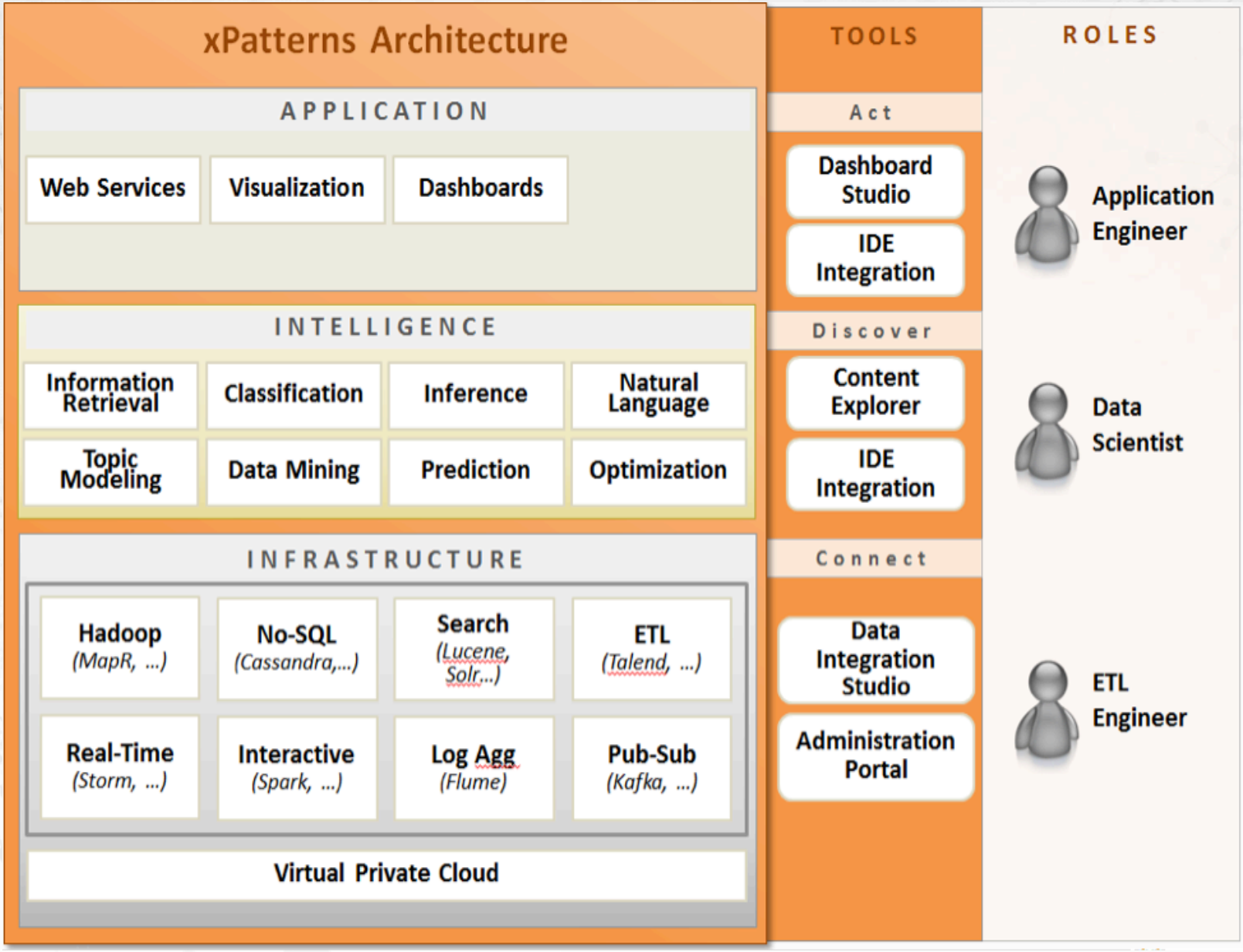
xPatterns on Spark, Shark, Tachyon and Mesos

