Experiences in Delivering Spark as a Service

July 27 2016

Michael Feiman, Khalid Ahmed
STSM, IBM Spectrum Computing

Agenda

1 Bluemix Spark Cloud Technical Challenges

2 Bluemix Spark Cloud Architecture & Implementation

3 IBM – Spark & Mesos Community

IBM Spectrum Computing
Infrastructure software for high performance applications

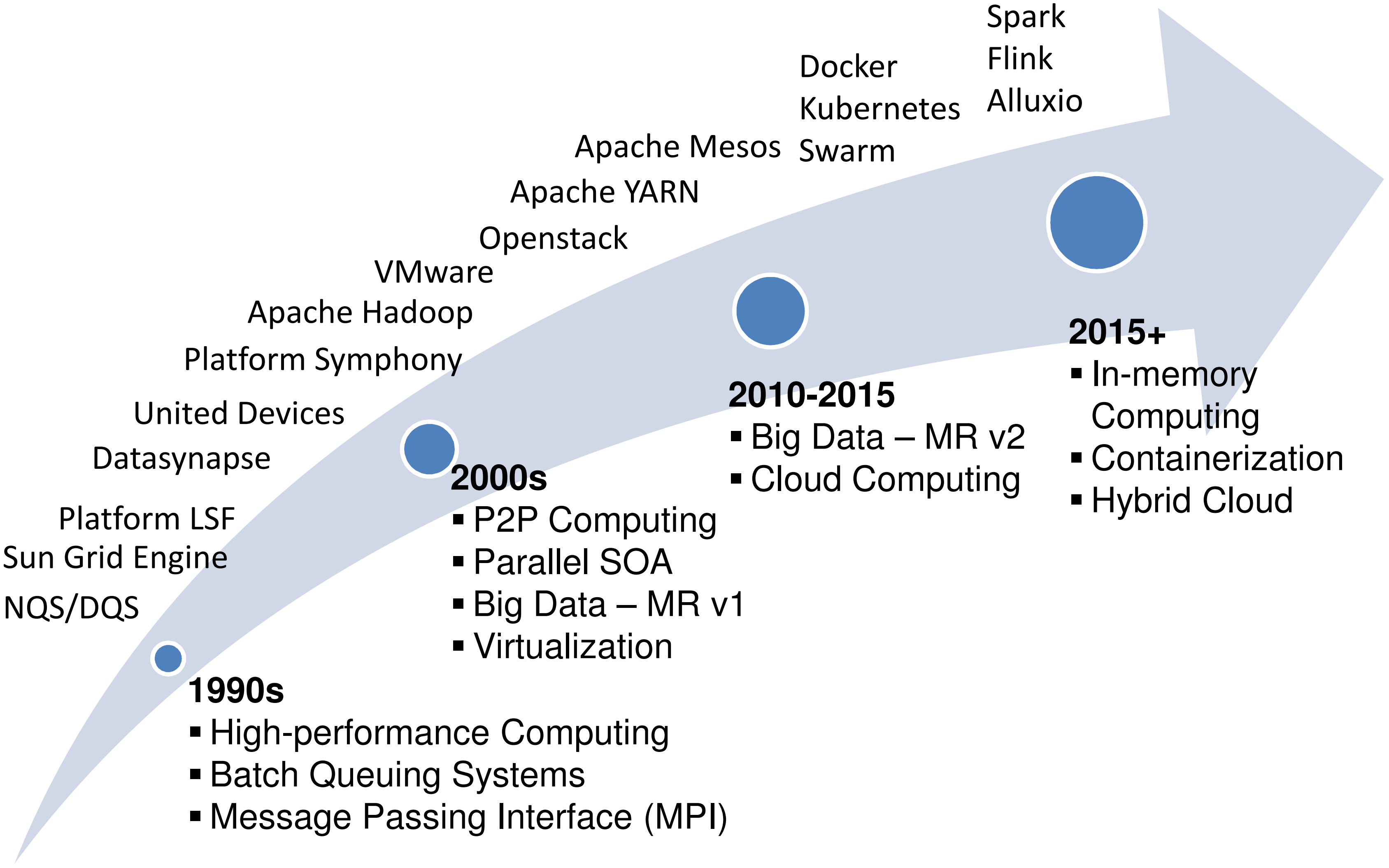
- Acquired by IBM in 2012
- 20 years managing distributed scale-out systems with 2000+ customers in many industries
- Market leading workload, resource and cluster management
- Unmatched scalability (small clusters to global grids) and enterprise production-proven reliability
- Heterogeneous environments – x86 and Power plus 3rd party systems, virtual and bare metal, accelerators / GPU, cloud, etc.
- Shared services for both compute and data intensive workloads

23 of 30 largest commercial enterprises

Over 5M CPUs under management

60% of top financial services companies

History of Distributed Cluster Management



1990s
- High-performance Computing
- Batch Queuing Systems
- Message Passing Interface (MPI)

2000s
- P2P Computing
- Parallel SOA
- Big Data – MR v1
- Virtualization

2010-2015
- Big Data – MR v2
- Cloud Computing

2015+
- In-memory Computing
- Containerization
- Hybrid Cloud

Agenda

1 Bluemix Spark Cloud Technical Challenges

2 Bluemix Spark Cloud Architecture & Implementation

3 IBM – Spark & OSS Community

IBM's vision for IBM Analytics for Apache Spark (Spark-as-a-Service)

We make Spark **ACCESSIBLE** and **USEFUL**

Spark is Important!

Standalone

Spark as a Service

IBM Spectrum Conductor with Spark (on-prem)

Within **Platforms**

IBM Open Platform (w/ Spark)

BigInsights (w/ Spark)

IBM Streams

…many others underway

Within **Solutions**

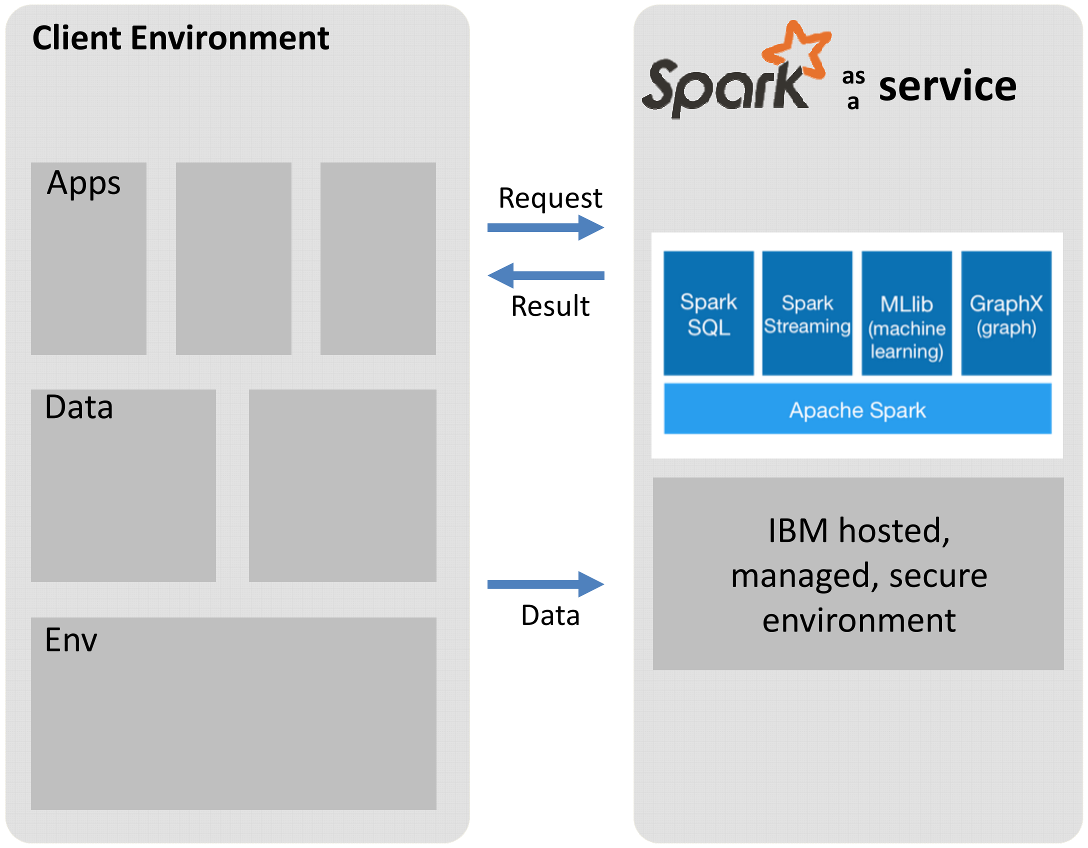
Analytics

Commerce

Watson Health

…many others underway

IBM’s Analytics for Apache Spark offering



What it is:

