Big Data vs. Fast Data: Shootout

Marcel Amende, ORACLE Deutschland B.V. & Co. KG, Düsseldorf

Schlüsselworte
Big Data, Fast Data, Complex Event Processing, CEP, OEP

Einleitung

Is BIG *fast* enough? Diese Frage stellt sich, seit Gartner die drei großen ***V***'s als Kennzeichen von Big Data Anwendungen postulierte: ***V****olume* (Datenmenge), ***V****ariety* (Vielfalt) und ***V****elocity* (Geschwindigkeit). Im Vertrieb hängt man gerne ein viertes ***V*** an: ***V****alue* (Wert), um das Werteversprechen für das anwendende Unternehmen hervorzuheben. Stellen wir die drei technischen Charakterisierungen auf den Prüfstand, indem wir sie mit einer Definition und der Arbeitsweise von Big Data Lösungen abgleichen.

Die Entmystifizierung von Big Data

Oft findet man im realen Leben Analogien zu IT-Lösungen und Architekturansätzen: Big Data erscheint mir vergleichbar mit dem Messie-Syndrom oder einer pathologischen Sammelleidenschaft: Man kann nichts wegschmeißen, auch wenn es im ersten Moment als wertlos erscheint. Man glaubt, es irgendwann vielleicht noch einmal gebrauchen zu können. Nur wenige werden sich herrschaftliche Villen *(=klassische Datenbanksysteme)* für ihre Sammlungen leisten können. Wahrscheinlicher mietet man sich einige günstige Garagen *(=günstige Festplatten)* für, um die Sammlung in Pappschachteln *(=Datenblöcke)* zu deponieren. Hat man Angst, Teile der Sammlung durch ein Feuer zu verlieren, verteilt man Duplikate von Dingen in verschiedenen Garagen *(=redundante Datenhaltung)*.

Will man Dinge wiederfinden, nummeriert man die Kartons und führt Listen mit ihren Inhalten *(=Blockliste)*. Sucht man nun etwas Bestimmtes, kann man es einfach in der Liste nachschlagen und aus der nächstliegenden der Garagen holen lassen.

Manchmal ist es nötig, die Sammlung nach neuen Kriterien zu sortieren *(=Map)* und auszuwerten *(=Reduce, "R")*. Für eine Ausstellung sucht man etwa nach allen Stühlen einer Bestimmten Epoche, nach Büchern zu einem bestimmten Thema oder allen Haushaltsgeräten aus den Siebzigern. Hierfür werden in jeder Garage nützliche Helfer benötigt, die in alle Kartons *(=Datenblöcke)* schauen und die gefundenen Artefakte *(=Values)* nach wechselnden Kriterien *(=Keys)* zu neuen Kollektionen zusammenstellen.

Im Privatfernsehen helfen dabei Menschen, wie Tine Wittler. Wir brauche aber eine passende IT-Lösung: In der sich ständig enger vernetzenden Welt stoßen Applikationen, die auf klassischen, transaktionalen Datenverarbeitungsalgorithmen beruhen, an ihre Grenzen. Ungeheure Mengen *(**V**olume)* von Daten in mannigfaltigen Formaten *(**V**ariety)* müssen gespeichert, verarbeitet und analysiert werden. Diese Mechanismen müssen günstig, aber dennoch zuverlässig sein, immerhin wollen wir Informationen ablegen, deren Wert im Einzelnen für unser Geschäft noch im Unklaren liegt, deren Extrakt aber in vielerlei Hinsicht wertvoll sein kann. Eine klassische Datenbank wäre hierfür eine komfortable, aber ziemlich teure Lösung. Aus welchen Komponenten ergibt sich nun ein tragfähiger und zugleich günstiger IT-Lösungsansatz?

Kernkomponenten einer "Big Data" Lösung

Wir benötigen drei Dinge:

1. Eine sehr günstige Möglichkeit der dauerhaften Speicherung von Massendaten. Die günstigsten verfügbaren Medien sind hier klassische *(SAS-)* Festplatten in TB-Größe, die in großen Verbünden

von hunderten bis tausenden von Platten Speicherplatz in PB-Größe liefern. Vor Datenverlust schützt man sich durch ein verteiltes Dateisystem *(z.B. HDFS)*, das Datenblöcke auf mehrere physikalische Platten und Server ausfallsicher repliziert. Typischerweise werden Daten unabhängig von ihrer Struktur einmalig und fortlaufend in große Datenblöcke geschrieben *(„write once")*. Aktualisierung und Löschen einzelner Datensätze ist konzeptionell nicht vorgesehen.

