Abfalldaten statt Datenabfall: So machen Sie Ihre Daten durch die neuen Big-Data-Technologien verfügbar

Der Einstieg in Big Data: In kleinen Schritten zu großen Daten

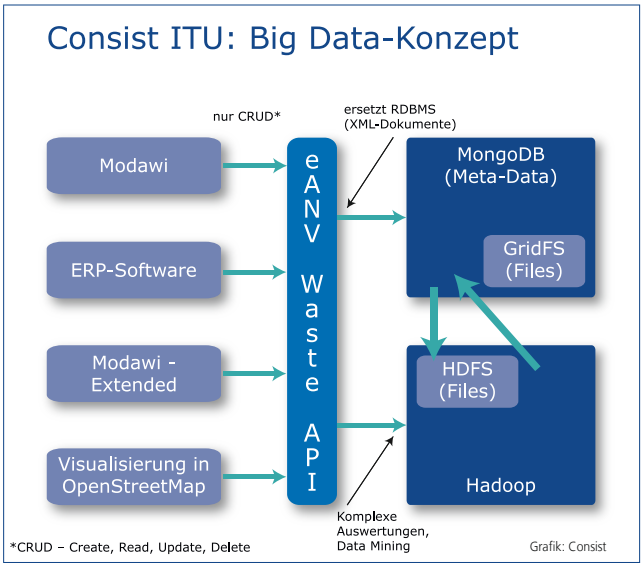
Von Ute Müller

Wollen Sie mehr wissen? Brauchen Sie schnellere Antworten? Dann kommt eine gewöhnliche relationale Datenbank an ihre Grenzen, denn um mehr aus Ihren Daten herauszuholen, müssen Sie sie möglichst „roh" speichern. Kundenportale, Wiegesysteme, die elektronische Nachweisführung und der steigende Telematikeinsatz in der Abfallwirtschaft erzeugen große Mengen an völlig unterschiedlich strukturierten Daten. Diese müssen einfach und dauerhaft abgelegt und für Ihren Geschäftserfolg ausgewertet werden. Big Data ist die Lösung, aber was heißt das konkret?

Wozu Big Data?

Im elektronischen Abfallnachweisverfahren werden XML-Dokumente mit qualifizierten elektronischen Signaturen und viele andere Dokumente bewegt und gelagert. Zustellquittungen (quasi Einschreiben-Rückscheine) und fachliche Quittungen über Fehler und fehlende Unterlagen fallen bei jedem Transport gefährlicher Abfälle mehrfach an. Bisher wurden die signierten Dokumente in den Blob-Feldern der relationalen Datenbanken zusammen mit bestimmten Meta-Daten in Feldern gespeichert. Aus den Quittungen wurden die – nach damaligem Kenntnisstand – relevanten Daten in die Tabellen übernommen, der Rest wurde gelöscht.

Recherchen über den Lebensweg eines Dokumentes oder Auswertungen über solche XML-Dateiinhalte, die nicht in die Meta-Daten übernommen wurden, waren damit nicht möglich (siehe auch Connect Artikel April 2013). Außerdem erwies sich die Speicherung von großen Mengen an XML-Dateien in den klassischen Datenbanken zunehmend als



Consist ITU setzt bei ihrer Big Data-Lösung auf OpenSource, und zwar die bekannte MongoDB und das Hadoop-Ökosystem. Der Zugriff erfolgt über das eANV-Waste-API.

problematisch. Die Lösung für diese Aufgabenstellungen ist „Big Data“, nur stellt sich die Frage, was genau ist das und was wird gebraucht?

Welche NoSQL-Datenbank ist die richtige?

Es gibt viele NoSQL-Datenbanken und dazu viele mehr oder weniger passende Funktionsbibliotheken, die alle ihre Vor- und Nachteile haben. Wie wählt man die richtige für eine Fragestellung aus? Wir haben in Zusammenarbeit mit dem HITeC e.V (Hamburger Informatik Technologie-Center am Fachbereich Informatik der Universität Hamburg; s. Kasten Seite 25) dieses Thema diskutiert. Abfallwirtschaftliche Usecase-Beschreibungen, nicht-funktionale Anforderungen, gemeinsame Workshops und die Expertise des HITeC e.V. führten zu der Empfehlung in der Abbildung Systemarchitekturen. Favorit ist natürlich die Lösung mit dem Rang 1: Das skizzierte System besteht aus drei Komponenten, u. a. weil nur so die unterschiedlichen Anwendungsszenarien erfüllt werden können.

