# ADC Analytics

## Database Futures Workshop

Mario.Lassnig@cern.ch

on behalf of ATLAS Distributed Computing

# ADC Analytics

- Understand our distributed systems
  - Usage characterisation — *What do our users do? What do our systems do?*
  - Performance characterisation — *How long does it take to retrieve? How fast can we insert? ...*

- Key capabilities
  - Correlate — Data from multiple systems
  - Model — Using raw and aggregated data with data mining and machine learning toolkits
  - Host — Third party analytics software
  - Curation — Analytics for experiment needs
  - Ad-hoc — Analytics for user-requested questions
  - Support — Documentation and expert help

# ADC Analytics

- Understand our distributed systems
  - Usage characterisation — *What do our users do? What do our systems do?*
  - Performance characterisation — *How long does it take to retrieve? How fast can we insert? ...*

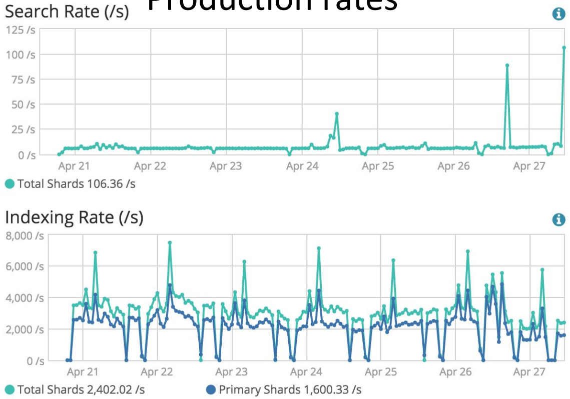
# Baseline Infrastructure

- ElasticSearch
  - Dedicated v5 instance for ATLAS hosted by CERN IT
  - Shared v2 instance for monitoring hosted by CERN IT
  - Shared v5 instance hosted by University of Chicago
- Notebooks
  - Dedicated Zeppelin instance hosted by CERN IT and administered by ADC
- Hadoop
  - HDFS to store raw and aggregated data
  - Preparation of data for ElasticSearch ingestion
  - Machine learning with Spark

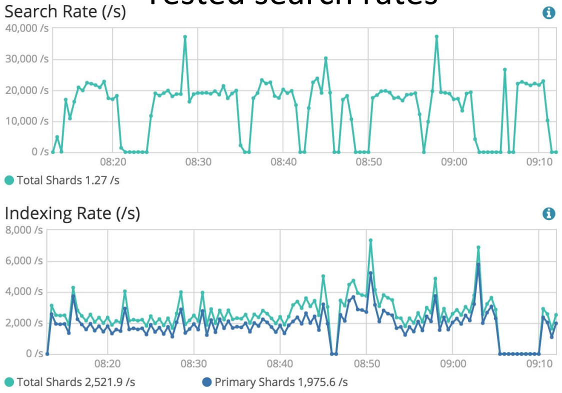
# ElasticSearch @ UChicago

- Hardware
  - 8 nodes total (each 8 core, 64GB RAM)
  - 5 data nodes (3x1 TB SSD each)
  - 3 master nodes (3 masters, 1 indexer, 2 kibana)
  - 10 Gbps NICs
- Contents
  - 15'000'000'000 documents
  - 16'000 shards
- Clients
  - Kibana
  - Notebooks
  - Embedded visualisations
  - Crons