- Fully-managed Spark environment accessible on-demand for interactive and batch workload

What you get:

- Access to Spark’s next-generation performance and capabilities, including built-in machine learning and others
- Pay only for what you use in either a pay-as-you-go model or through dedicated, enterprise instances
- No lock-in – 100% standard Spark
- Elastic scaling – start with experimentation, extend to development and scale to production, all within the same environment
- Quick start – service is immediately ready for analysis, skipping setup hurdles, hassles and time
- Peace of mind – fully managed and secured, no DBAs or other admins necessary

Common Spark use cases

1. Interactive querying of very large data sets (e.g. BI)
2. Running large data processing batch jobs (e.g. nightly ETL from production systems, primary Hadoop use case)
3. Complex analytics and data mining across various types of data
4. Building and deploying rich analytics models (e.g. risk metrics)
5. Implementing near-realtime stream event processing (e.g. fraud / security detection)

Spark-as-a-Service targets 4 key personas

Data Scientist

- Access powerful tools to tease out the insights they’re looking for, then make them actionable immediately

```
square <- function(x)
{
    return(x*x)
}
…
```

```
{
    “employees” : {
        “ID” : “1234”
        …
    }
}
```

App Developer

- Add intelligence to their apps in a simple and straightforward no-hassle manner

Business Analyst

- Answer the questions that the organization needs quickly and easily, and without getting IT involved

Data Engineer

- Easily build data pipelines that power dashboards and data platforms while ensuring high quality

```
SELECT
    FirstName, LastName
FROM
    employees
WHERE
    id = 1234;
```

```
# mysqlimport -u root -
ptmppassword --local test
employee.txt
```

Challenges in managing Spark applications

- Creating infrastructure silos to accommodate apps is inefficient

Low Utilization
= Higher cost

Many new solution workloads in addition to existing apps → ***Leads to costly, complex, siloed, under-utilized infrastructure and replicated data***

Multi-User/Multi-Tenancy Support

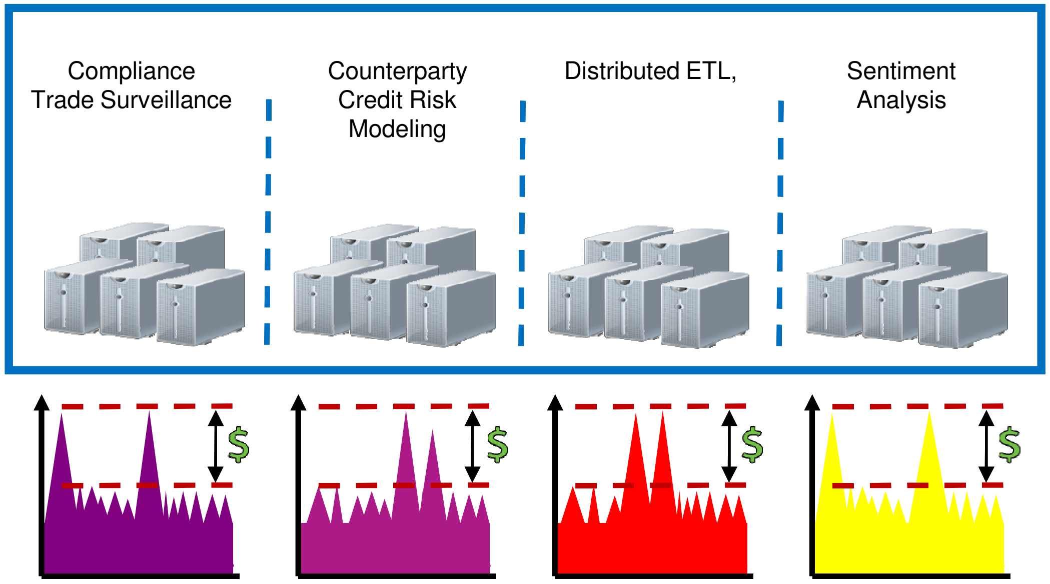
- Different LOBs
- Siloed organization
- Application SLA
- DEV, UAT, PROD

Spark Lifecycle Management

- Multiple Spark versions
- Different notebooks and versions
- Different data sources, e.g., HDFS, Cassandra

Enterprise Production Barriers

- Existing applications
- Security, governance
- Monitoring & Logging
- Limited by technology
- It is new!…


Compliance Trade Surveillance
Counterparty Credit Risk Modeling
Distributed ETL,
Sentiment Analysis
$
$
$
$

Multi Tenant Spark Cloud Options

- Spark Stand-alone Cluster in VMs
- Use Resource Manager like Mesos/YARN
- Spark-centric Workload Management