2. Eine Parallelverarbeitung, die unstrukturierte Daten in den Blöcken nach immer neuen Kriterien durchforsten und sortieren kann *(„read many")*. Dies sind physikalische und logische Threads, die mit direktem Zugriff auf das Dateisystem möglichst die Daten der lokalen Platten verarbeiten, nach Schlüsselinformationen sortieren *(„Map")* und verdichten *(„Reduce")*.

3. Mächtige Werkzeuge zur Analyse und Weiterverarbeitung der gefundenen Informationen, die entweder in den Parallelverarbeitungsprozess eingebunden werden oder auf den Ergebnismengen aufsetzen. Dies können Statistikpakete *(„R")*, klassische BI-Lösungen, Not-only-SQL- und relationale Datenbanksysteme sein.

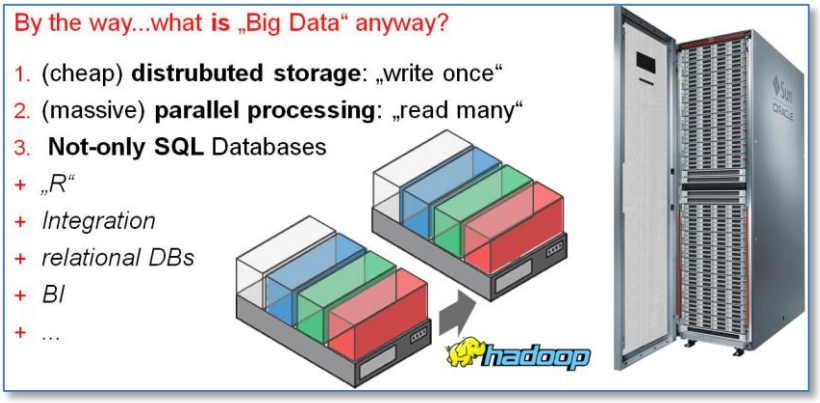

Abb. 1: Was ist Big Data?

Eine Big Data Verarbeitung erfolgt demnach asynchron, in einer sequentiellen Kette von verschiedenen, teils zeitintensiven Aktivitäten, die am Kriterium ***V****elocity* (Geschwindigkeit) zweifeln lassen. Big Data ist eher mit einem LKW-Transport vergleichbar: Dieser muss vor der Abfahrt erst voll beladen werden, bis er in die Zentrale fährt, wo der Inhalt bewertet und Aktionen eingeleitet werden können.

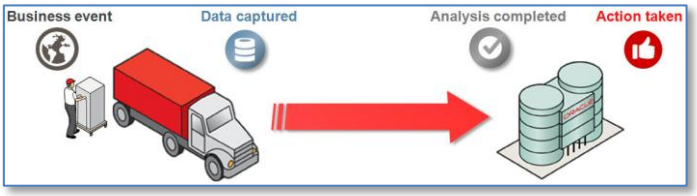

Abb. 2: Vergleich von Big Data mit einem LKW-Transport

Der Wert von Information über die Zeit

Typischerweise verliert eine Information in einem Unternehmen über die Zeit an Wert. Die Kenntnis über eine Fehlverladung verliert z.B. nach wenigen Minuten an Wert, sobald die Lieferung den Verladeort verlässt. Bei schneller Benachrichtigung kann dem Fehler noch effektiv begegnet werden, indem die Fehlverladung vor Ort korrigiert wird. Ein zu spätes Erkennen resultiert hingegen in kosten- und aufwandsintensiven Korrekturmaßnahmen, hier Retoure und Neuauslieferung.