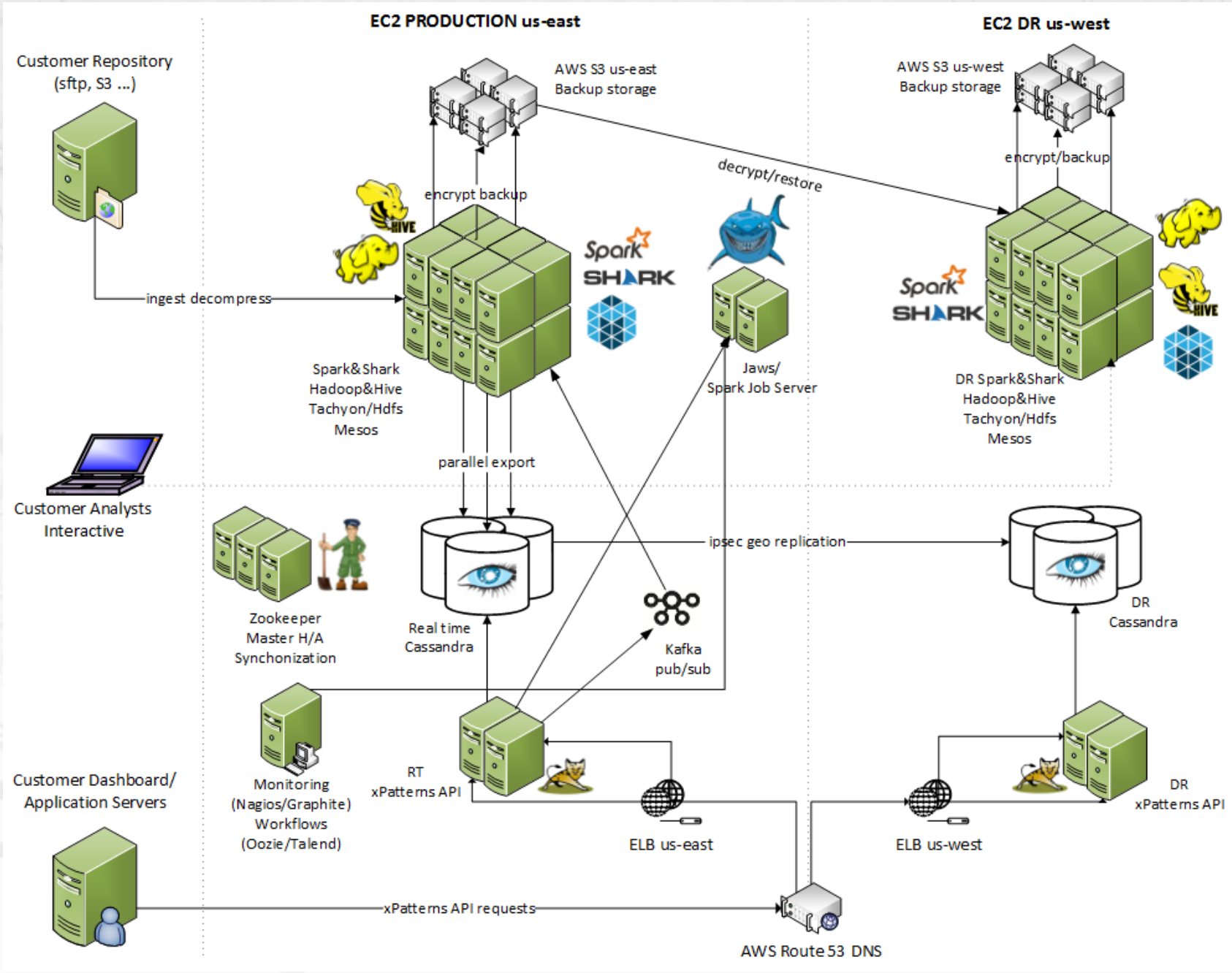
Spark Summit 2014

Claudiu Barbura
Sr. Director of Engineering
Atigeo

Agenda

- xPatterns Architecture
- From Hadoop to BDAS & our contributions
- Lessons learned with Spark: from 0.8.0 to 0.9.1
- Demo: xPatterns APIs and GUIs
  - Ingestion (EL)
  - Transformation (T)
  - Jaws Http SharkServer (warehouse explorer)
  - Export to NoSql API (data publishing)
  - xPatterns monitoring and instrumentation (Demo)
- Q & A


xPatterns Architecture
APPLICATION
Web Services
Visualization
Dashboards
INTELLIGENCE
Information Retrieval
Classification
Inference
Natural Language
Topic Modeling
Data Mining
Prediction
Optimization
INFRASTRUCTURE
Hadoop (MapR, ...)
No-SQL (Cassandra,...)
Search (Lucene, Solr...)
ETL (Talend, ...)
Real-Time (Storm, ...)
Interactive (Spark, ...)
Log Agg (Flume)
Pub-Sub (Kafka, ...)
Virtual Private Cloud
TOOLS
Act
Dashboard Studio
IDE Integration
Discover
Content Explorer
IDE Integration
Connect
Data Integration Studio
Administration Portal
ROLES
Application Engineer
Data Scientist
ETL Engineer


EC2 PRODUCTION us-east
EC2 DR us-west
Customer Repository
(sftp, S3 ...)
AWS S3 us-east
Backup storage
AWS S3 us-west
Backup storage
encrypt backup
decrypt/restore
encrypt/backup
ingest decompress
Spark&Shark
Hadoop&Hive
Tachyon/Hdfs
Mesos
Jaws/
Spark Job Server
DR Spark&Shark
Hadoop&Hive
Tachyon/Hdfs
Mesos
parallel export
Customer Analysts
Interactive
ipsec geo replication
Zookeeper
Master H/A
Synchonization
Real time
Cassandra
Kafka
pub/sub
DR
Cassandra
Monitoring
(Nagios/Graphite)
Workflows
(Oozie/Talend)
RT
xPatterns API
ELB us-east
ELB us-west
DR
xPatterns API
Customer Dashboard/
Application Servers
xPatterns API requests
AWS Route 53 DNS

Hadoop to BDAS

- Hadoop MR -> **Spark**
  - core + graphx
- Hive -> **Shark**
  - Cli + SharkServer2 + ... Jaws!
- NO resource manager - > **Mesos**
  - Spark Job Servers, Jaws, SharkServer2, Hadoop, Aurora
- No Cache -> **Tachyon**
  - sharing data between contexts, satellite cluster file system, faster for long running queries … GC friendlier, survives JVM crashes
- Hadoop distro dashboards-> **Ganglia**
  - + Nagios & Graphite

BDAS to BDAS++

- **Jaws,** xPatterns http Shark server, open sourcing today! http://github.com/Atigeo/http-shark-server

- **Spark Job Server**
  - multiple contexts in same JVM
  - job submission in Java + Scala

- Mesos framework starvation bug
  - fixed … detailed Tech Blog link at http://xpatterns.com/sparksummit

- *SchedulerBackend update, job cancellation in Mesos fine-grained mode, 0.9.0 patches (shuffle spill, Mesos fine-grained)

- Databricks certified!

Spark … 0.8.0 to 1.0

- **0.8.0** - first POC … lots of OOM
- **0.8.1** – first production deployment, still lots of OOM
  - 20 billion healthcare records, 200 TB of compressed hdfs data
  - Hadoop MR: 100 m1.xlarge (4c x 15GB)
  - BDAS: 20 cc2.8xlarge (32c x 60.8 GB), still lots of OOM map & reducer side
  - Perf gains of 4x to 40x, required individual dataset and query fine-tuning
  - Mixed Hive & Shark workloads where it made sense
  - Daily processing reduced from 14 hours to 1.5hours!
- **0.9.0 -** fixes many of the problems, but still requires patches! (spill & mesos fine-grained)
- **1.0** upgrade in progress, Jaws being migrated to Spark SQL
- set mapreduce.job.reduces=…, set shark.column.compress=true, spark.default.parallelism=…, spark.storage.memoryFraction=0.3, spark.shuffle.memoryFraction=0.6, spark.shuffle.consolidateFiles=true, spark.shuffle.spill=false|true,

Distributed Data Ingestion API & GUI

- Highly available, scalable and resilient distributed download tool exposed through Restful API & GUI
- Supports encryption/decryption, compression/decompression, automatic backup & restore (aws S3) and geo-failover (hdfs and S3 in both us-east and us-west ec2 regions)
- Support multiple input sources: sftp, S3 and 450+ sources through Talend Integration
- Configurable throughput (number of parallel Spark processors, in both fine-grained and coarse-grained Mesos modes)
- File Transfer log and file transition state history for auditing purposes (pluggable persistence model, Cassandra/hdfs), configurable alerts, reports
- Ingest + Backup: download + decompression + hdfs persistence + encryption + S3 upload
- Restore: S3 download + decryption + decompress + hdfs persistence
- Geo-failover: backup on S3 us-east + restore from S3 us-east into west-coast hdfs + backup on S3 us-west
- Ingestion jobs can be resumed from any stage after failure (# of Spark task retries exhausted)
- Logs, job status and progress pushed asynchronously to GUI though web sockets
- Http streaming API exposed for high-throughput push model ingestion (ingestion into Kafka pub-sub, batch Spark job for transfer into hdfs)