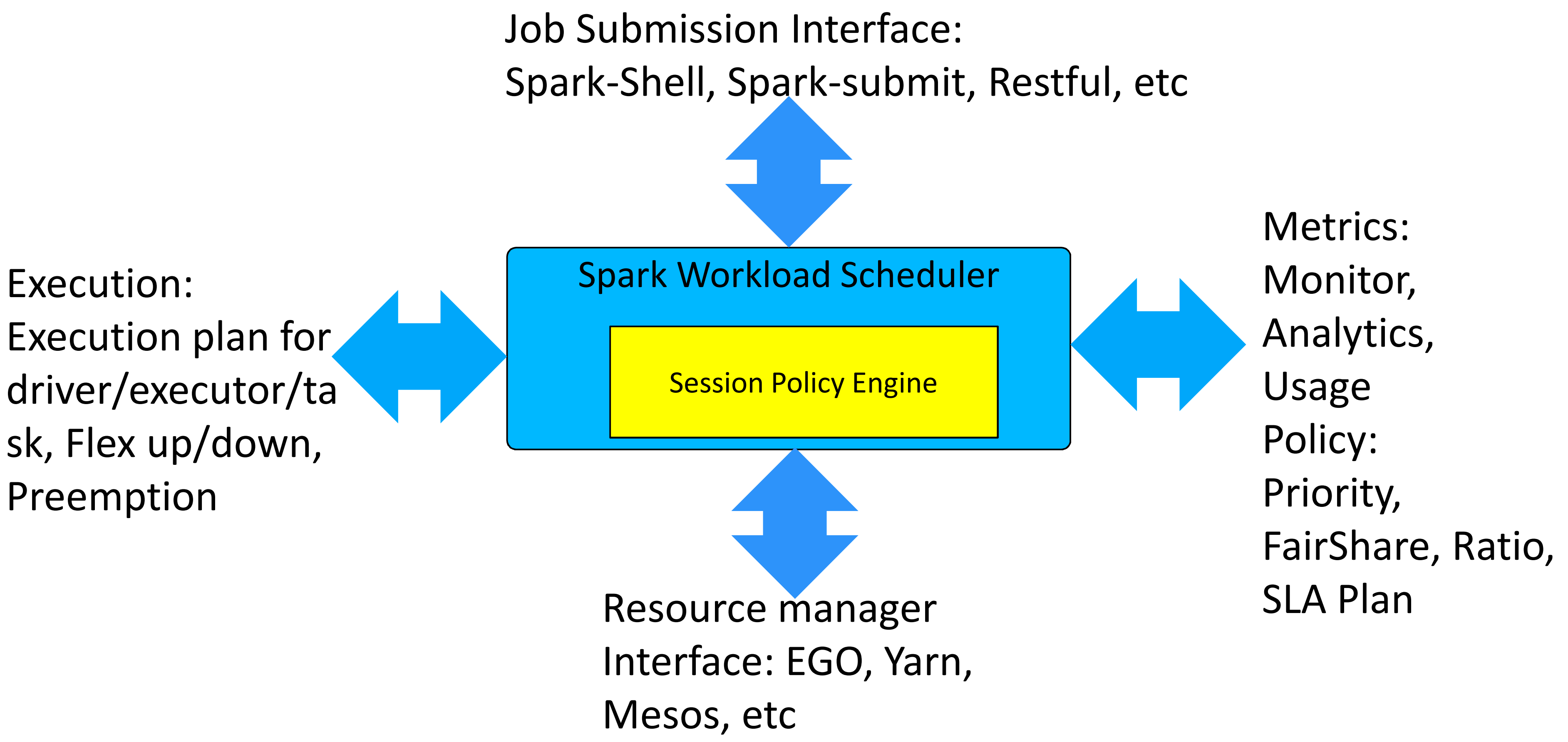
Challenges in Spark Standalone

- Adoption – High
  - ✓ 48% user are using standalone deployment
  - ✓ Major cloud provider – Google Dataproc, Databricks, Amazon
- However …
  - ❖ Single user only – no user grouping for jobs, no authentication, no execution control
  - ❖ No SLA management, application scheduling is not based on real biz priority
  - ❖ Static allocation, rely on user estimate for resource allocation, which needs expertise
  - ❖ No preemption, there could be starvation and deadlock situations
  - ❖ Personal clusters – many silos, high cost in HW and operation/management.
  - ❖ Cannot support multiple frameworks.

Challenges with Spark on Mesos/YARN

- Adoption – Low
  - ✓ 11% user are using Mesos
  - ✓ Mesos intends to to share the resources across hybird workloads
- However …
  - ❖ Fine grained scheduling not really working, no longer supported – Cannot effectively support notebook interactive use case
  - ❖ Single user is running ok, multi-user support requires workload management in Spark scheduler
  - ❖ No SLA management, job round trip time is NOT deterministic when resources are constrained vs demand
  - ❖ Offering mechanism needs to consider resource requirement and demand - lead to in-efficiency, starvation (greedy job keep entire cluster), low utilization.

Need: Workload Management for Spark Environments


Job Submission Interface:
Spark-Shell, Spark-submit, Restful, etc
Spark Workload Scheduler
Session Policy Engine
Execution:
Execution plan for driver/executor/task, Flex up/down, Preemption
Metrics:
Monitor,
Analytics,
Usage
Policy:
Priority,
FairShare, Ratio,
SLA Plan
Resource manager Interface: EGO, Yarn, Mesos, etc

