MarkLogic Vortrag: Heterogene Daten blitzschnell analysiert

Vortrag von Jochen Jörg Principal Sales Engineer MarkLogic
München, 28. Oktober 2015

Slide 1

Laut Gartner lässt sich die heutige Datenlandschaft in verschiedene Katagorien unterteilen. Dies sind hochstrukturierte Daten, semistrukturierte und unstrukutierte Daten. Für hoch strukturiete Daten haben sich relationale Datenbanksysteme bewährt. Unstrukturierte Daten werden in der Regel von Suchmaschinen bearbeitet. Für semisrukturiete Daten gibt es nur sehr wenige Lösungen, die sich für unternehmenskritische Anwendungen eignen. Marklogic als Enterprise NoSQL Datenbank adressiert das komplette Datenspektrum.

Slide 2

Bevor wir uns eine Beispielanwendung im Detail betrachten, möchte ich gerne einen Überblick über Marklogic geben. MarkLogic vereint die Funktionalitäten einer Multimodeldatenbank, einer Suchmaschine und einer Plattform in einem Produkt. D. h. man muss sich in der Applikationsschicht nicht mehr um die Integration dieser verschiedenen Aspekte kümmern und kann sich statt dessen vollends auf die Lösung des Geschäftsproblemes fokussieren.

Slide 3

MarkLogic ist eine dokumentenorienterte Datenbank, d. h. die Datensätze werden in Form von Dokumenten verwaltet. Diese Dokumente können nahezu beliebige Formate haben (Jaxon, Binary, XML, HTML, Text). Analog zu einer relationalen Datenbank, in der ein Datensatz verschiedene Felder besitzt, sehen wir ein Dokument als Informations-Container. Mit dem Unterschied, dass nicht jeder Informationskontainer einem vorgegebenen Struktur entsprechen muss. Mit MarkLogic ist es möglich auf diesen Informationscontainer in beliebiger Granulatirät zuzugreifen – lesend und schreibend. Desweiteren lassen sich komplexe Suchanfragen oder Queries Analysen, analytische Queries und Volltextsuchen beliebig kombinieren. Die Ergebnisse der Anfragen können in unterschiedlichen Ausgabeformaten bereitgestellt werden. MarkLogic garantiert, dass sämtliche lesenden und schreibenden Zugriffe transaktional sind und gewährleistet, dass die Daten für jede Anwendung bereitgestellt werden. Ein Hauptmerkmal von MarkLogic ist die Tatsache, dass Marklogic schema-agnostisch ist. Das bedeutet, dass mann kein Datenmodell definieren muss, bevor man Daten in MarkLogic importiert. Das Datenmodell in MarkLogic ist somit dynamisch, es leitet sich von der Struktur der importierten Daten ab. Diese Eigenschaft ermöglicht es in der Datenschicht sehr flexibel mit Daten und Inhalten umzugehen und dadurch in kurzer Zeit auf Änderungen und neue Anforderungen umzusetzen oder einzugehen.

Slide 4

Hier sehen Sie ein Beispiel (Slide 7). Ein Dokument hat eine hierarchische Struktur. Es enthält Attribute und Werte und kann auch Volltext enthalten. All diese Aspekte werden von MarkLogic automatisch indexiert und können beim Datenzugriff verwendet werden.

Neben den bereits erwähnten Eigenschaften kann MarkLogic auch orstspezifische Informationen und semantische Triples indexieren.

Slide 9

Der Datenintegrationsprozess mit MarkLogic beginnt typischerweise mit dem Laden von heterogenen Daten - in ihrem ursprünglichen Format. Wie bereits erwähnt, ist ein Datenmodell dabei nicht notwendig. Nach dem Datenimport können mit Hilfe von Werkzeugen, die MarkLogic bereit stellt, die Daten sofort durchsucht und inspiziert werden. Mit anderen Worten: MarkLogic hilft oft Kunden die Daten erstmal kennen zu lernen. Danach werden typischerweise die Daten konsolidiert, transformiert und in eine Form gebracht, die den Anforderungen der Anwendung entspricht. Diesen Prozess sehen wir als einen iterativen Prozess an. Er ermöglicht die agilie Entwicklung von Anwendungen.

Jetzt gehe ich näher auf die konkrete Beispielanwendung ein, die auf Basis von MarkLogic entwickelt wurde. Folgende Frameworks und Tools kommen hierbei zum Einsatz: Marklogic als Enterprise NoSQL Datenbank, Spring Boot, Spring MVC für die Middle-Tier, in der die Geschäftslogik implementiert ist. Thymeleaf als Templating Engine für die Generierung von Sichten in der Webapplikation. Wir setzen Gradle als Buildtool für das Set up und Deployment von MarkLogic ein. Das Populäre Framework Apache Camel verwenden wir für die Datenaggregation.

Demo einer Beispielanwendung

Gezeigt werden soll der Datenimport von JSON, XML, HTML-Dateien, die Informationen über die Fußballweltmeisterschaft enthalten. Darüber hinaus werden diese Daten konsolidiert und können mit Hilfe der Applikation analysiert werden. Im wesentlichen handelt es sich um Daten über die Spieler, Mannschaften und Spielpartien während des Turniers. Neben Suchen können auch analytische Abfragen durchgeführt werden.

