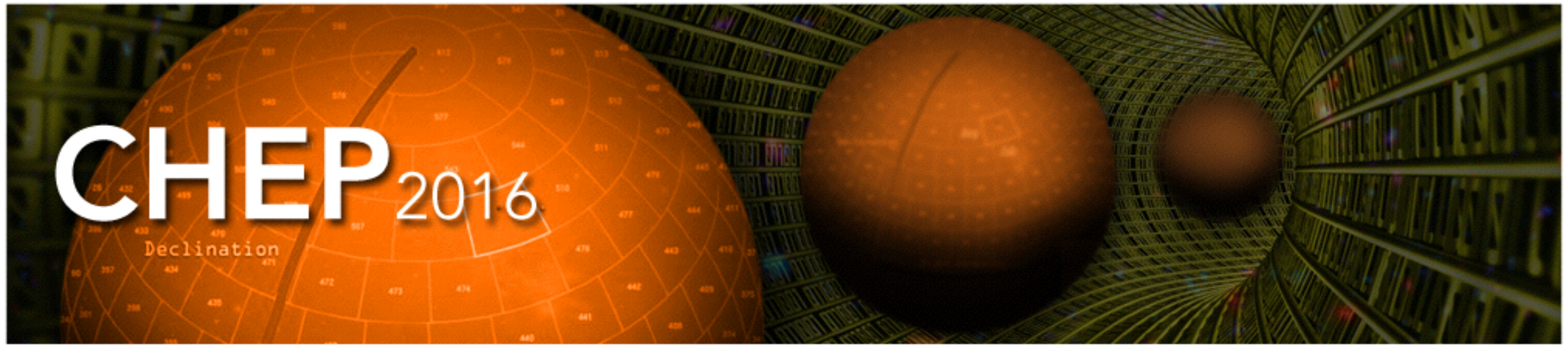


22nd International Conference on Computing in High Energy and Nuclear Physics, Hosted by SLAC and LBNL, Fall 2016

Hadoop and friends

first experience at CERN with a new platform for high throughput analysis steps

D. Duellmann, L. Menichetti, K. Surdy, R. Toebbicke - CERN, Switzerland
D. Gupta, A. Kumar, V. Menon - BARC, India

presented by Prasanth Kothuri

Infrastructure Analytics at CERN

- Goal: quantitatively understand the computing involving the CERN Computer Centre
    - eg file transfers on the grid, eg accuracy of CPU benchmark scores
- Scope
    - medium to long term metrics analysis: e.g. weeks to years
    - using statistics and/or machine learning tools
    - beyond time series / across single computing sub-systems
- Analytics Working Group (AWG)
    - experts from most IT groups and experiments
    - collaboration with BARC, Mumbai

Analysis Data Sources

- Using (pre-existing) metrics from all levels of the application stack
  - Lemon and Agile Infrastructure - box and VM level metrics
  - storage & batch - job cpu and file usage
  - experiment workflow info, where available
    (job type, role in a larger task, … - eg from Panda, PhEDEx)
- Rely on **IT monitoring project** for metrics transport to a **single HDFS repository**
  - HDFS & Hadoop act as long term storage with local processing in
    CERN's "λ-infrastructure"