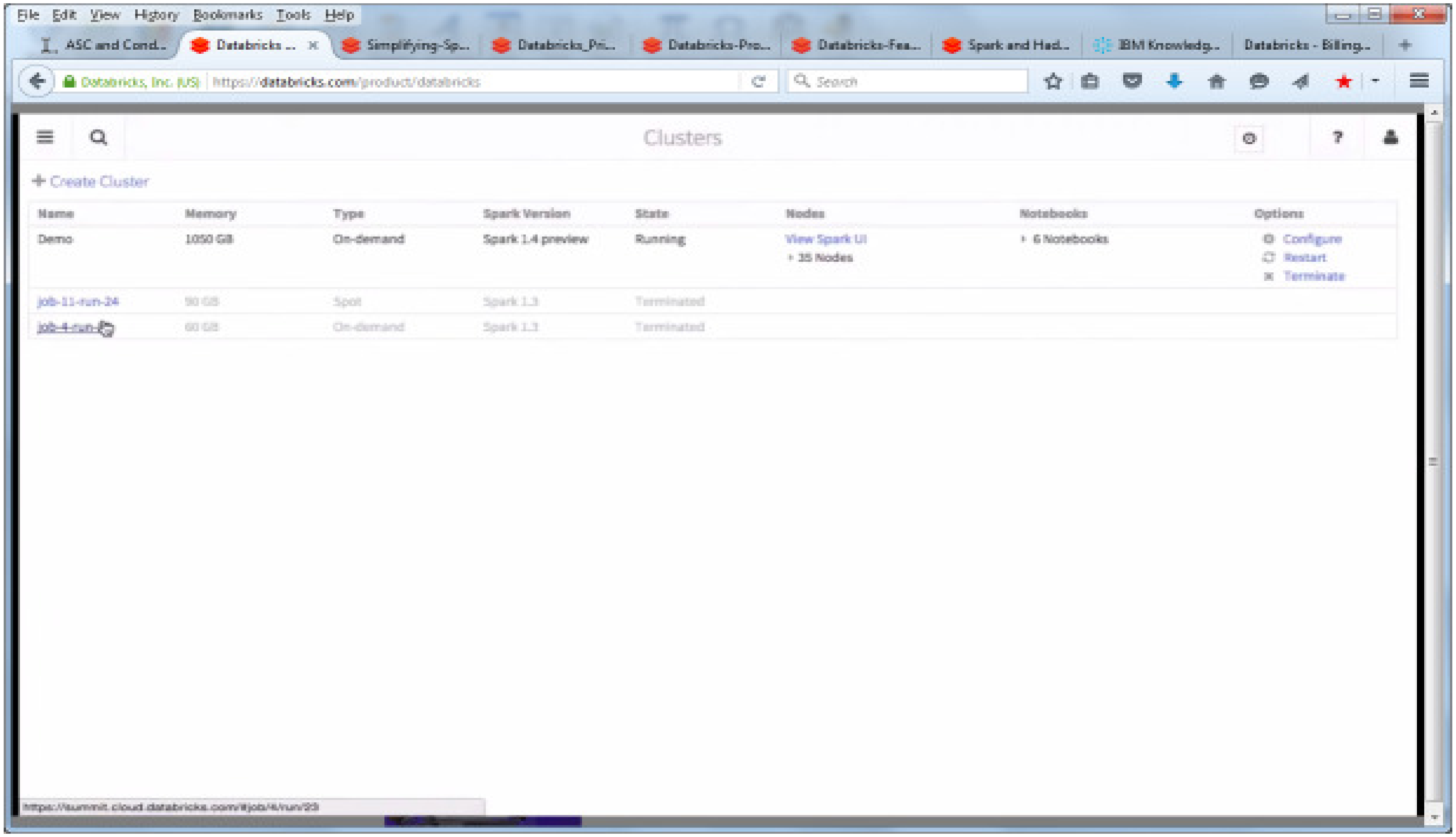
Agenda

1 Bluemix Spark Cloud Technical Challenges

2 Bluemix Spark Cloud Architecture & Implementation

3 IBM – Spark & Mesos Community

Spark cloud approach: Spark Clusters



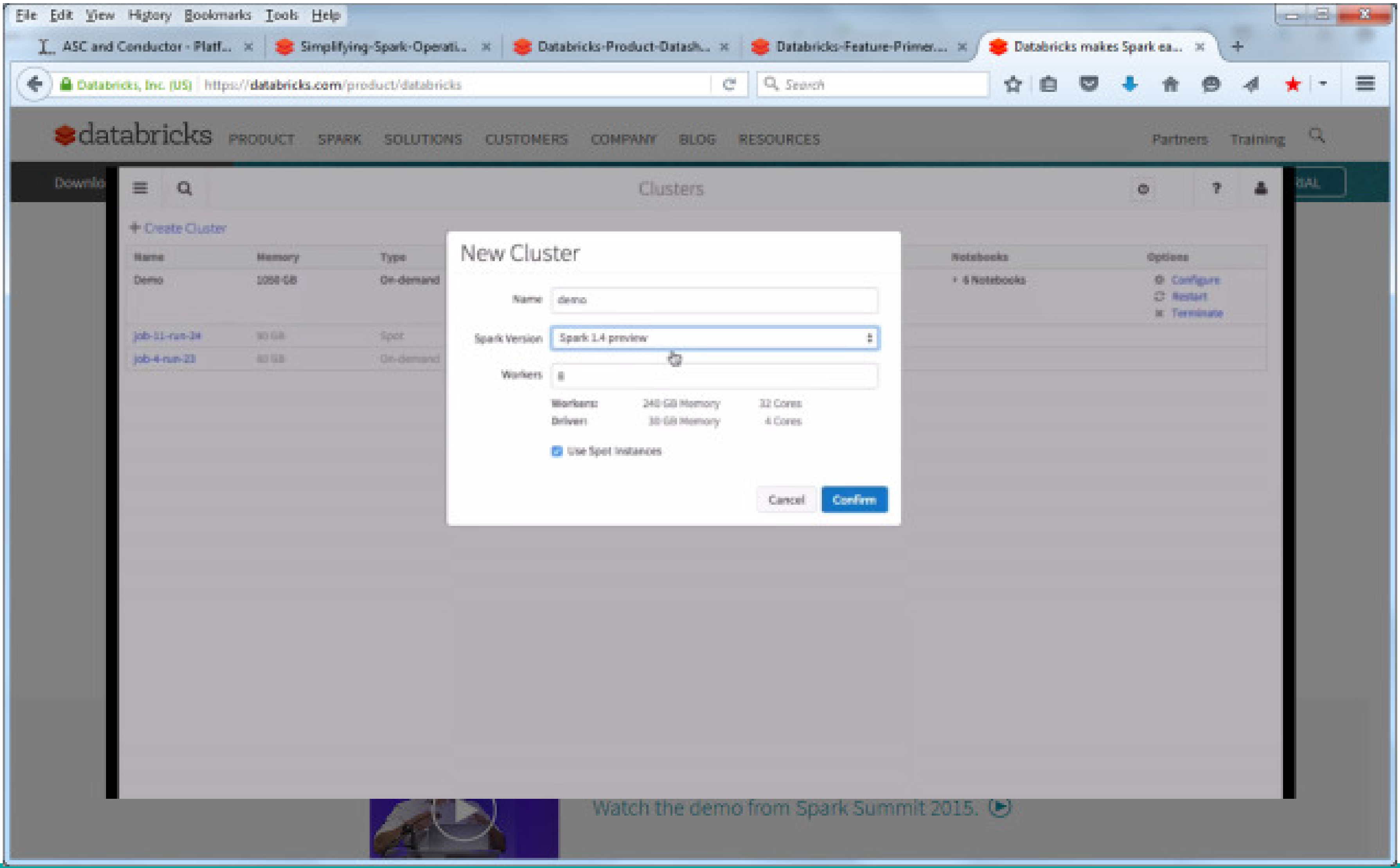
Clusters

+ Create Cluster

Name	Memory	Type	Spark Version	State	Nodes	Notebooks	Options
Demo	1050 GB	On-demand	Spark 1.4 preview	Running	View Spark UI › 35 Nodes	› 6 Notebooks	Configure Restart Terminate
job-11-run-24	90 GB	Spot	Spark 1.3	Terminated			
job-4-run-	60 GB	On-demand	Spark 1.3	Terminated			

https://summit.cloud.databricks.com/#job/4/run/23

Spark Cloud approach:
New Spark Cluster per tenant


File Edit View History Bookmarks Tools Help
ASC and Conductor - Platf...
Simplifying-Spark-Operati...
Databricks-Product-Datash...
Databricks-Feature-Primer...
Databricks makes Spark ea...
Databricks, Inc. (US) https://databricks.com/product/databricks
Search
databricks PRODUCT SPARK SOLUTIONS CUSTOMERS COMPANY BLOG RESOURCES Partners Training
Clusters
+ Create Cluster
Watch the demo from Spark Summit 2015.

Name	Memory	Type	Notebooks	Options
Demo	1088 GB	On-demand	6 Notebooks	Configure, Restart, Terminate
job-11-run-24	90 GB	Spot		
job-4-run-23	60 GB	On-demand		

New Cluster

Name: demo

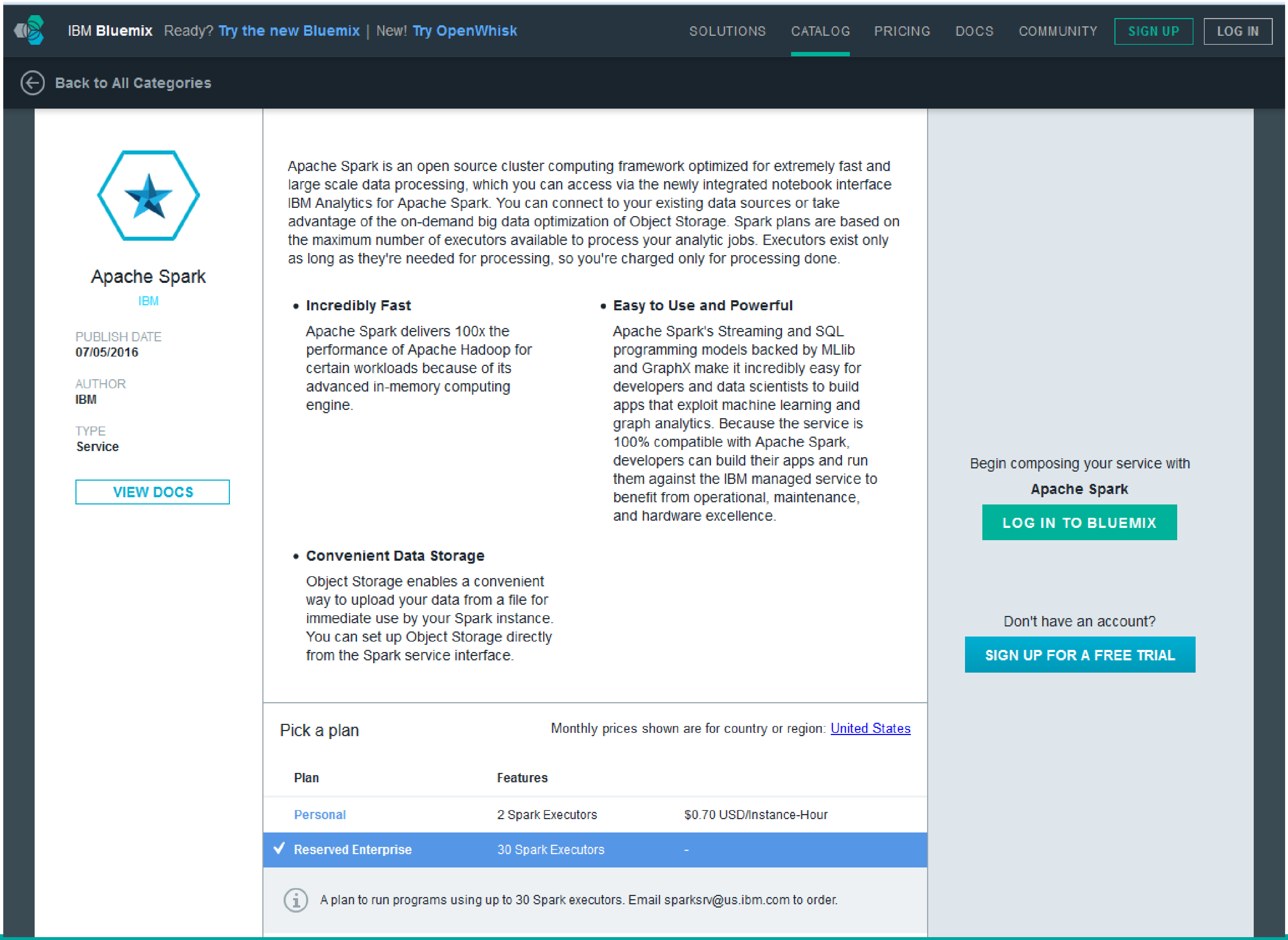
Spark Version: Spark 1.4 preview

Workers: 8

Workers:	240 GB Memory	32 Cores
Driver:	30 GB Memory	4 Cores

[✓] Use Spot Instances

Cancel | Confirm



Apache Spark

IBM

PUBLISH DATE
07/05/2016

AUTHOR
IBM

TYPE
Service

Apache Spark is an open source cluster computing framework optimized for extremely fast and large scale data processing, which you can access via the newly integrated notebook interface IBM Analytics for Apache Spark. You can connect to your existing data sources or take advantage of the on-demand big data optimization of Object Storage. Spark plans are based on the maximum number of executors available to process your analytic jobs. Executors exist only as long as they're needed for processing, so you're charged only for processing done.