Datasets: + Add

- MovieLens_100K
- MovieLens_1M
- continous_integration
- test1
- test2

Job definitions | Jobs | Files

All jobs history

Add SFTP job definition | Add S3 job definition

Definition name	Source	Connection
MovieLens_100K_S3	s3	demo-summit
MovieLens_100K_S3_fine_grained	s3	demo-summit

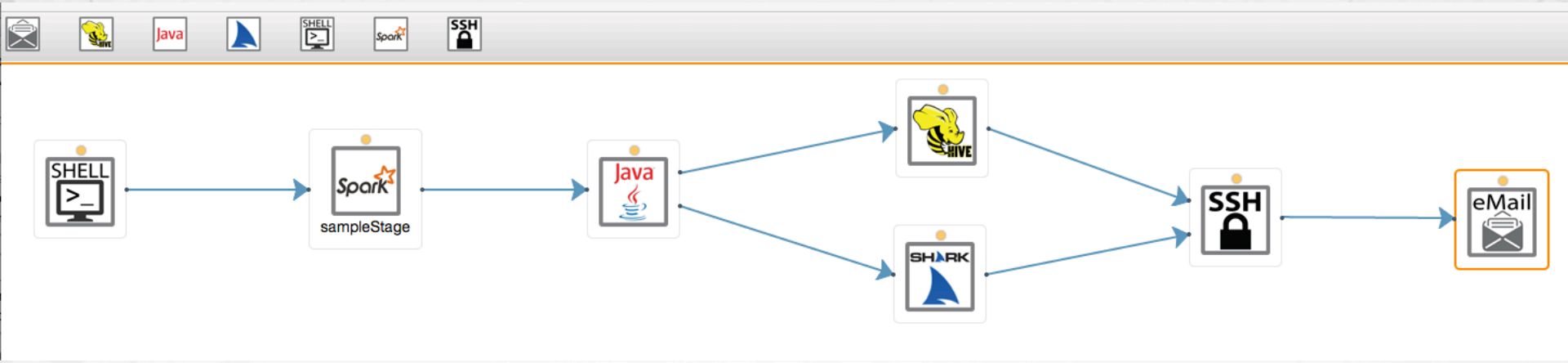
Run | Edit | Clone | Delete

Source: s3
Job name: MovieLens_100K_S3_fine_grained
S3 Bucket: demo-summit
Source folder: ml-100k
Access key: AKIA****
Secret key: +YV7****
Overwrite files: Yes
Parallelism level: Unlimited

MovieLens_1M_S3	s3	demo-summit
MovieLens_1M_S3_finegrained	s3	demo-summit
continuous_integration	s3	xpatterns_ingestion_tests
decompress_bug	s3	xpatterns_ingestion_tests
decompress_bug2	s3	xpatterns_ingestion_tests
decompress_bug4	s3	xpatterns_ingestion_tests
host_probes1	s3	strata_datasets
ip_id1	s3	strata_datasets

Showing 1 - 10 out of 13 « Previous 1 2 Next »

T-Component API & GUI



- Data Transformation component for building a data pipeline with monitoring and quality gates
- Exposes all of Oozie's action types and adds Spark (Java & Scala) and Shark (QL) stages
- Uses our own Spark JobServer (multiple Spark contexts in same JVM!)
- Spark stage required to run code that accepts an xPatterns-managed Spark context (coarse-grained or fine-grained) as parameter
- DAG and job execution info persistence in Hive Metastore
- Exposes full API for job, stages, resources management and scheduled pipeline execution
- Logs, job status and progress pushed asynchronously to GUI though web sockets

- T-component DAG executed by Oozie
- Spark and Shark stages executed through ssh actions
- Spark stage sent to Spark JobServer
- SharSk stage executed through shark CLI for now (SharkServer2 in the future)
- Support for pySpark stage coming soon