Slide 15

Die Anwendugnsentwicklung betrachten wir als einen iterativen und agilen Prozess, der aus folgenden Schritten besteht: Datenimport, Datenkonsolidierung, Definition von speziellen Indexen für die Analyse, Webapplikationsentwicklung und User Interface. Dieser Prozess wird in der idealen Welt sequenziell durchlaufen. In der realen Welt ist es ein interativer Prozess, d. h. man muss alle diese Schritte mehrmals anwenden, um auf neue Herausforderungen regieren zu können. Mit einer relationalen Datenbank ist es aufwändig Datenmodellierungs- und Datenmigrationsschritte durchzuführen. Aufgrund der flexiblen Eigenschaften von MarkLogic ist dies mit der MarkLogic Technologie wesentlich einfacher und mit weniger Aufwand verbunden. Besonders bei Anwendungsfällen in denen man mit heterogenen und unterschiedlich strukturierten Daten konfrontiert wird, eignet sich MarkLogic sehr gut. Es gibt viele ähnliche Ansätze in der Welt der noSQL Technologien. Wir grenzen uns von diesen in zwei wesentlichen

Punkten ab: 1. MarkLogic bietet Suche, Datenbank und Semantic in einem Guss an. 2. Man muss keinen Kompromiss bezglich der Enterprise-Eigenschaften eingehen. D. h. Daten, die in MarkLogic abgelegt sind können für unternehmenskritische Applikationen benutzt werden. Genauso wie man das in der Welt der relationalen Datenbanken gewährleitet.

Slide 16
Die Architektur der Applikation beseht aus folgenden Komponenten: In der Client-Schicht verwenden wir HTML 5, Bootstrap, JavaScript, in der Middle Tier verwenden wir das Spring Framework. Wir definieren fachliche Services, über die man letzendlich auf MarkLogic zugreifen kann.

Im folgenden möchte ich den Multi model-Ansatz von MarkLogic beschreiben. MarkLogic bietet out of the Box die Funktionalitäten einer Suchmaschine. Diese bestehen aus Stemming, sprachspezifische Unterstützung, Ranking, Alerting und auch ortsspezifische Suche. MarkLogic ist ebenfalls ein semantischer Tripple Store. Mit Hilfe eines Tripples lassen sich Entitäten in Beziehung setzen. In der Form von Subjekt, Prädikat und Objekt. MarkLogic ermöglicht es nun Milliarden von semantischen Beziehungen zu speichern und auf diesen Beziehungen zu navigieren. Somit lassen sich Dokument-Metadaten mit LinkedData (private und öffentliche) z. B. Wikipedia miteinander verknüpfen. Diese Eigenschaft öffnet die Türe für komplett neue Lösungsansätze:

- Kontextbezogene Suche
- Verbesserte Erschließung von Information
- Verlinkung von Daten
- Automatische Herleitung neuer Fakten (Inferencing)
- Kontenxtbezogene Aggregation und Bereitstellung von Daten

Slide 22
Mit MarkLogic ist es somit möglich, Daten als Dokumente zu verwalten und hierbei bewusst zu „Denormalisieren“ Gleichzeitig ist es aber möglich Daten beliebig miteinander in Beziehung zu setzen. Diese Kombination ermöglicht mehr Modellierungsmöglichkeiten im Vergleich zum relationalen Ansatz.

Slide 23
Als Programmierschnittstellen stellt Marklogic eine Java API, eine NodeJS API und eine erweiterbare REST API zur Verfügung. Des weiteren kann man auch im „nur Lesezugriff“ per SQL auf Daten von MarkLogic zugreifen.

Slide 24
MarkLogic ermöglicht es Daten bitemporal zu verwalten. Darunter verstehen wir die Datenanalyse über zwei verschiedene und von einander unabhängige Zeitachsen. Erstens die „valid time“, zweitens die „recording time“. Damit ist es sehr einfach möglich zu ermitteln, welcher zeitabhängige Wert wann für einen bestimmten Zeitraum gültig war. Diese Eigenschaft ist besonders wertvoll bei Anwendungsfällen in denen man Entscheidungen aufgrund der Datenbank rekonstruieren muss (Auditing). Ein weiterer Anwendungsfall ist die Historisierung zeitabhängiger Daten. Z. B. wann ist ein bestimmer Vertrag, Aktienkurs gültig, wann wurde eine Änderung dieser zeitabhängigen Daten vorgenommen.

MarkLogic skaliert horizontal. Ein Cluster kann bei Bedarf ohne Downtime erweitert werden oder verkleinert werden. Dies kann sowohl bei Zunahme des Datenvolumens als auch zunehmende Last auf das System erfolderlich sein. Wir sprechen in diesem Zusammenhang von Elastizität.

Slide 27
Die Eigenschaften von MarkLogic lassen sich in funktional und nicht funktional kategorisieren. Über die Jahre hinweg ist MarkLogic zu einem Produkt gereift, das Lösungen für unterschiedliche Domänen (Verlag, Medien, Versicherungen, Banken, Behörden) sicher und zuverlässig bereitstellt.

Slide 28
Hier sehen sie einen Auszug unserer Kunden:
BBC, ISO, Wiley, Warner Bros., Springer, McGraw Hill, Sony, Thieme, Deutsche Bank, Bank of America, UBS, City...

Diese setzen MarkLogic wie folgt ein

- Aggregieren und Konsolidieren
- Effektiver Zugriff auf Inhalte
- Flexibilität und Anpassungsfähigkeit in der Datenhaltung
- Neue Geschäftsmodelle
- Datenanalyse
- Vereinfachung von Infrastruktur und Betrieb

Für Fragen stehe ich Ihnen jederzeit gerne zur Verfügung. Meine Kontaktdaten:

Jochen.joerg@marklogic.com