- **Incredibly Fast**

  Apache Spark delivers 100x the performance of Apache Hadoop for certain workloads because of its advanced in-memory computing engine.

- **Easy to Use and Powerful**

  Apache Spark's Streaming and SQL programming models backed by MLlib and GraphX make it incredibly easy for developers and data scientists to build apps that exploit machine learning and graph analytics. Because the service is 100% compatible with Apache Spark, developers can build their apps and run them against the IBM managed service to benefit from operational, maintenance, and hardware excellence.

- **Convenient Data Storage**

  Object Storage enables a convenient way to upload your data from a file for immediate use by your Spark instance. You can set up Object Storage directly from the Spark service interface.

Begin composing your service with **Apache Spark**

Don't have an account?

Pick a plan

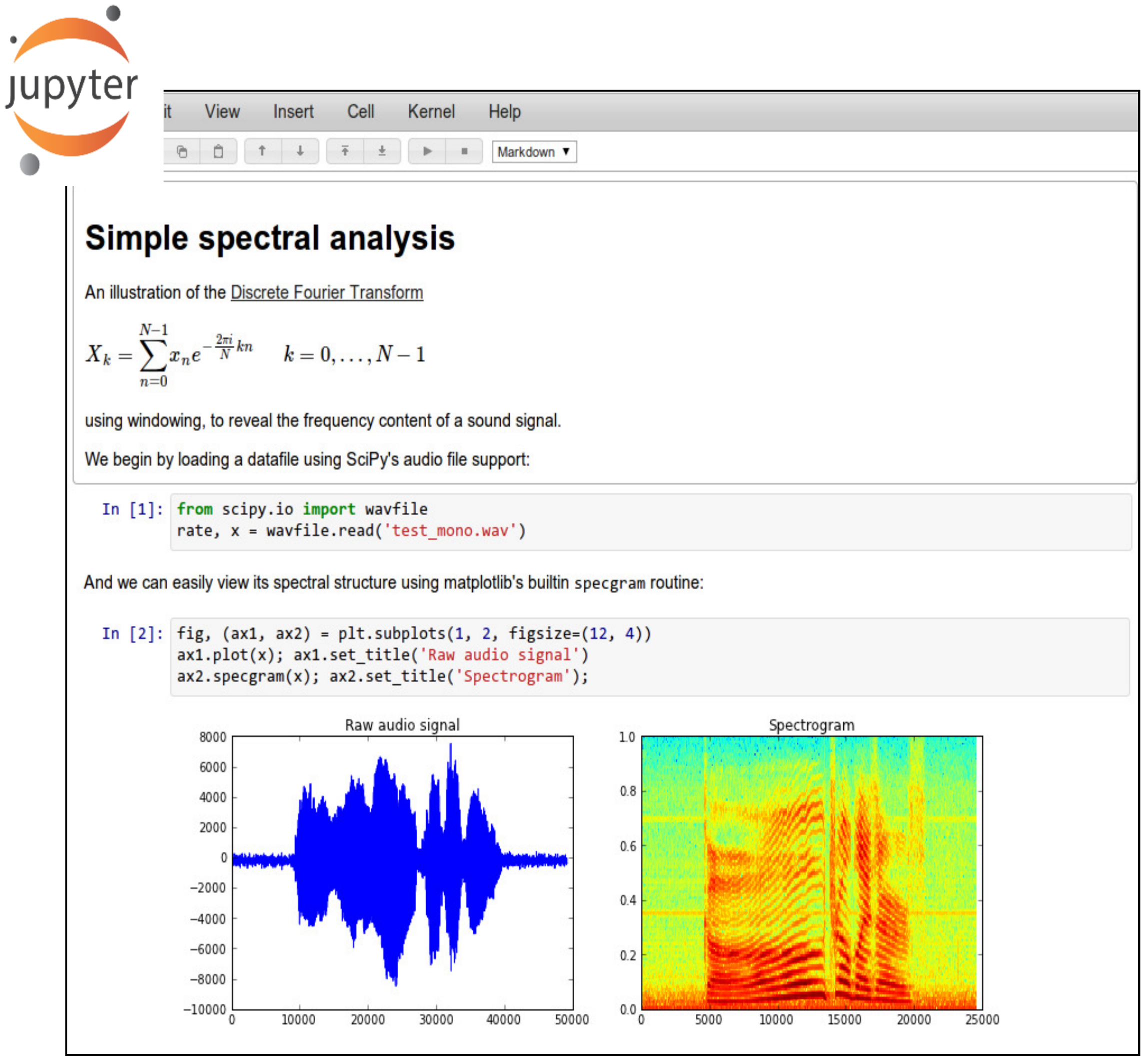
Monthly prices shown are for country or region: United States

Plan	Features	
Personal	2 Spark Executors	$0.70 USD/Instance-Hour
✓ Reserved Enterprise	30 Spark Executors	-

A plan to run programs using up to 30 Spark executors. Email sparksrv@us.ibm.com to order.

Jupyter interactive notebooks

Notebooks are **interactive** computational environments, in which you can **combine** code execution, rich text, mathematics, plots and rich media.



Simple spectral analysis

An illustration of the Discrete Fourier Transform

$$X_k = \sum_{n=0}^{N-1} x_n e^{-\frac{2\pi i}{N} kn} \qquad k = 0, \ldots, N-1$$

using windowing, to reveal the frequency content of a sound signal.

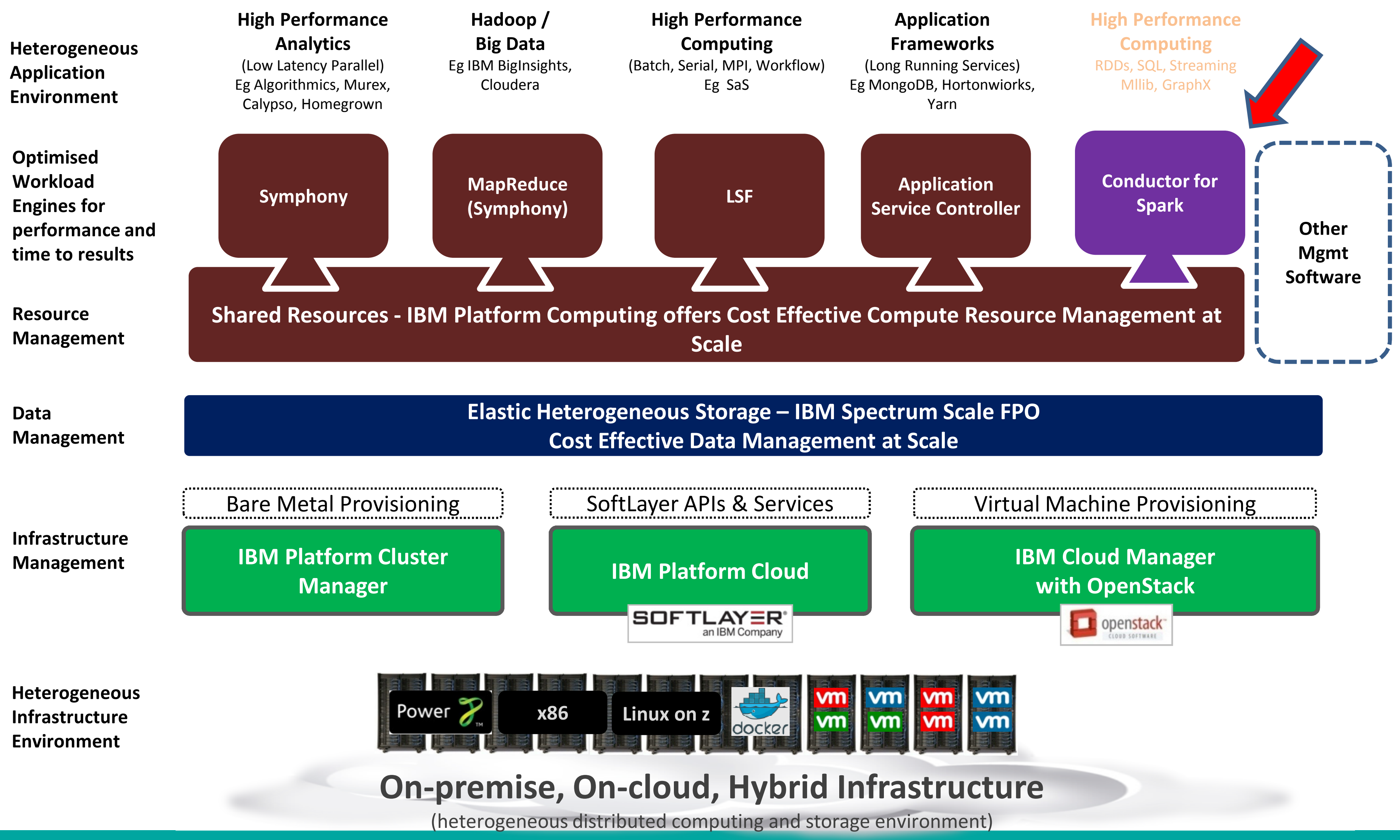
We begin by loading a datafile using SciPy's audio file support:

```
In [1]: from scipy.io import wavfile
        rate, x = wavfile.read('test_mono.wav')
```