Workflow atigeo_demo_job

WORKFLOW
atigeo_demo_job

SUBMITTER
hdfs

STATUS
SUCCEEDED

PROGRESS
100%

ID
0000000-140521145759296-oozie-oozi-W

VARIABLES

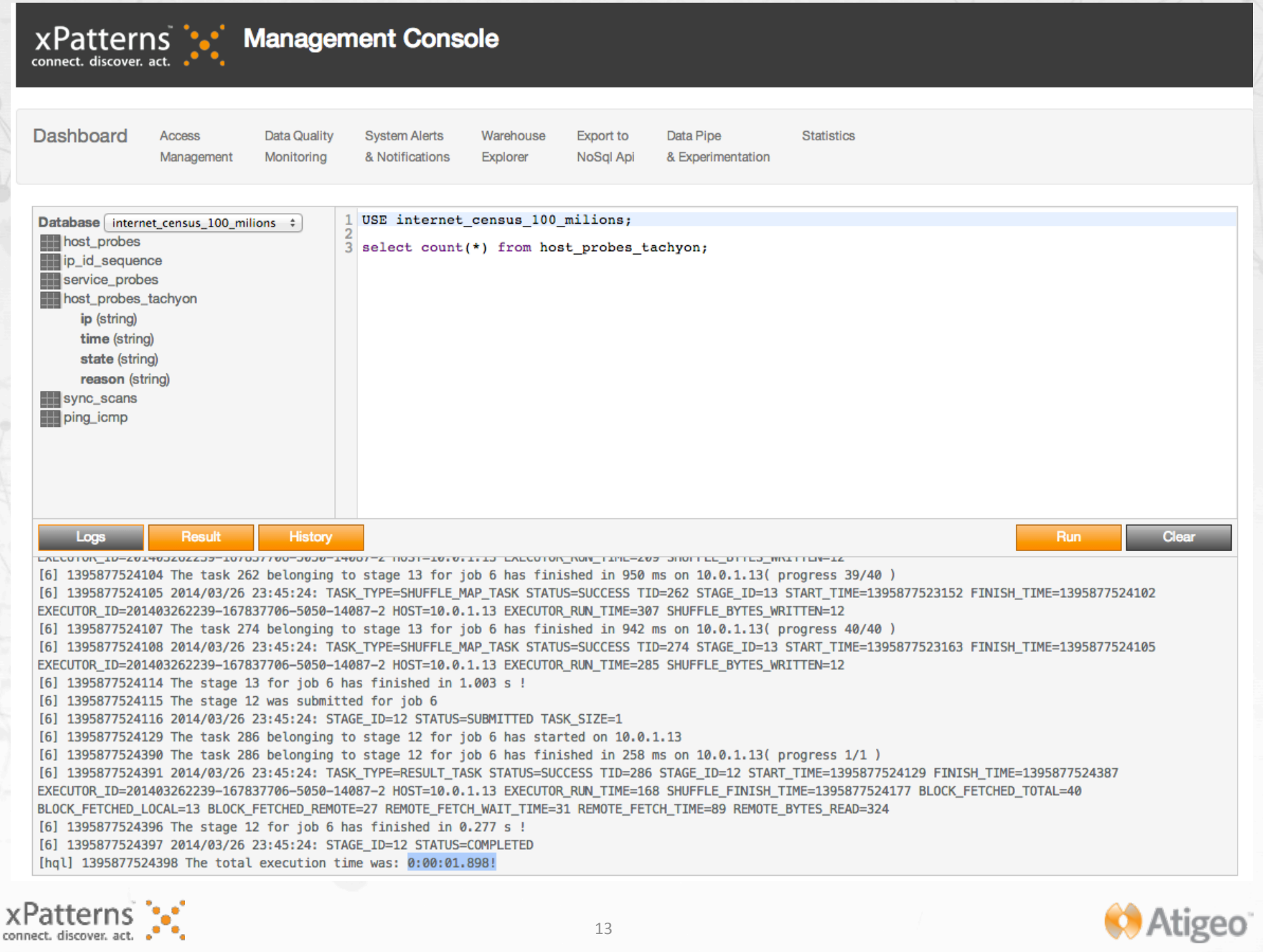
MANAGE
Rerun

Graph | Actions | Details | Configuration | Log | Definition

Logs	Id	Name	Type	Status	External Id	Start Time	End Time	Retries	Error Code	Error Message	Transition	Data
	0000000-140521145759296-oozie-oozi-W@spark_stage	spark_stage	ssh	OK		Wed, 21 May 2014 13:23:34	Wed, 21 May 2014 13:23:50	0			shark_stage	# #Wed May 21 20:23:50 UTC 2014 "status"="OK", *=-type Usage=<main class> [options] }= {= type=of action (shark/spark) Options= location=of the action on hdfs "result"="Job done\!"
	0000000-140521145759296-oozie-oozi-W@shark_stage	shark_stage	ssh	OK		Wed, 21 May 2014 13:23:50	Wed, 21 May 2014 13:24:11	0			ssh_stage	# #Wed May 21 20:24:11 UTC 2014 *=-type Usage=<main class> [options] Moved='hdfs\://ip-10-0-1-18.ec2.internal\:8020/user/hive/warehouse/atigeo_demo.db/service_probes_normalized' to trash at\: hdfs\://ip-10-0-1-18.ec2.internal\:8020/user/ubuntu/.Trash/Current Starting=the Shark Command Line Client type=of action (shark/spark) Options= 2.785= [GC 559232K->22380K(2027264K), 0.0184580 secs] location=of the action on hdfs -f=shark_tables.hql -hivevar dir1\=hdfs\://ip-10-0-1-18.ec2.internal\:8020/user/root/datasets/demoUser/service_probes/clean -hivevar dir2\=hdfs\://ip-10-0-1-18.ec2.internal\:8020/user/root/datasets/demoUser/service_probes/normalized/ -hivevar db\=atigeo_demo /home/ubuntu/latest-mssh/shark-0.9.1/bin/shark=-f shark_tables.hql -hivevar dir1\=hdfs\://ip-10-0-1-18.ec2.internal\:8020/user/root/datasets/demoUser/service_probes/clean -hivevar dir2\=hdfs\://ip-10-0-1-18.ec2.internal\:8020/user/root/datasets/demoUser/service_probes/normalized/ -hivevar db\=atigeo_demo
	0000000-140521145759296-oozie-oozi-W@ssh_stage	ssh_stage	ssh	OK		Wed, 21 May 2014 13:24:11	Wed, 21 May 2014 13:24:16	0			email_stage	# #Wed May 21 20:24:16 UTC 2014
	0000000-140521145759296-oozie-oozi-W@email_stage	email_stage	email	OK		Wed, 21 May 2014 13:24:16	Wed, 21 May 2014 13:24:17	0			end	

Jaws REST SharkServer & GUI

- **Jaws**: a highly scalable and resilient restful (http) interface on top of a managed Shark session that can concurrently and asynchronously submit Shark queries, return persisted results (automatically limited in size or paged), execution logs and job information (Cassandra or hdfs persisted).
- Jaws can be load balanced for higher availability and scalability and it fuels a web-based GUI that is integrated in the xPatterns Management Console (Warehouse Explorer)
- Jaws exposes configuration options for fine-tuning Spark & Shark performance and running against a stand-alone Spark deployment, with or without Tachyon as in-memory distributed file system on top of HDFS, and with or without Mesos as resource manager
- Shark editor provides analysts, data scientists with a view into the warehouse through a metadata explorer, provides a query editor with intelligent features like auto-complete, a results viewer, logs viewer and historical queries for asynchronously retrieving persisted results, logs and query information for both running and historical queries
- web-style pagination and query cancellation, spray io http layer (REST on Akka)
- Open Sourced at the Summit! http://github.com/Atigeo/http-shark-server


xPatterns
connect. discover. act.
Management Console
Dashboard
Access Management
Data Quality Monitoring
System Alerts & Notifications
Warehouse Explorer
Export to NoSql Api
Data Pipe & Experimentation
Statistics
Database internet_census_100_milions
host_probes
ip_id_sequence
service_probes
host_probes_tachyon
ip (string)
time (string)
state (string)
reason (string)
sync_scans
ping_icmp
1 USE internet_census_100_milions;
2
3 select count(*) from host_probes_tachyon;
Logs
Result
History
Run
Clear
[6] 1395877524104 The task 262 belonging to stage 13 for job 6 has finished in 950 ms on 10.0.1.13(progress 39/40)
[6] 1395877524105 2014/03/26 23:45:24: TASK_TYPE=SHUFFLE_MAP_TASK STATUS=SUCCESS TID=262 STAGE_ID=13 START_TIME=1395877523152 FINISH_TIME=1395877524102
EXECUTOR_ID=201403262239-167837706-5050-14087-2 HOST=10.0.1.13 EXECUTOR_RUN_TIME=307 SHUFFLE_BYTES_WRITTEN=12
[6] 1395877524107 The task 274 belonging to stage 13 for job 6 has finished in 942 ms on 10.0.1.13(progress 40/40)
[6] 1395877524108 2014/03/26 23:45:24: TASK_TYPE=SHUFFLE_MAP_TASK STATUS=SUCCESS TID=274 STAGE_ID=13 START_TIME=1395877523163 FINISH_TIME=1395877524105
EXECUTOR_ID=201403262239-167837706-5050-14087-2 HOST=10.0.1.13 EXECUTOR_RUN_TIME=285 SHUFFLE_BYTES_WRITTEN=12
[6] 1395877524114 The stage 13 for job 6 has finished in 1.003 s !
[6] 1395877524115 The stage 12 was submitted for job 6
[6] 1395877524116 2014/03/26 23:45:24: STAGE_ID=12 STATUS=SUBMITTED TASK_SIZE=1
[6] 1395877524129 The task 286 belonging to stage 12 for job 6 has started on 10.0.1.13
[6] 1395877524390 The task 286 belonging to stage 12 for job 6 has finished in 258 ms on 10.0.1.13(progress 1/1)
[6] 1395877524391 2014/03/26 23:45:24: TASK_TYPE=RESULT_TASK STATUS=SUCCESS TID=286 STAGE_ID=12 START_TIME=1395877524129 FINISH_TIME=1395877524387
EXECUTOR_ID=201403262239-167837706-5050-14087-2 HOST=10.0.1.13 EXECUTOR_RUN_TIME=168 SHUFFLE_FINISH_TIME=1395877524177 BLOCK_FETCHED_TOTAL=40
BLOCK_FETCHED_LOCAL=13 BLOCK_FETCHED_REMOTE=27 REMOTE_FETCH_WAIT_TIME=31 REMOTE_FETCH_TIME=89 REMOTE_BYTES_READ=324
[6] 1395877524396 The stage 12 for job 6 has finished in 0.277 s !
[6] 1395877524397 2014/03/26 23:45:24: STAGE_ID=12 STATUS=COMPLETED
[hql] 1395877524398 The total execution time was: 0:00:01.898!