Production rates




Tested search rates

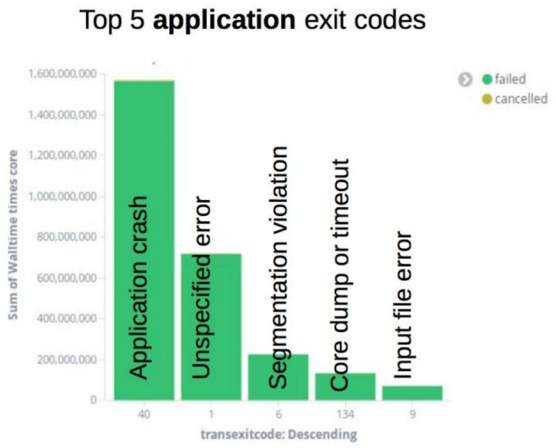


# ElasticSearch @ CERN

- ITES Cluster
  - *cf. Pablo's & Ulrich's talk*

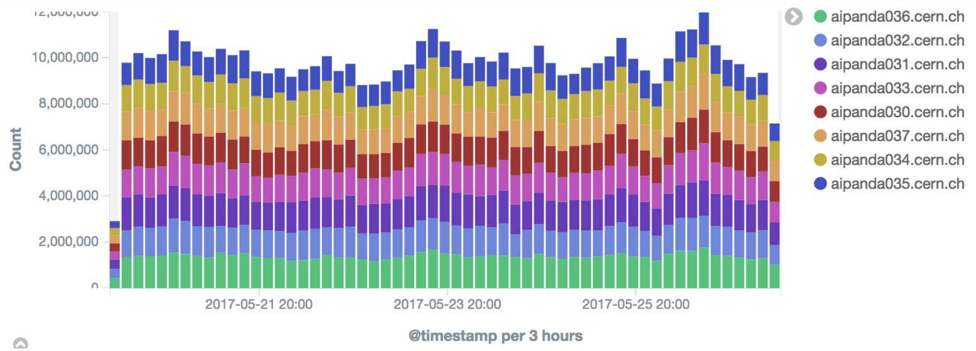
# ElasticSearch example

- PanDA & JEDI (Workflow Management)
  - ATLAS runs ~2 million jobs per day
  - ES index with completed (failed & succeeded) jobs for analytics
    - Memory usage
    - User workflow evaluation
    - User resource consumption
    - Queueing time of jobs
    - Reasons for job failures
    - …
  - The average user prefers ES because it's easier to use, but the data is also in Hadoop
  - Machine Learning on Hadoop: Task Time to Complete (TTC) and anomaly detection
    - Using regression trees with Spark MLlib



# ElasticSearch example

- PanDA & JEDI (Workflow Management)
  - Logstash infrastructure to ship PanDA core and monitoring logs to es-atlas
  - Insight into WM decisions, used for daily operations and debugging
    - Task and job brokerage: e.g. why did this task broker here or why is this site not getting jobs
    - Load on dispatch servers
  - Implementation of alarms and potentially service monitoring
  - Extensive wealth of data, still more potential to utilize



# Overall ElasticSearch rates

- 16 indices in production across ITES (8) and UChicago (8)
  - 230 GB/day (=82TB/year) if we keep everything we want to keep at our current rates
  - Biggest indices
    - DDM traces (ITES), events (ITES), and logs (UChicago)
    - WMFS jobs (UChicago) and logs (ITES)
  - These few indices exist on Hadoop as well (mostly Flume double sinks), but most others do not
- Few options available
  - We keep rolling buffers
    - Cannot look further back than 30 days — not enough for many of our reports
  - We selectively throw away data
    - Painful to know what could be important upfront, esp. for operations
  - We reimplement all the fancy dashboards from Kibana/Grafana on Spark with custom notebooks
    - Not enough human capacities for this

# Hadoop @ CERN

- We exclusively use the "analytix" cluster
  - 40 nodes, 2TB RAM total, 2PB storage total

- Rucio (Data Management)
  - Dumps from Oracle
    - Custom dumps with sqlplus (tab-separated)
    - Table dumps with sqoop (avro)
  - Flume from DDM servers and daemons (REST calls, logs, …)
  - Custom servlet to serve HTTP streams directly from HDFS
  - Reporting with Pig — results shown via notebooks
  - Kerberos authentication via cumbersome acron method

**DDM space usage since Run-1**

| | |
|---|---|
| Daily Oracle Dumps | 160TB |
| Logs | 120TB |
| Archive | 30TB |
| Traces | |
| 30TB | |
| Machine Learning | 60GB |
| Reports | 50GB |

# Notebooks / Reporting / Ad-hoc

- Custom notebook server installations, connected to ElasticSearch & Hadoop
  - Zeppelin @ CERN — elasticsearch-py, pyHDFS, Keras/Tensorflow, dist-keras, …
  - Jupyter @ UChicago — same, but powerful hardware (32cores, 128GB RAM, 2x Tesla K20c)

- Correlation studies between ADC systems (anomaly detection, operations support)
  - Do (and if yes, how) failures propagate between systems, and propose solutions to shifters
- Scrutiny group reports
  - Spark jobs that analyze year's worth of historical data accesses
- Dynamic computation of network metrics
  - Merges, cleans, and computes metrics from many sources: ElasticSearch, Oracle, HTTP servers, …
  - Pushes into Redis and back into ElasticSearch — used for job brokering
- Time-To-Complete modelling of data transfers for replica selection/placement
  - Recurrent neural networks using Keras

# Summary and conclusions

- ElasticSearch and Hadoop are both critical for us and used in production
  - Most of the tools/capabilities that we need are available
  - Interplay between tools not always ideal and efficient, lots of custom-made duct-tape
  - And new requirements always come, we have to be flexible
- Hadoop is working well, ES hardware is sufficient for now (=< late 2017)
  - We are limiting ourselves hard from the application side (slidings windows, throwing away data, …)
  - Serious upgrades will be necessary wrt. storage space and IO rates at least on ITES cluster
  - Run-3/4 considerations
    - 10 time increase of rates and volume, in line with WFMS and DDM upgrades
    - Event-level processing workflows will require unprecedented instrumentation
- Users are much more comfortable now than half a year ago
  - However, inexperienced users can hit the infrastructure hard
  - Documentation is still a problem — systems are changing fast, still easier to ask colleagues