And we can easily view its spectral structure using matplotlib's builtin `specgram` routine:

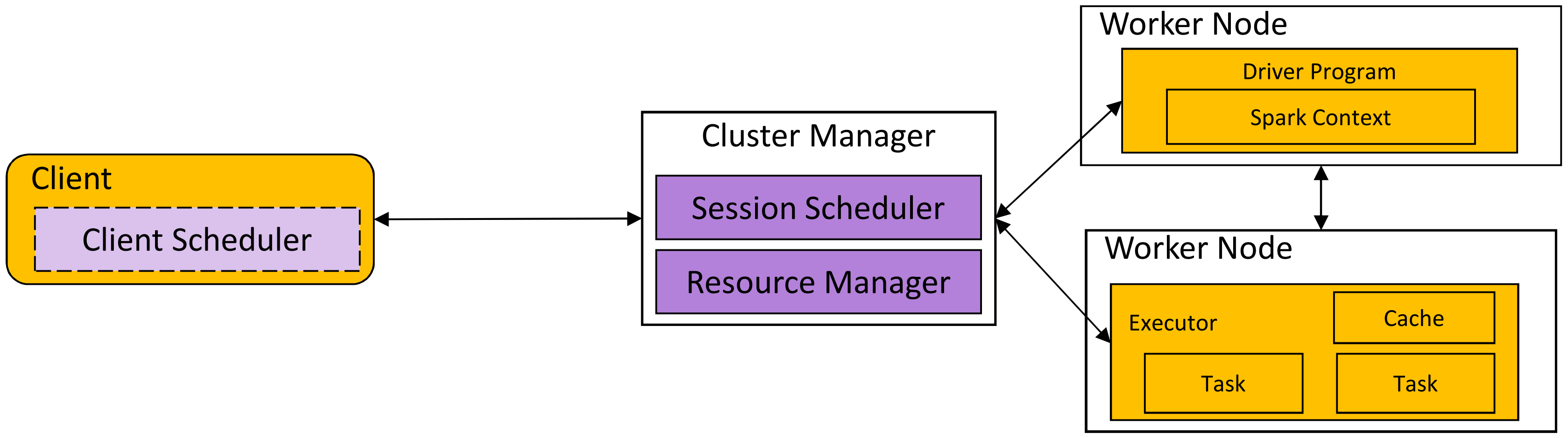
```
In [2]: fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(12, 4))
        ax1.plot(x); ax1.set_title('Raw audio signal')
        ax2.specgram(x); ax2.set_title('Spectrogram');
```

IBM Software Defined Infrastructure Extends to Conductor for Spark

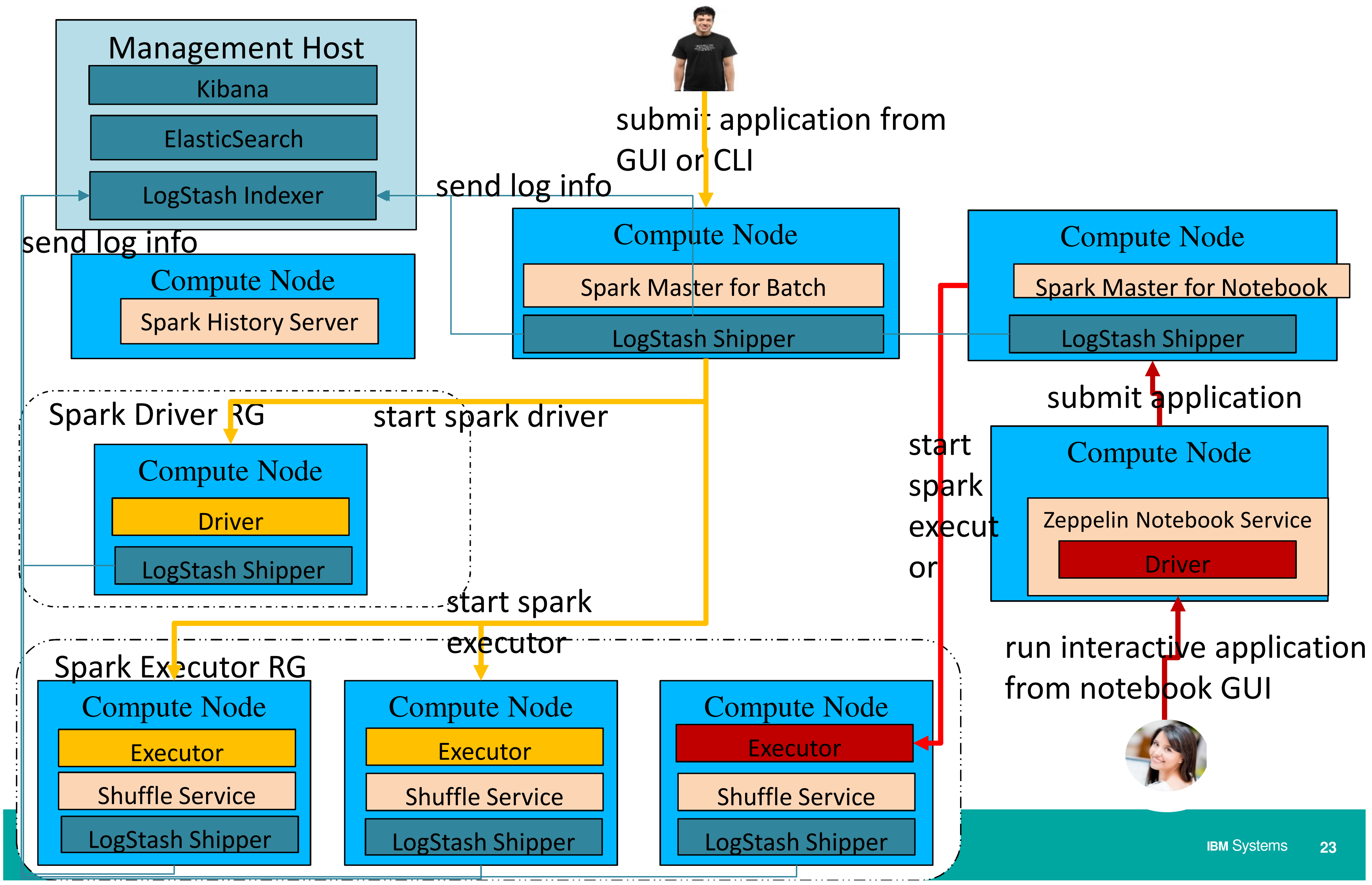

Heterogeneous Application Environment
High Performance Analytics (Low Latency Parallel) Eg Algorithmics, Murex, Calypso, Homegrown
Hadoop / Big Data Eg IBM BigInsights, Cloudera
High Performance Computing (Batch, Serial, MPI, Workflow) Eg SaS
Application Frameworks (Long Running Services) Eg MongoDB, Hortonwiorks, Yarn
High Performance Computing RDDs, SQL, Streaming Mllib, GraphX
Optimised Workload Engines for performance and time to results
Symphony
MapReduce (Symphony)
LSF
Application Service Controller
Conductor for Spark
Other Mgmt Software
Resource Management
Shared Resources - IBM Platform Computing offers Cost Effective Compute Resource Management at Scale
Data Management
Elastic Heterogeneous Storage – IBM Spectrum Scale FPO
Cost Effective Data Management at Scale
Infrastructure Management
Bare Metal Provisioning
SoftLayer APIs & Services
Virtual Machine Provisioning
IBM Platform Cluster Manager
IBM Platform Cloud
SOFTLAYER an IBM Company
IBM Cloud Manager with OpenStack
openstack CLOUD SOFTWARE
Heterogeneous Infrastructure Environment
Power
x86
Linux on z
docker
vm
On-premise, On-cloud, Hybrid Infrastructure
(heterogeneous distributed computing and storage environment)