Tachyon Summary

Started:	05-09-2014 13:46:52:728
Uptime:	11 day(s), 6 hour(s), 16 minute(s), and 11 second(s)
Version:	0.4.1
Running Workers:	4

Cluster Usage Summary

Memory Capacity:	60.00 GB
Memory Free / Used:	15.70 GB / 44.30 GB
UnderFS Capacity:	13272.42 GB
UnderFS Free / Used:	11852.42 GB / 1420.01 GB

Pin List

/pinfiles

/pindata

White List

/

Detailed Nodes Summary

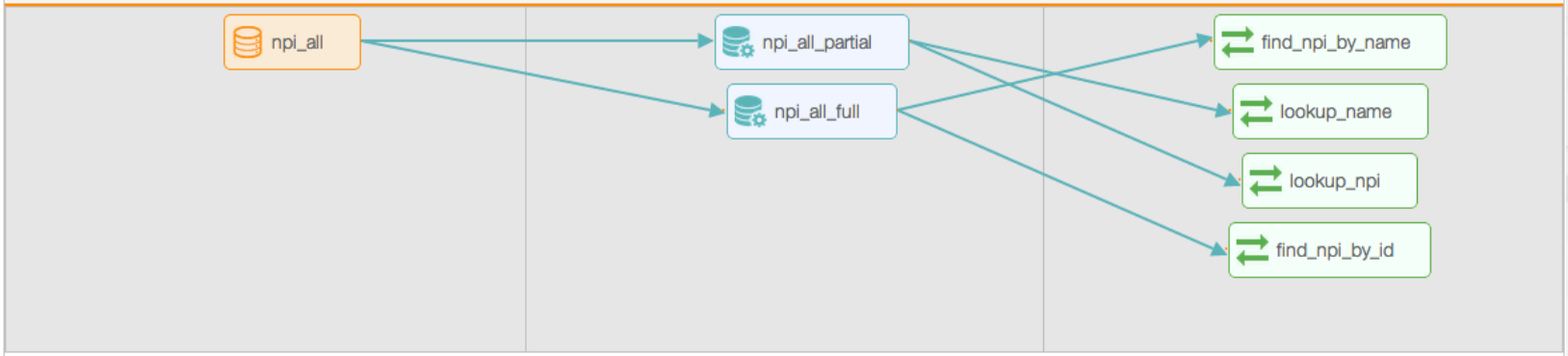
Node Name	[D]Uptime	Last Heartbeat	State	Memory Usage
ip-10-0-1-19	11 d, 6 h, 16 m, and 8 s	0	In Service	78%Used
ip-10-0-1-20	11 d, 6 h, 16 m, and 8 s	0	In Service	67%Used
ip-10-0-1-21	11 d, 6 h, 16 m, and 8 s	0	In Service	77%Used
ip-10-0-1-22	11 d, 6 h, 16 m, and 8 s	0	In Service	71%Used

