums bekommen Sie einen intensiven Einblick in die Praxis und werden langsam in das „Arbeitsleben“ eingeführt. Wir bieten Ihnen als Studierenden umfangreiche Möglichkeiten, schon während des Studiums theoretisches Wissen mit praktischer Erfahrung zu verknüpfen.

Die Studieninhalte werden durch IT-Seminare mit erfahrenen Trainern in unserem öffentlichen Seminarzentrum in Wiesbaden ergänzt.

Mit dem Wechsel von Theoriephasen an der Hochschule und Praxisphasen sichern Sie sich beste Chancen auf dem Arbeitsmarkt. Sie arbeiten eng mit unseren Fachabteilungen und unseren Kunden zusammen.

Die Praxis-, Bachelor- oder Masterarbeit erstellen Sie in Zusammenhang mit realen Projekten. So qualifizieren Sie sich bestens für das Berufsleben.

Berufliche Perspektiven

Unser Ziel ist es, Sie nach Ihrem Studium direkt bei uns in Vollzeit anzustellen. Nach Ihrem Studienabschluss wartet auf Sie ein Arbeitsplatz mit vielen interessanten Aufgaben - wahlweise an den Standorten Paderborn, Köln, Wiesbaden oder Augsburg.

Ihre Übernahme in ein Beschäftigungsverhältnis können wir zwar nicht schon zu Beginn der Ausbildung garantieren,

aber so viel ist sicher: Wenn Sie uns mit ansprechenden Leistungen überzeugen, dann wird Ihrem „Job fürs Leben“ bei der ORDIX AG nichts im Wege stehen.

Optional unterstützen wir Sie nach dem Bachelor-Studium mit einem berufsbegleitenden Master-Studiengang. So schlagen Sie eine ideale Brücke zwischen akademischem Wissen und berufspraktischem Know-how.

Studiengänge

Wir bieten in Kooperation mit den Hochschulen folgende duale Studiengänge für die Standorte Paderborn, Köln und Wiesbaden an:

- Duales Studium zum Bachelor of Science in Angewandte Informatik (w/m) (Paderborn)
- Duales Studium zum Bachelor of Science in IT-Sicherheit (w/m) (Wiesbaden)
- Duales Studium zum Bachelor of Science in Wirtschaftsinformatik (w/m) (Paderborn, Wiesbaden, Köln)
- Duales Studium zum Bachelor of Science in Informatik (w/m) (Wiesbaden)
- Duales Studium zum Master of Science in Informatik (w/m) (Wiesbaden)

Hochschulpartner

h_da HOCHSCHULE DARMSTADT UNIVERSITY OF APPLIED SCIENCES fbi FACHBEREICH INFORMATIK

FHDW Fachhochschule der Wirtschaft

BA Hessische Berufsakademie University of Cooperative Education

ba BerufsAkademie Rhein-Main University of Cooperative Education

Mehr Informationen und Informationen zum nächste Studienjahr!

http://ordix.de/karriere/

Erfahrungsbericht aus dem dualen Studium bei ORDIX

Veronika, Duales Studium, Bachelor of Science, FHDW Paderborn, Geschäftsstelle Paderborn

"Ich mache meinen Bachelor of Science in Wirtschaftsinformatik dual an der Fachhochschule der Wirtschaft (FHDW) in Paderborn. Das Studium wechselt zwischen Theorie- und Praxisphasen, die jeweils drei Monate lang sind.

An der Fachhochschule werden neben den klassischen Unternehmensfunktionen wie Marketing, Rechnungswesen und Vertrieb Themen wie Projektmanagement, Unternehmensführung, Analyse und Optimierung von Geschäftsprozessen sowie Anwendungsentwicklung im betriebswirtschaftlichen Umfeld behandelt. Dabei sind die Vorlesungen sehr praxisorientiert und die Dozenten bringen häufig treffende Beispiele aus der eigenen Berufserfahrung mit.

Die Praxisphasen finden dann bei ORDIX statt. Dort bekommt man einen guten Einblick in verschiedenste Bereiche (Entwicklung, Datenbanken, Betriebssysteme etc.) und kann einige Themen aus der Theorie vertiefen und viele neue Kenntnisse erlangen. Man arbeitet meist selbstständig, wird aber immer gut betreut und hat genug Ansprechpartner, an die man sich bei Fragen oder Problemen wenden kann. Außerdem werden regelmäßig Seminare zur Weiterbildung angeboten, die alle Mitarbeiter noch mehr fördern.

Das duale Studium bei der ORDIX AG ist durch den Wechsel aus Theorie und Praxis und der individuellen Gestaltung der Praxisphasen spannend und abwechslungsreich. Das gute Miteinander der Kollegen und die lockere Atmosphäre im Unternehmen runden dies noch zusätzlich ab, sodass ich mich jederzeit wieder für das Studium bei der ORDIX AG entscheiden würde!"


Ein frohes Weihnachtsfest
verbunden mit den besten
Wünschen für das neue Jahr 2017
Spenden statt Geschenke - auch in diesem Jahr unterstützen
wir folgende Organisationen mit einer Spende:
www.sterntaler-kinder.de
Sterntaler
Hilfe für schwerkranke Kinder e.V.
Bärenherz
Stiftung
für schwerstkranke Kinder
Elisabethstift
Jugendhilfe im Diakonischen Werk