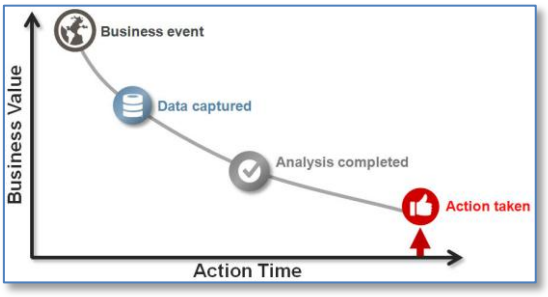

Abb. 3: Information verliert mit der Zeit an Wert

Fast Data: Agieren ich Echtzeit

Ist die schnelle Reaktionsfähigkeit bzw. Verarbeitungsgeschwindigkeit *(**Velocity**)* ein wichtiges Kriterium, bietet sich die Einbindung einer Fast-Data Strategie an. Konkret kann dies in Form einer „Complex Event Processing"-Engine (z.B. Oracle Event Processing, OEP) erfolgen:

Beim „Complex Event Processing" handelt es sich um ein Konzept zur Analyse und Verarbeitung hochvolumiger Datenströme. Ereignisse werden miteinander korreliert, um in Echtzeit oder mit vorhersagbarer Antwortzeit Aktionen auslösen zu können.

Die Besonderheit dabei ist, dass die zeitliche Abfolge von Ereignissen und Algorithmen zur Mustererkennung einfach in strukturierte Abfragen eingebunden werden können. Dafür wird die „Continuous Query Language (CQL)", eine an der Stanford Universität entwickelte Erweiterung des klassischen SQL, in Kombination mit regulären Ausdrücken in einem sogenannten „Ereignisverarbeitenden Netzwerk" genutzt.

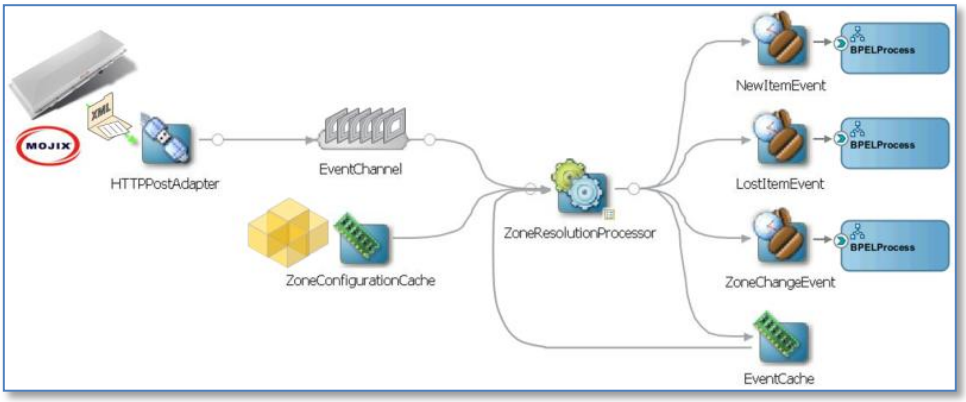

Abb. 4: Beispiel für ein ereignisverarbeitendes Netzwerk

Folgende Abfrage liefert als Ergebnismenge in Echtzeit die IDs aller Dinge, die nach einer Erwärmung auf über 30° (Ereignis „A") auf unter 0° (Ereignis „C") abkühlen:

```
SELECT its.itemId
FROM ItemTempStream MATCH_RECOGNIZE (
                    PARTITION BY itemId
                    MEASURES A.itemId as itemId
                    PATTERN (A B* C)
                    DEFINE
```

A AS (A.temp > 30),
B AS (B.temp < 30),
C AS (C.temp < 0)

…

Fazit

"Big Data" Lösungen haben offensichtlich ihre Stärke in der Ablage und -Verarbeitung von unstrukturierten *(**Variety**)* Massendaten *(**Volume**)*. Für Echtzeitanwendungen oder kurze, vorhersagbare Antwortzeiten *(**Velocity**)* eignet sich diese Technik nicht. Hier bieten sich „Fast Data" Konzepte als ideale Ergänzung an, z.B. in der Vorverarbeitung und –Filterung von hochvolumigen *(**Volume**)* und kontinuierlichen Datenströmen, die in Kombination für ein Unternehmen und das Geschäft noch einen deutlich größeren Wert haben können *(**Value**)*.

	Big Data	Fast Data
Volume	✓	✓
Variety	✓	
Velocity		✓
Value	✓	

Abb. 5: Shootout "Big" vs. "Fast" Data

Kontaktadresse:

Marcel Amende
Senior Leitender Systemberater
ORACLE Deutschland B.V. & Co. KG
Hamborner Str. 51
D-40472 Düsseldorf
Telefon: +49 (0) 211-74839 539
E-Mail: Marcel.Amende@oracle.com
Internet: www.oracle.de