Building Scalable Big Data Pipelines

Christian Gügi, Solution Architect

19.09.2013
AGENDA

- Opportunities & Challenges
- Integrating Hadoop
- Lambda Architecture
- Lambda in Practice
- Recommendations
ABOUT ME

- Solution Architect @ YMC
- Founder and organizer Swiss Big Data User Group
  - [http://www.bigdata-usergroup.ch/](http://www.bigdata-usergroup.ch/)
- Contact
  - christian.guegi@ymc.ch
  - [http://about.me/cguegi](http://about.me/cguegi)
  - @chrisgugi
ABOUT YMC

- Founded in 2001
- Based in Kreuzlingen, Switzerland
- Big Data Analytics, Web Solutions and Mobile Applications
- 24 experts
  - Consulting, creation, engineering
BIG DATA – WHAT IS THE BIG DEAL?

A. New sources and types from inside & outside organisations
   - “Internet of things”, sensors, RFID, intelligent devices, etc.
   - Unstructured information – documents, web logs, email, social media, etc.
   - Trusted 3rd party sources – industry provider & aggregators, governments “Open Data”, weather, etc.

B. Technology innovations to exploit new world of data
   - Low cost storage and process power (cloud, on-premise & hybrid)
   - New software patterns to handle speed & volume, structured and unstructured (In-memory computation, Hadoop, Mapreduce, etc.)
   - Revolution in user experience, analytics, recommendations
BIG DATA – CHALLENGES

- Volume
- Velocity
- Variety
- Veracity

Character of data

Organisational issues

Overwhelming landscape & integration

Available talent

- Align business strategy
- Data Management
- Privacy protection

- Lack of skilled and experienced people
INTEGRATING

HADOOP
TYPICAL RDBMS Szenario

Apps
- BI
- Web
- Mobile

Data Systems
- DWH
- ETL
- RDBMS

Data Sources
- RDBMS
- NFS
- Others
BIG DATA SZENARIO

Data Sources: RDBMS, NFS, Logs, Social Media, Sensors

Data Systems: DWH, RDBMS, Hadoop

Apps: BI, Web, Mobile

1) Recommendations, etc.
LAMBDA

ARCHITECTURE
LAMBDA ARCHITECTURE

- Credits Nathan Marz
- Former Engineer at Twitter
- Storm, Cascalog, ElephantDB

http://www.manning.com/marz/
DESIGN PRINCIPLES

Lambda Architecture

- Human fault-tolerance
- Data immutability
- Re-computation
HUMAN FAULT-TOLERANCE

Lambda Architecture

- Design for human error
  - Bugs in code
  - Accidental data loss
  - Data corruption
- Protect good data, so you can always fix what went wrong
DATA IMMUTABILITY
Lambda Architecture

- Store data in its rawest form
- Create and read but no *update*
- No data can be lost
  - To fix the system just delete bad data
  - Can always revert to a true state
## DATA IMMUTABILITY

**Lambda Architecture**

### Capturing change traditionally (mutability)

<table>
<thead>
<tr>
<th>Name</th>
<th>Location</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alice</td>
<td>Zurich</td>
<td>2009/03/29</td>
</tr>
<tr>
<td>Bob</td>
<td>Lucerne</td>
<td>2012/04/12</td>
</tr>
<tr>
<td>Tom</td>
<td>Bern</td>
<td>2010/04/09</td>
</tr>
</tbody>
</table>

### Capturing change (immutability)

<table>
<thead>
<tr>
<th>Name</th>
<th>Location</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alice</td>
<td>Zurich</td>
<td>2009/03/29</td>
</tr>
<tr>
<td>Bob</td>
<td>Lucerne</td>
<td>2012/04/12</td>
</tr>
<tr>
<td>Tom</td>
<td>Bern</td>
<td>2010/04/09</td>
</tr>
<tr>
<td>Alice</td>
<td>Basel</td>
<td>2013/08/20</td>
</tr>
</tbody>
</table>
Always able to re-compute from historical data

Basis for all data systems
  query = function(all data)
Lambda in Practice
ONLINE MARKETING

- Tracking and analytics solution
- Improve customer targeting and segmentation
- Various reports
- Real-time not required
DATA PIPELINE

Extracting

Transformation

Loading

AdServer → Flume → HDFS
Campaign Database → Sqoop → HDFS
FTP → fs -put → HDFS

HDFS → M/R → Avro

M/R → Avro

HDFS → Tracking

HDFS → Bulk Importer

HDFS → Profiles

HDFS → DWH
ADVANTAGES

- Extensible – easily add speed layer later on
- Complements existing DWH/BI system
- ETL phases are decoupled
- Reliable
  - Infrastructure
  - Each step can be replayed
- Scalable
  - Storage
  - Processing
- Highly available
- Ad-hoc analysis right from the beginning
RECOMMENDATIONS
RECOMMENDATIONS

- Not a fixed, one-size-fits-all approach
  - Adopt to your needs/requirements
- Hadoop complements existing systems
- How real-time do I need to be?
- Immutability and pre-computation are just good ideas!
  - Store information in rawest format possible
  - Use a serialization framework (Avro, Thrift, Protocol Buffers)
THANK YOU!
CONTACT

cristian.guegi@ymc.ch
Tel. +41 (0)71 508 24 76
www.ymc.ch
@chrisgugi