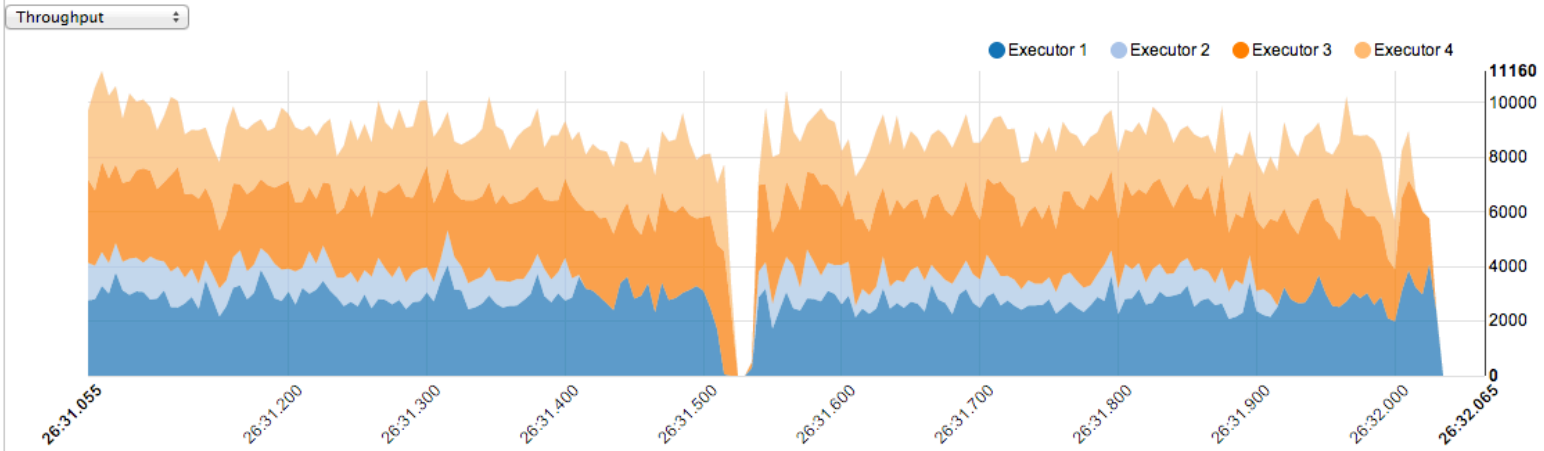
Export to NoSql API

- Datasets in the warehouse need to be exposed to high-throughput low-latency real-time APIs. Each application requires extra processing performed on top of the core datasets, hence additional transformations are executed for building data marts inside the warehouse
- Exporter tool builds the efficient data model and runs an export of data from a Shark/Hive table to a Cassandra Column Family, through a custom Spark job with configurable throughput (configurable Spark processors against a Cassandra ring) (instrumentation dashboard embedded, logs, progress and instrumentation events pushed though SSE)
- Data Modeling is driven by the read access patterns provided by an application engineer building dashboards and visualizations: lookup key, columns (record fields to read), paging, sorting, filtering
- The end result of a job run is a REST API endpoint (instrumented, monitored, resilient, geo-replicated) that uses the underlying generated Cassandra data model and fuels the data in the dashboards
- Configuration API provided for creating export jobs and executing them (ad-hoc or scheduled).
- Logs, job status and progress pushed asynchronously to GUI though web sockets

Job Runs

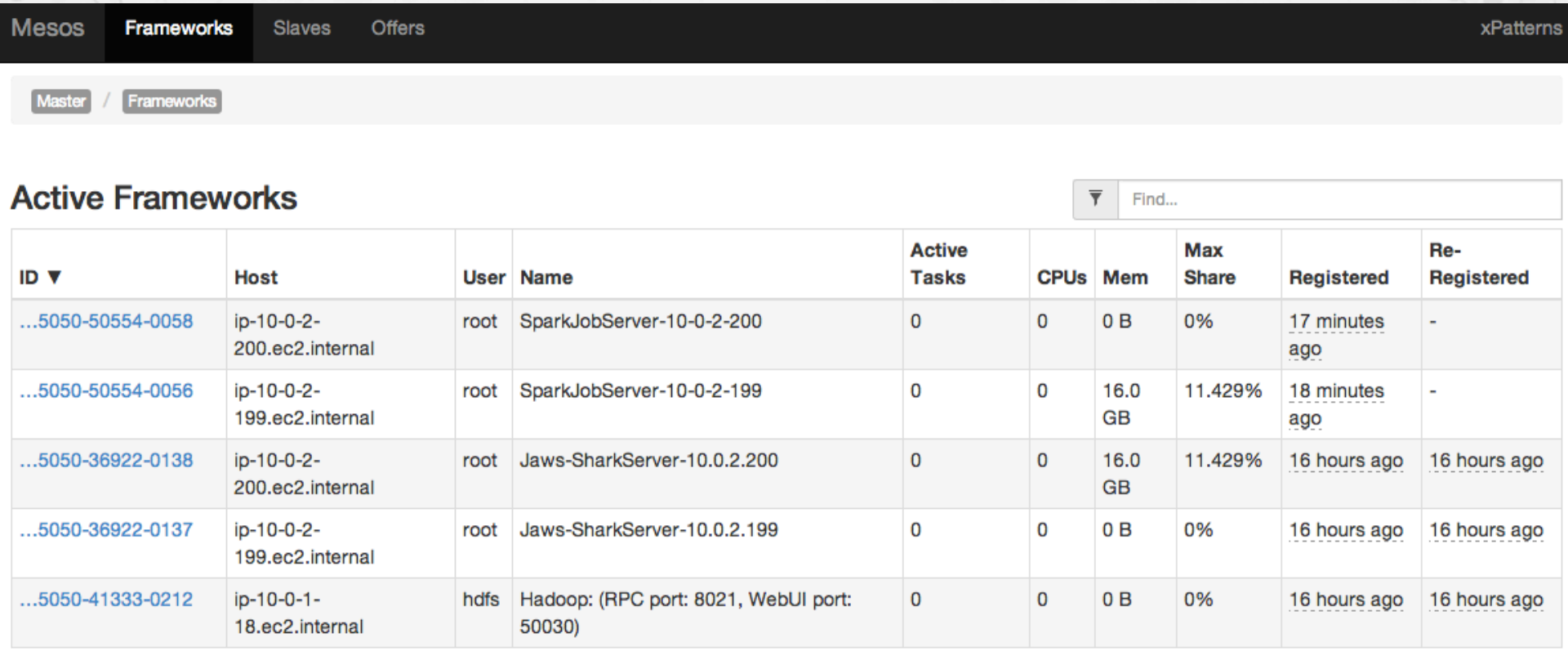
#Run	Start time	Duration	Sources	Mappings	REST Endpoints	Status
5	04/24/2014 6:57:32	17 minutes	npi_all, npi_all	npi_all_partial, npi_all_full	find_npi_by_name, lookup_name, lookup_npi, find_npi_by_id	ONLINE