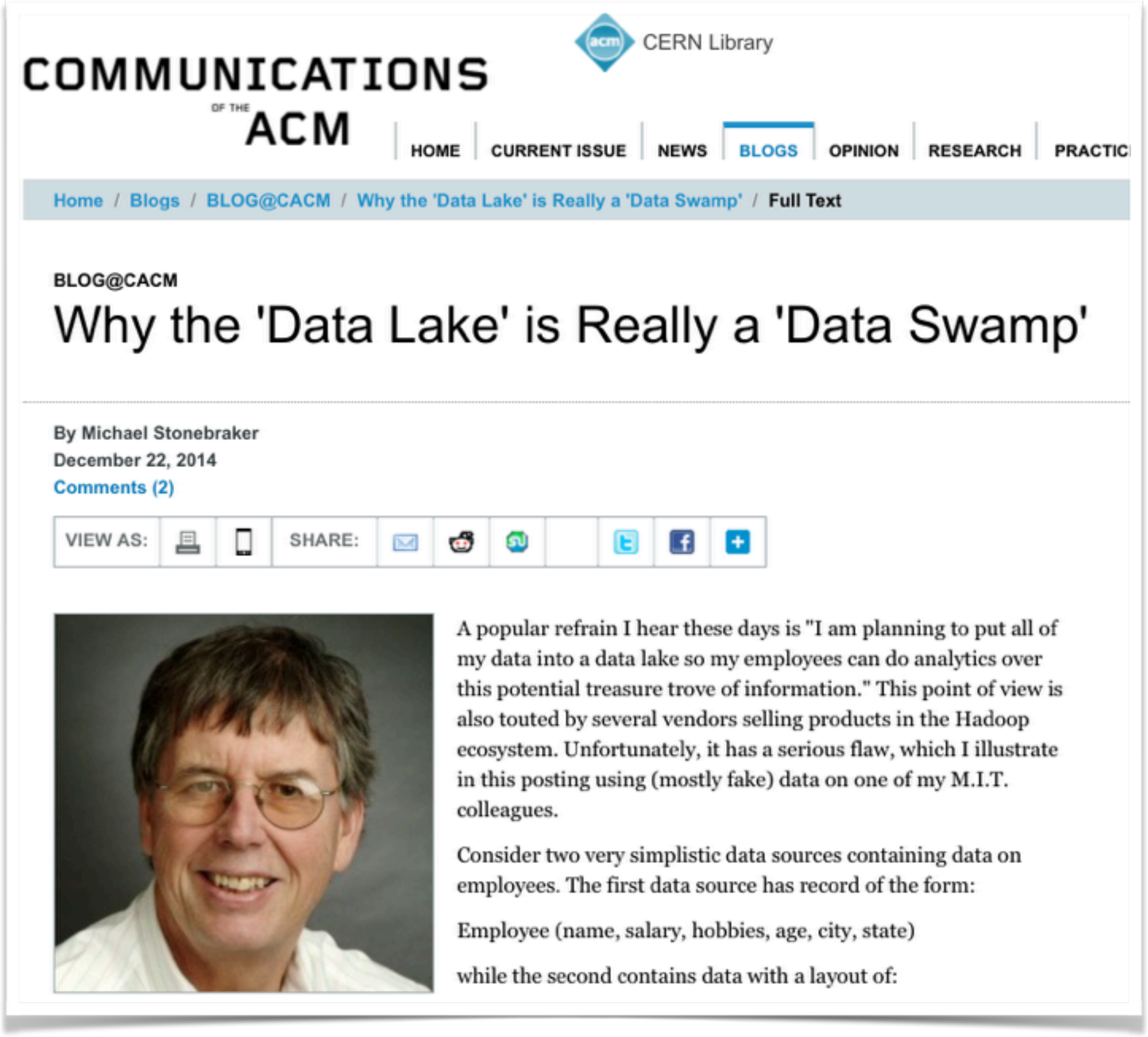
Raw Input Data

Subsystem	Location	Amount	
lemon	hdfs	78 TB	box level
squid	hdfs	110 GB	http cache access
openstack	hdfs	12 TB	agile infrastructure
syslog	hdfs	23 TB	unstructured box logs
eos	hdfs	12 TB	file access metrics
castor	hdfs	55 TB	tape archive access
LANdb	hdfs	small O(100 MB)	host,ip,hypervisor, location
perfsonar	hdfs	small O(10 GB)	network link status
exp. dashboard	hdfs	small (< 1TB)	job summaries
exp. file popularity	hdfs	small O(200GB)	user data access
batch	hdfs	500 GB	accounting & queue-config
hw specs	afs	100MB)	h/w rating per model

(6/2016)

Data Volume & Structure

- In contrast to physics data analysis:
  - often unstructured
  - no up-front, designed data model
- Medium volume
  - usually several tens of TB per analysis dataset
  - not “Big Data”, but processing times can be large enough (hours to days) to disrupt interactive analysis
- Prepare data extracts for analysts
  - keep people focussed on understanding the data
  - … not just on waiting for batch jobs to select data