Die MongoDB ist die allen Szenarien gemeinsame NoSQL-Datenbank, die je nach Anwendungsfall um die beiden weiteren Komponenten ergänzt wird. Hadoop ist ein Framework für skalierbare, verteilt arbeitende Software (s. http://de.wikipedia.org/wiki/Apache_Hadoop) und stellt im Wesentlichen eine Implementierung der MapReduce-Algorithmen von Google dar. Ein weiterer wichtiger Bestandteil des Hadoop-Ökosystems ist HDFS, ein Dateisystem, das sehr große Datenmengen redundant auf mehreren Rechnern speichert.

Systemarchitekturen

Rang	System	Kommentar
1	MongoDB + HDFS+ Hadoop	■ Metadaten: MongoDB (mächtige Querys), Dateien HDFS ■ Hadoop und HDFS für kleines Deployment optional → GridFS ■ Für "Low Latency": GridFS + ETL
(1)	Dynamo DB +S3+ EMR	■ Vollständig Cloud-basiert, pay-as-you-go ■ Administration und initiales Know-how sehr vereinfacht ■ DynamoDB Datenmodellierung berücksichtigt Querys
2	HBase + HDFS + Hadoop	■ Datenmodell berücksichtigt Querys, Dokumente in HDFS ■ Ein Ökosystem, gute MapReduce Integration, komplexe Administration ■ Nicht geeignet für kleines Deployment
3	DB + Riak + Hadoop	■ Vorhandenes RDBMS kann genutzt werden ■ Hadoop für kleines Deployment optional ■ Wenn Riak für Metadaten → Query-Materialisierung

Abbildung Systemarchitekturen: Es besteht ein klarer Favorit für die Lösung.

- **Standardszenario:** Moderate Datenmengen (Gigabyte-Bereich) und Standardauswertungen, Speichern der Dateien im MongoDB-eigenen GridFS
- **Mittleres Szenario:** Moderate Datenmengen und individuelle Auswertungen: Wie Standard, dann transformieren in einem ETL nach HDFS, dort MapReduce-Auswertung und Rückgabe der Ergebnisse
- **Umfassendes Szenario:** Große Datenmengen (Terabyte-Bereich) und individuelle Auswertungen; Speichern der Dateien im HDFS, Erstellung von spezifischen MapReduce-Auswertungen

Wie testet man Big Data?

Um die besonderen Aspekte einer Big Data Architektur in der Entwicklung oder zu Demonstrationszwecken darstellen zu können, empfiehlt es sich, tatsächlich mit unterschiedlichen Rechnern statt mit VM's zu arbeiten. Eine kostengünstige, platzsparende und trotzdem ausreichend leistungsfähige Umgebung lässt sich z. B. mit Cubietrucks (s. Foto rechts) errichten. Systeme, die wie die Cubietrucks auf ARM-Socs basieren, versprechen energieeffizienten Betrieb bei moderaten Investitionskosten. Dies dürfte gerade bei einem Szenario, das auf Skalierung durch zusätzliche Rechner basiert, ein gewichtiges Argument darstellen. Unsere Umgebung umfasst derzeit drei Rechner mit der MongoDB und drei Rechner mit HDFS und Hadoop, jeweils mit 750 GB Festplattenkapazität. Da die Anwendungen vollständig skalieren, können bei Bedarf weitere Rechner hinzugefügt werden.

Ein API für Abfallwirtschaft und Umweltschutz

Eine Schnittstelle mit passenden Funktionen, ein API, erlaubt den Zugriff auf die NoSQL-Datenbank. Die Funktionen zur Ablage von Dokumenten (XML, PDF, Log-Dateien, etc.) berücksichtigen die speziellen Anforderungen der abfallwirtschaftlichen Daten. Das aufrufende ERP-System übergibt nur die Daten und Dokumente und muss selbst nichts über