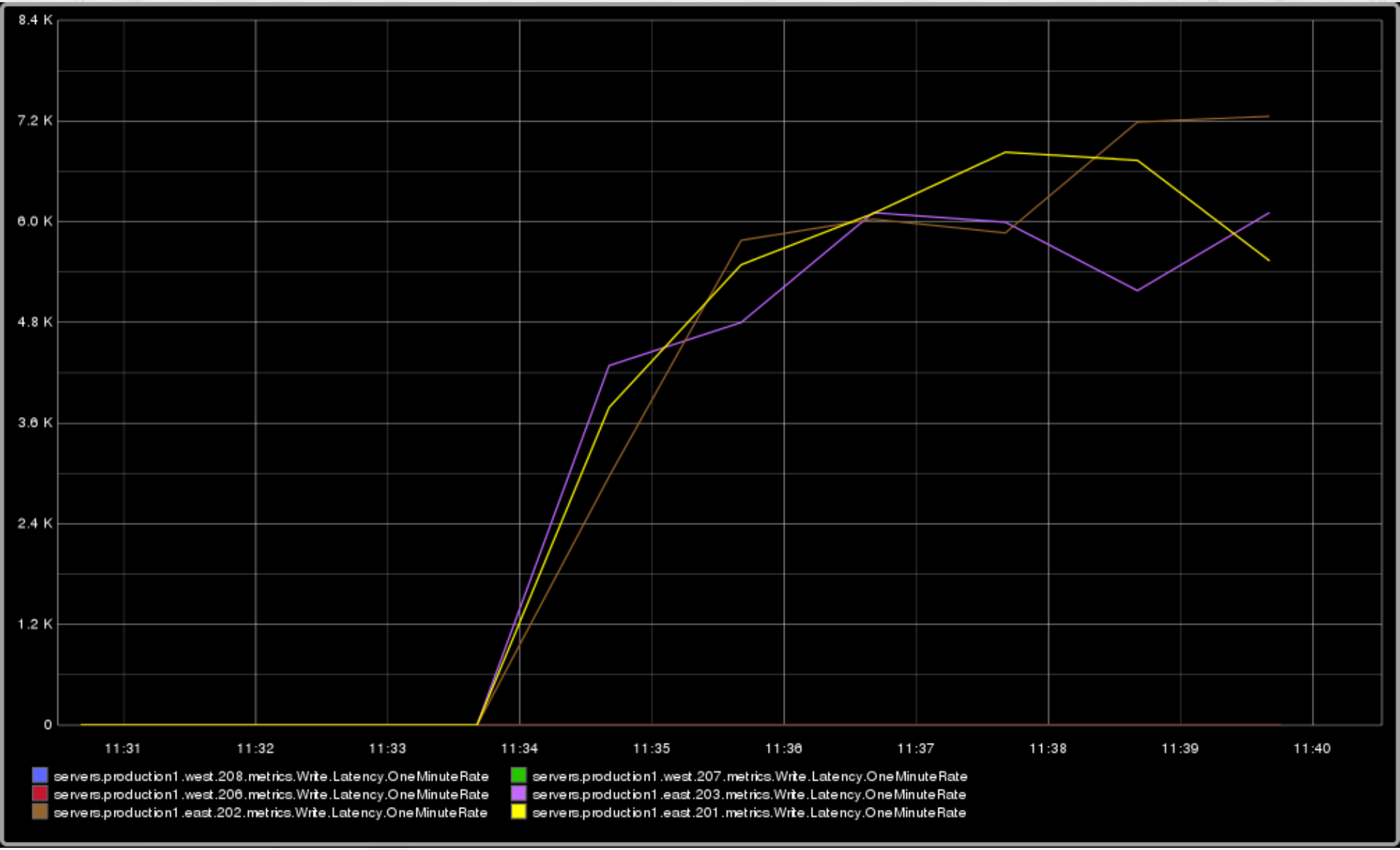

```
[2014-04-25 02:06:04.186] Start data export to PlatformData.npi_all_full_1398391564186 column family in Cassandra ring (10.0.2.201, 10.0.2.202, 10.0.2.203, 176.0.1.206, 176.0.1.206, 176.0.1.207, 176.0.1.208)
[2014-04-25 02:14:25.044] Export complete
Published REST Endpoint find_npi_by_name at: http://services.xpatterns.com/xpatterns-export-nosql-apis/userId/atigeo/jobName/nppes/apiName/find_n
```

Mesos/Spark cluster

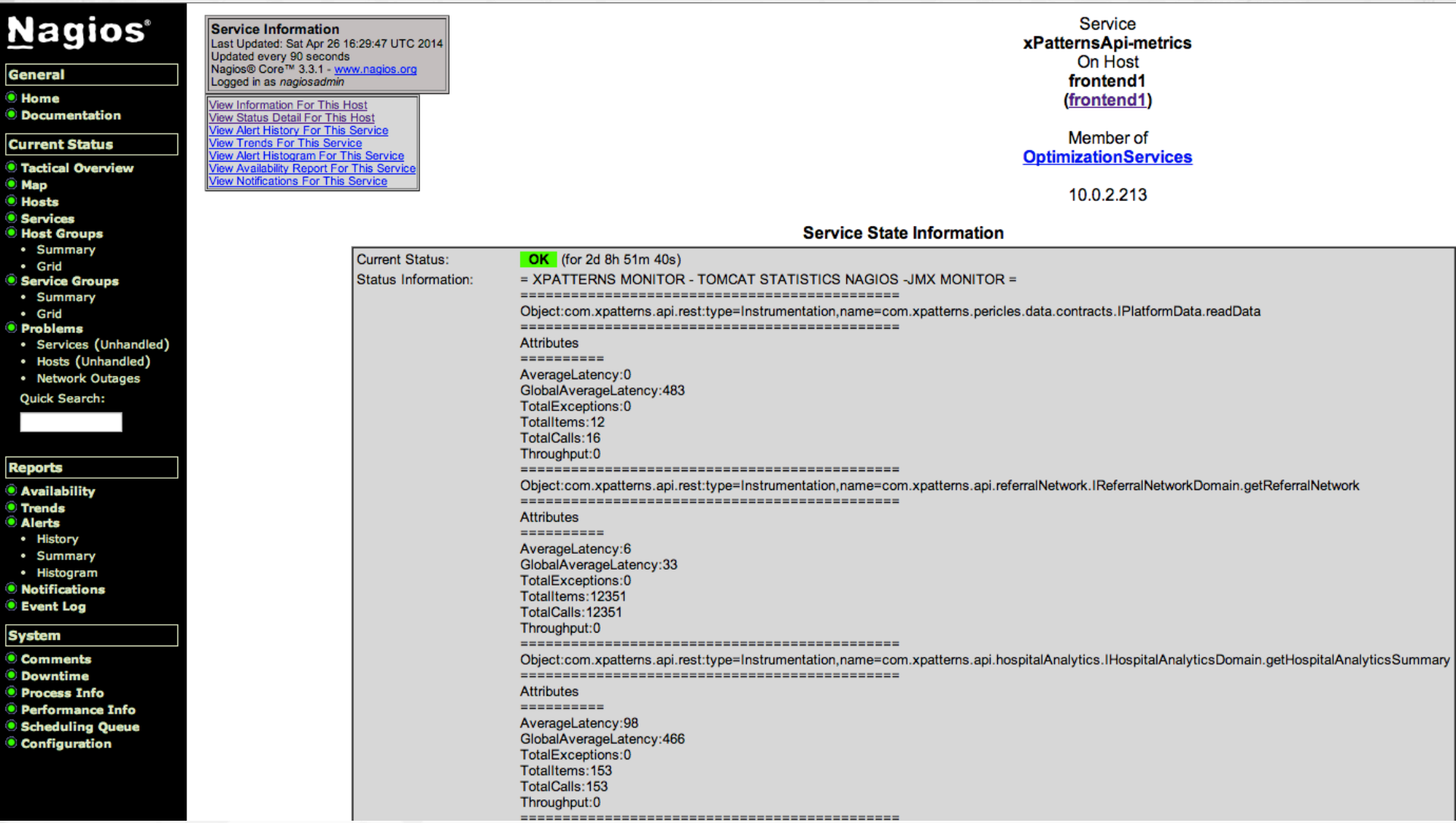


ID ▼	Host	User	Name	Active Tasks	CPUs	Mem	Max Share	Registered	Re-Registered
...5050-50554-0058	ip-10-0-2-200.ec2.internal	root	SparkJobServer-10-0-2-200	0	0	0 B	0%	17 minutes ago	-
...5050-50554-0056	ip-10-0-2-199.ec2.internal	root	SparkJobServer-10-0-2-199	0	0	16.0 GB	11.429%	18 minutes ago	-
...5050-36922-0138	ip-10-0-2-200.ec2.internal	root	Jaws-SharkServer-10.0.2.200	0	0	16.0 GB	11.429%	16 hours ago	16 hours ago
...5050-36922-0137	ip-10-0-2-199.ec2.internal	root	Jaws-SharkServer-10.0.2.199	0	0	0 B	0%	16 hours ago	16 hours ago
...5050-41333-0212	ip-10-0-1-18.ec2.internal	hdfs	Hadoop: (RPC port: 8021, WebUI port: 50030)	0	0	0 B	0%	16 hours ago	16 hours ago