COMMUNICATIONS OF THE ACM
CERN Library
HOME
CURRENT ISSUE
NEWS
BLOGS
OPINION
RESEARCH
PRACTIC
Home / Blogs / BLOG@CACM / Why the 'Data Lake' is Really a 'Data Swamp' / Full Text
VIEW AS:
SHARE:

BLOG@CACM

Why the 'Data Lake' is Really a 'Data Swamp'

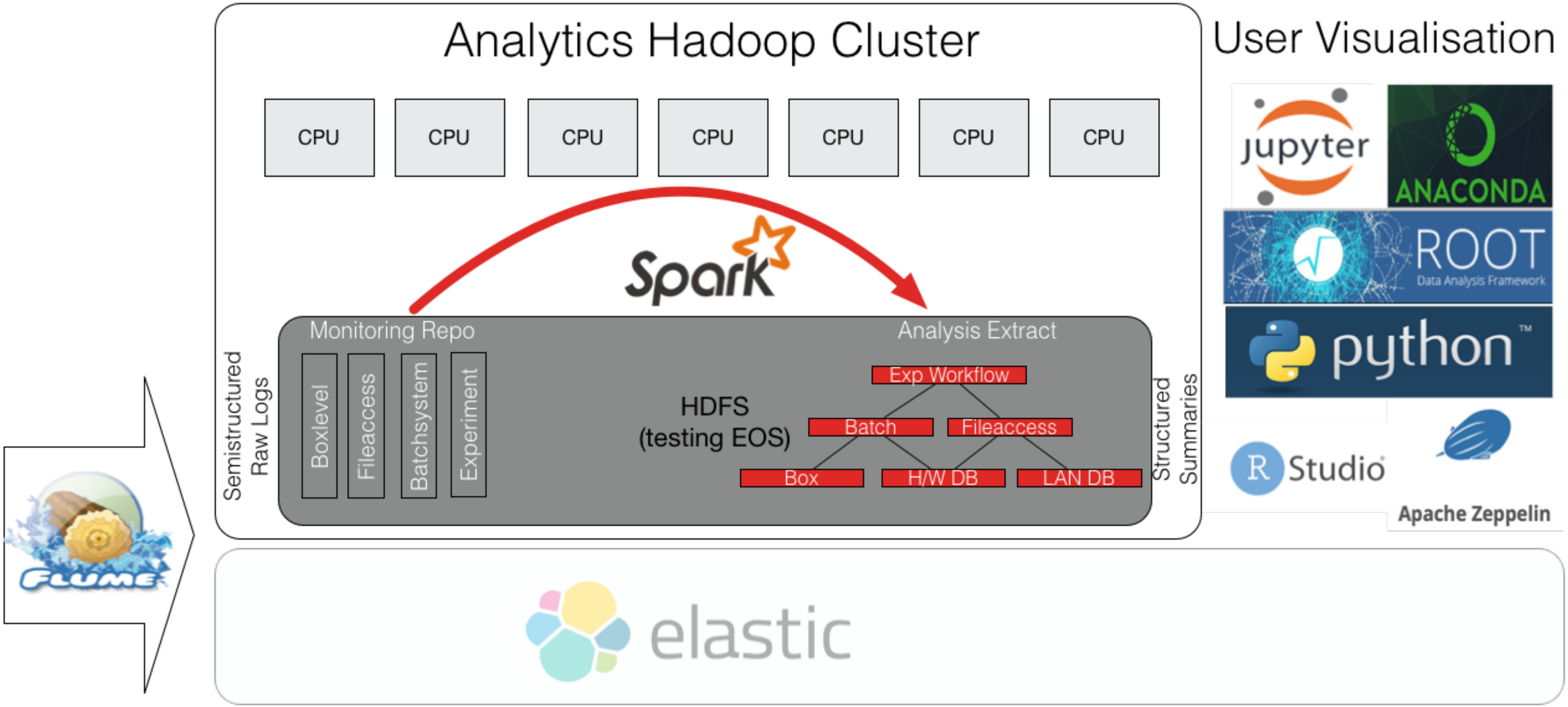
By Michael Stonebraker
December 22, 2014
Comments (2)

A popular refrain I hear these days is "I am planning to put all of my data into a data lake so my employees can do analytics over this potential treasure trove of information." This point of view is also touted by several vendors selling products in the Hadoop ecosystem. Unfortunately, it has a serious flaw, which I illustrate in this posting using (mostly fake) data on one of my M.I.T. colleagues.