Workload Scheduling with Session Scheduler

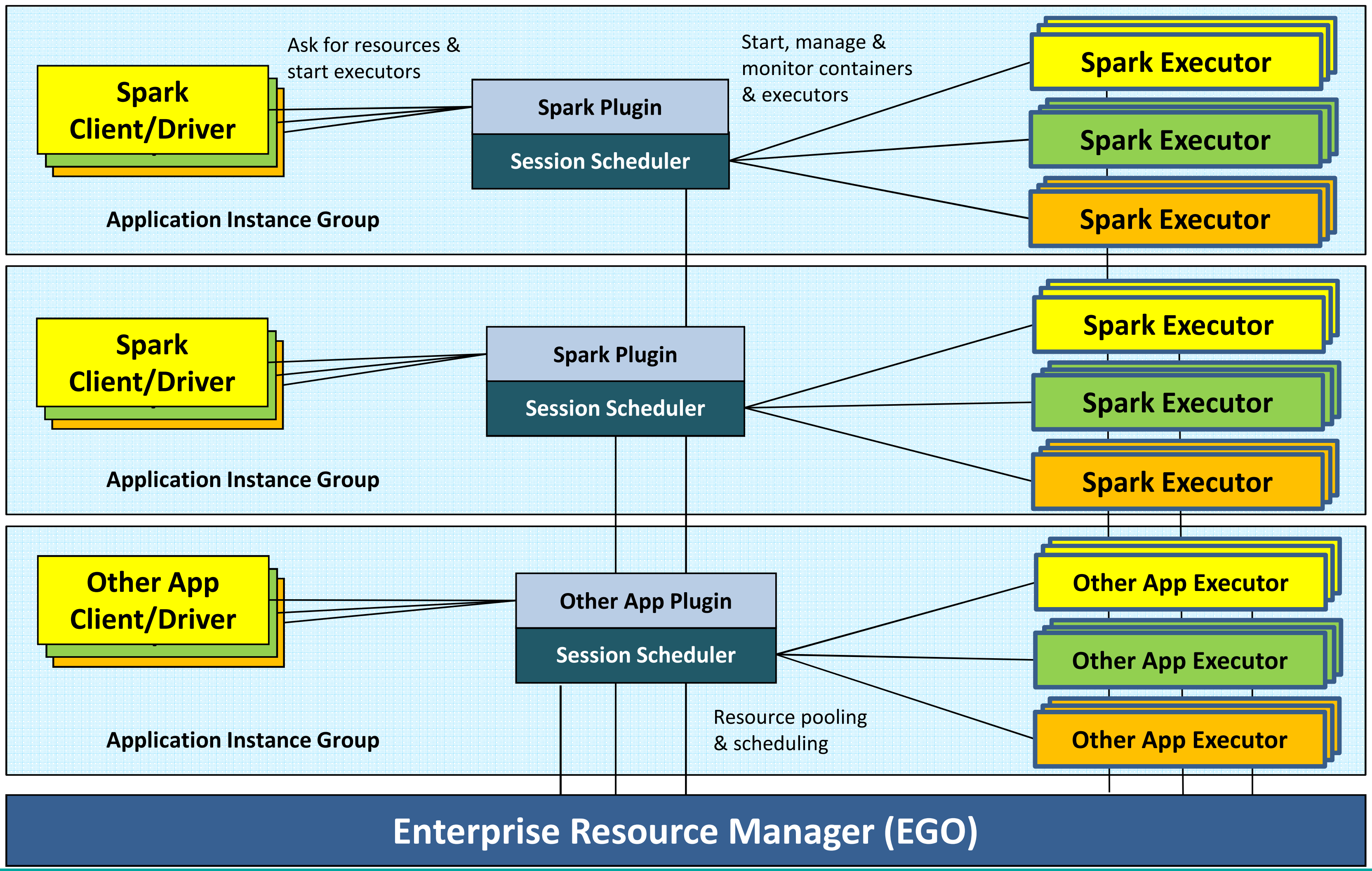
- Improved task scheduling, No changes to Apache Spark APIs
  - Multi-tenant scheduling with Session Scheduler
  - Resource sharing and reclaim
  - Fine-grained scheduling
  - Multiple resource scheduling policy (FIFO / Fairshare) for each tenant