Cassandra multi DC ring – write latency



Nagios monitoring



Ganglia .sourceforge.net

Atigeo Grid Report for Sat, 24 May 2014 01:50:51 +0000 Get Fresh

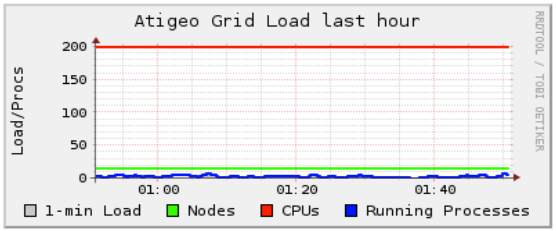
Last hour **Sorted** descending

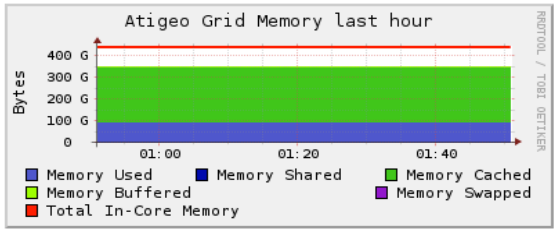
Atigeo Grid > --Choose a Source

Atigeo Grid (5 sources) (tree view)

CPUs Total: **200**
Hosts up: **15**
Hosts down: **0**

Avg Load (15, 5, 1m):
1%, 1%, 1%
Localtime:
2014-05-24 01:50

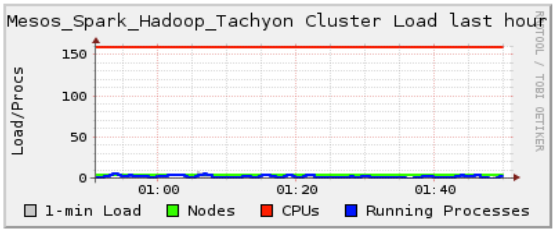


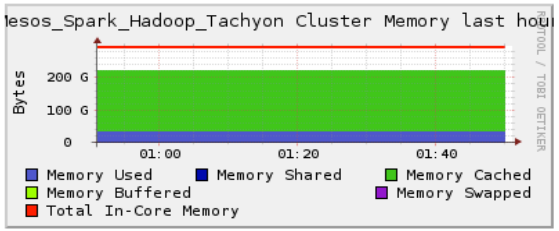


Mesos_Spark_Hadoop_Tachyon (physical view)

CPUs Total: **160**
Hosts up: **5**
Hosts down: **0**

Avg Load (15, 5, 1m):
1%, 1%, 1%
Localtime:
2014-05-24 01:50

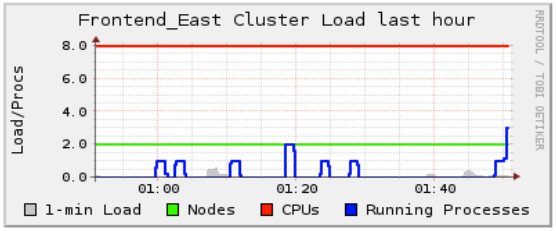


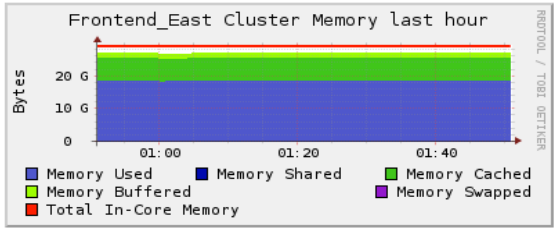


Frontend_East (physical view)

CPUs Total: **8**
Hosts up: **2**
Hosts down: **0**

Avg Load (15, 5, 1m):
1%, 1%, 1%
Localtime:
2014-05-24 01:51

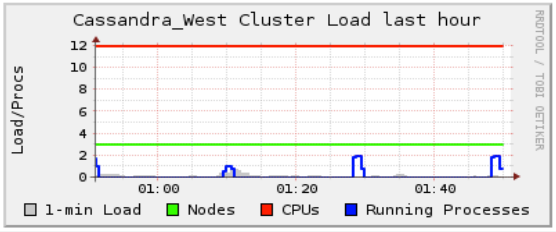


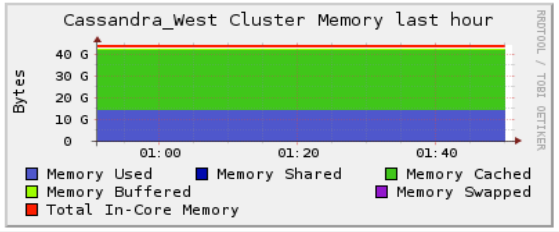


Cassandra_West (physical view)

CPUs Total: **12**
Hosts up: **3**
Hosts down: **0**

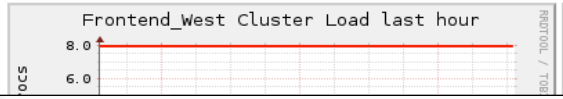
Avg Load (15, 5, 1m):
1%, 0%, 0%
Localtime:
2014-05-24 01:50

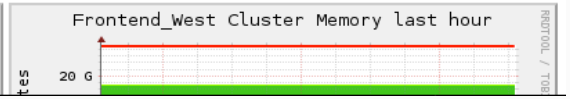




Frontend_West (physical view)

CPUs Total: **8**
Hosts up: **2**
Hosts down: **0**





Coming soon …

- Export to Semantic Search API (solrCloud/lucene)
- pySpark Job Server
- pySpark ←→ Shark/Tachyon interop (either)
- pySpark ←→ Spark SQL (1.0) interop (or)
- Parquet columnar storage for warehouse data

We need your feedback!

Be the first to test new features, get updates, and give feedback by signing up at

http://xpatterns.com/sparksummit

- claudiu.barbura@atigeo.com
- @claudiubarbura
- @atigeo