Consider two very simplistic data sources containing data on employees. The first data source has record of the form:

Employee (name, salary, hobbies, age, city, state)

while the second contains data with a layout of:



Analytics Hadoop Cluster

CPU CPU CPU CPU CPU CPU CPU

Semistructured Raw Logs

Monitoring Repo

Boxlevel

Fileaccess

Batchsystem

Experiment

HDFS (testing EOS)

Analysis Extract

Exp Workflow

Batch

Fileaccess

Box

H/W DB

LAN DB

Structured Summaries

User Visualisation

One example - Lemon Sensors

- Some 100 TB - around 1000 different metrics
  - a large fraction of the metrics is designed to aid operations and problem tracking. Hence less relevant for quantitative analysis…
  - some metrics have data quality issues
    (eg precision and accuracy issues, wrong units, missing or corrupted measurements, delayed arrival)
  - already useful for simple time series visualisation
    (but not for quantitative analysis)

Example: DB data - PhEDEx transfers

- CMS data transfer service and replica catalog
  - ~1TB is currently stored
- daily imports from Oracle with Sqoop
  - incremental updates of the catalog
  - blocks replica table snapshots
- Benefits of processing in Hadoop
  - thanks to Hadoop capacity, we can store daily snapshots and hence keep the transfer history
  - Oracle is offloaded from computing statistics and aggregations
  - out-of-the-box parallelised queries

Preparing for Analysis

- Frequency of metric collection (eg every few minutes)
  - adapted for operations monitoring/dashboards
  - but, unnecessarily high for most med/long term analysis
- Input data format is JSON
  - {convenient for data integration and transport}
  - but requires high CPU & IO overhead during analysis

Apache Spark

- Selected Spark for preselection and aggregation
  - open source, large and active use base
  - library support for
    - (a) unstructured input data
    - (b) efficient analysis storage formats
    - (c) stats and machine learning algorithms
  - provides parallel processing primitives - either:
    - declarative - traditional SQL queries
    - imperative (no-SQL) ~= additional control over query execution
  - bindings to most popular analysis languages: Python, R, Scala, Java

Spark Aggregation

- With colleagues from BARC, Mumbai we developed a Spark data aggregation package that
  - turns raw metrics data into an efficient analysis repository
  - generates statistical relevant summaries from individual measurements (eg hourly stats, standard deviation, min/max)
  - scalable re-aggregation by cluster, service (puppet “host groups”) or group of VMs (AI “projects”)
  - executed as periodic jobs: data is ready for further human analysis

Network Connection Study

- Goal: optimise the utilisation of the various IPV4 networks at CERN
    - determine the maximum concurrent connection count on all subnets
    - Input: about half a year of subnet connection records
    - Implementations studied
        - SQL and non-SQL implementations to determine peak and average network usage
        - optimised query from initial runtime of 50h (on oracle) to few minutes (spark cluster)

Summary

- CERN has in collaboration with BARC setup an analysis repository with combined metrics from different subsystems

- Apache Spark is used to implement a periodic processing pipeline that extracts relevant metrics and derives statistical aggregates from raw metrics

- This analysis input is provided in a variety of formats, including the CPU and storage efficient “parquet” and “avro”

- Complements elasticsearch with support for more flexible analytics/visualisation and long data retention

Pointers and Links

- Analysis working group resources
  - A Twiki with working group info can be found [here]
  - Ongoing work items are tracker in JIRA [here]
  - Meetings and presentations are logged [here]

Related CHEP 2016 contributions

- (369) Exploiting analytics techniques in CMS computing monitoring
- (229) First results from a combined analysis of CERN computing infrastructure metrics