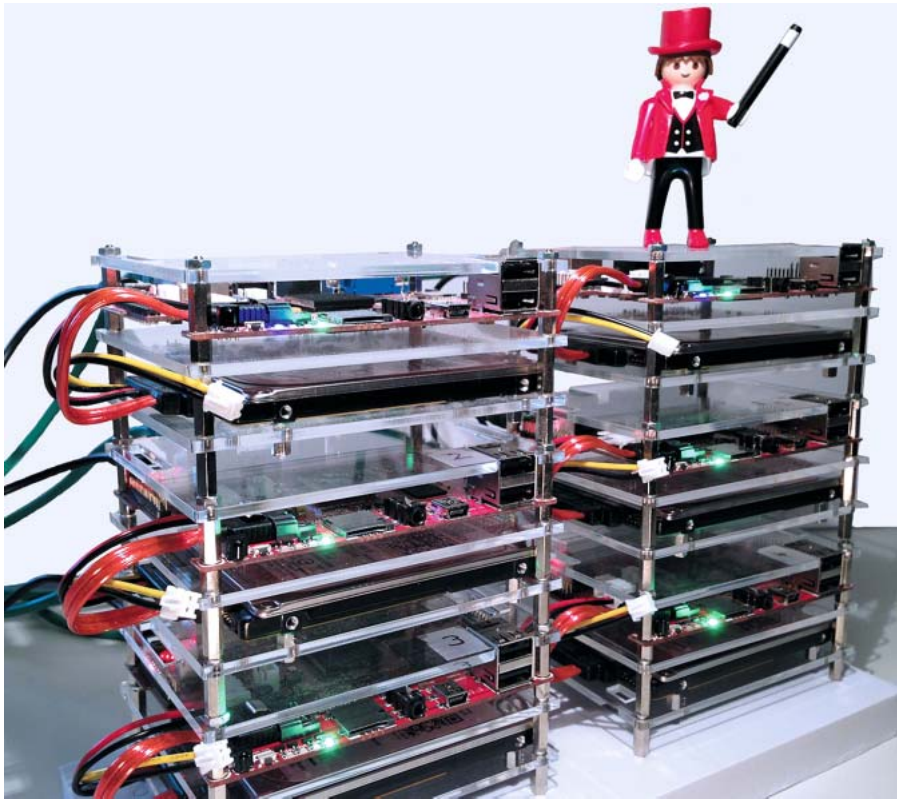


Foto: Consist

Cubietrucks simulieren eine Testumgebung für Big Data mit verschiedenen Systemen.

Über den HITeC e. V.

Der Hamburger Informatik Technologie-Center e.V. (HITeC) ist ein eingetragener, gemeinnütziger Verein, der von Mitgliedern des Fachbereichs Informatik der Universität Hamburg getragen wird. Der Verein ist über einen Kooperationsvertrag mit der Universität Hamburg verbunden. HITeC sieht seine Hauptaufgaben in der:

- Durchführung anwendungsorientierter Forschungsvorhaben
- Verbreitung anwendungsorientierter Forschungsergebnisse
- Durchführung von Weiterbildungsveranstaltungen
- Vermittlung von Kontakten zwischen Firmen und Studierenden
- Verbesserung der praxisorientierten Ausbildung
- Unterstützung bei Unternehmensgründungen

die besonderen Methoden der Big Data-Datenhaltung wissen.

Für das Data Mining in den abgelegten HDFS-Daten per Hadoop stehen vorbereite Auswertungen zur Verfügung, die z. B. die komplexen XML-Formate auswerten können. Des Weiteren liefern sie bei Bedarf komplette Beschreibungen des „Lebensweges" eines Dokumentes, basierend auf den Informationen aus Sende- und Empfangsprotokollen, Logdateien und Quittungen. Die zeitaufwändige Suche und Rekonstruktion von einzelnen Dokumenten entfällt damit.

Natürlich können beliebige andere Dokumente, Quittungen oder Logdateien außerhalb des eANV auf dem gleichen Wege abgelegt und durchsucht werden. Daten z. B. aus der Straßenreinigung verknüpft mit Witterungsinformationen, Salzverbräuchen, Beschwerdedaten u.v.m. können ein zuverlässiges Planungsinstrument liefern. Für jeden Anwendungsfall in Abfallwirtschaft und Umweltschutz gilt, dass das eANV-Waste-API von Consist ITU der vorhandenen Software die notwendigen Big-Data-Funktionen einfach zur Verfügung stellt.

Begriffserklärungen:

- MongoDB: (abgeleitet vom engl. humongous, „gigantisch"): hochperformante, schema-freie, dokumentenorientierte Open-Source-Datenbank
- GridFS: Protokoll, um Dateien aus der MongoDB abzurufen
- HDFS: hochverfügbares, leistungsfähiges Dateisystem zur Speicherung sehr großer Datenmengen auf den Dateisystemen mehrerer Rechner (Knoten)
- Apache Hadoop: Framework für skalierbare, verteilt arbeitende Software
- NoSQL (Not only SQL): nichtrelationale Datenbanken
- ARM-Soc: Ein-Chip-System von ARM
- ETL (-Prozess): Extract, Transform, Load
- VM: Virtuelle Maschine
- MapReduce: Programmiermodell (und Name der Implementierung) für nebenläufige Berechnungen über sehr große Datenmengen auf Computerclustern

Weitere Informationen:

Ute Müller
Telefon: 040/30625-116
E-Mail: ute.mueller@consist-itu.de

Karsten Evers
Telefon: 0431/3993-590
E-Mail: evers@consist.de