Architecture: Runtime of a Spark Instance Group

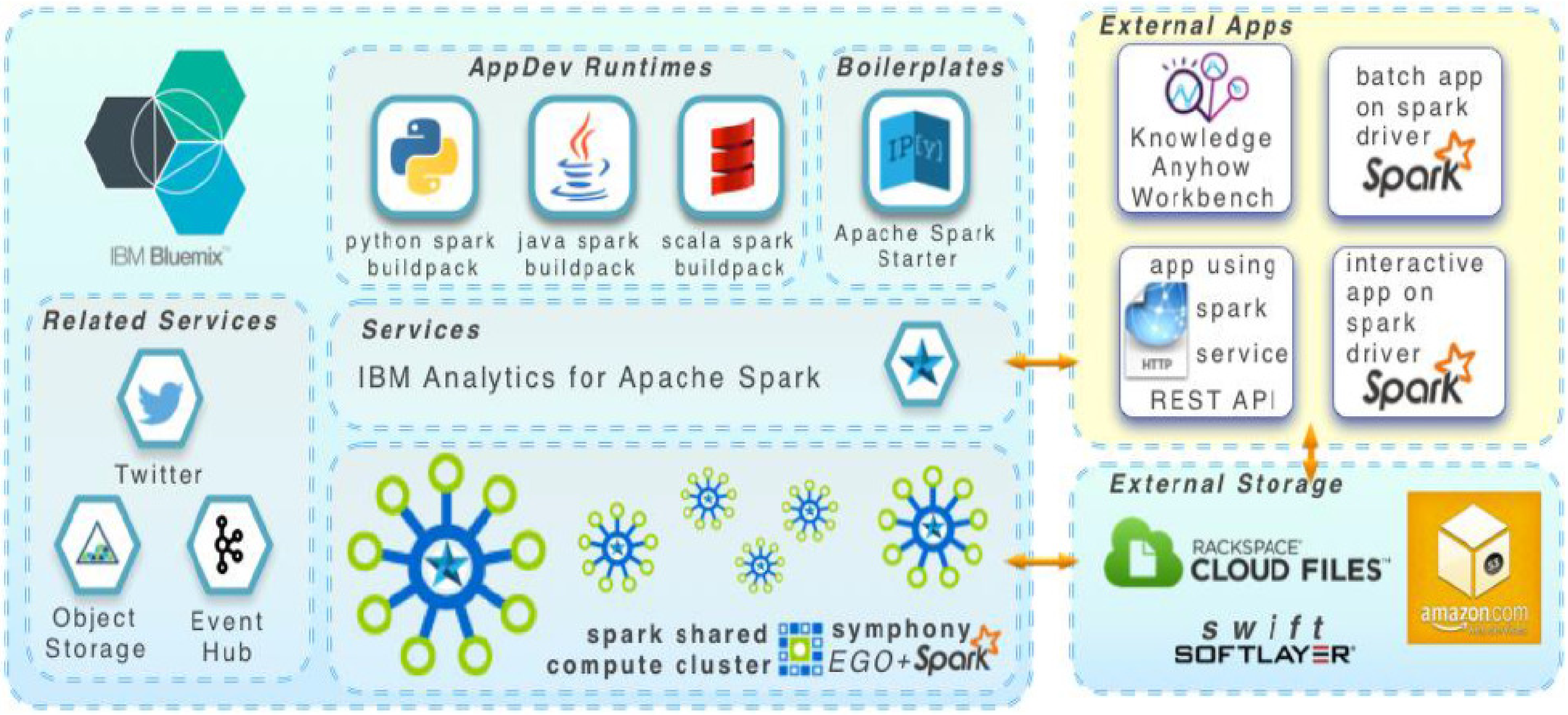


Deploy, manage and schedule multiple application instance groups

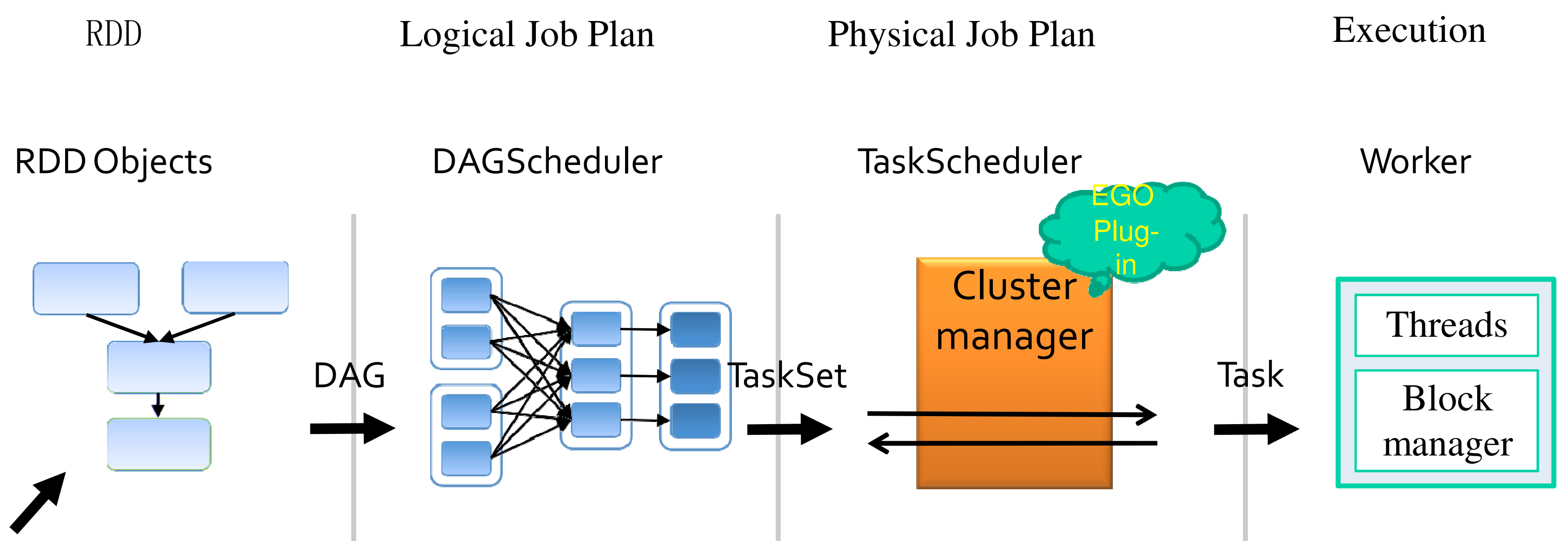


IBM Analytics Spark Cloud Service

- Multi-Tenancy
- Service Provisioning
- Fine Grain Scheduling
- Security Isolation
- High Resource Utilization
- Enterprise Class Solution



How Scheduler plug-in works with Apache Spark



```
rdd1.join(rdd2)
    .groupBy(…)
    .fi1ker(…)
```

build operator DAG

split graph into *stages* of tasks

submit each stage as ready

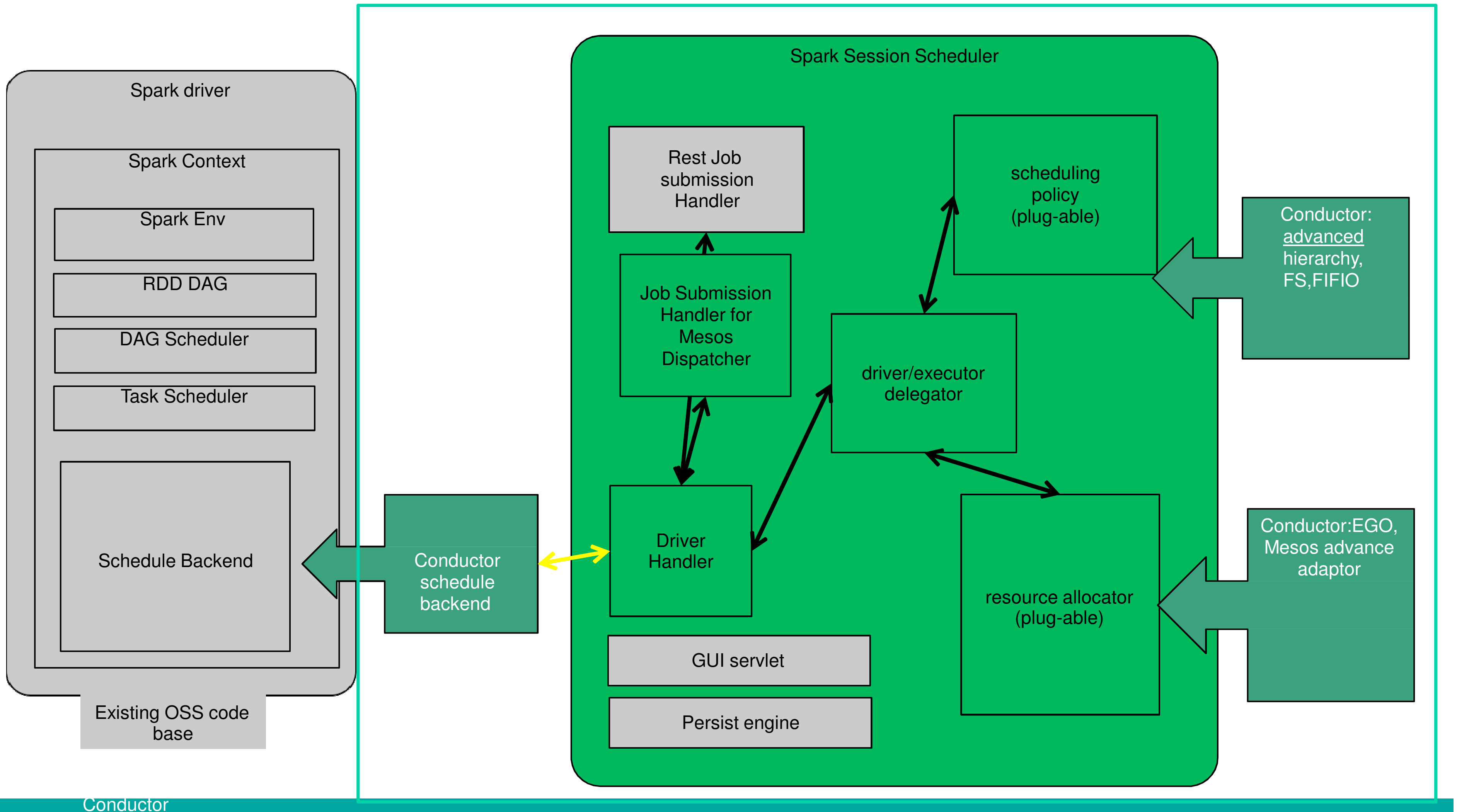
launch tasks via cluster manager

retry failed or straggling tasks

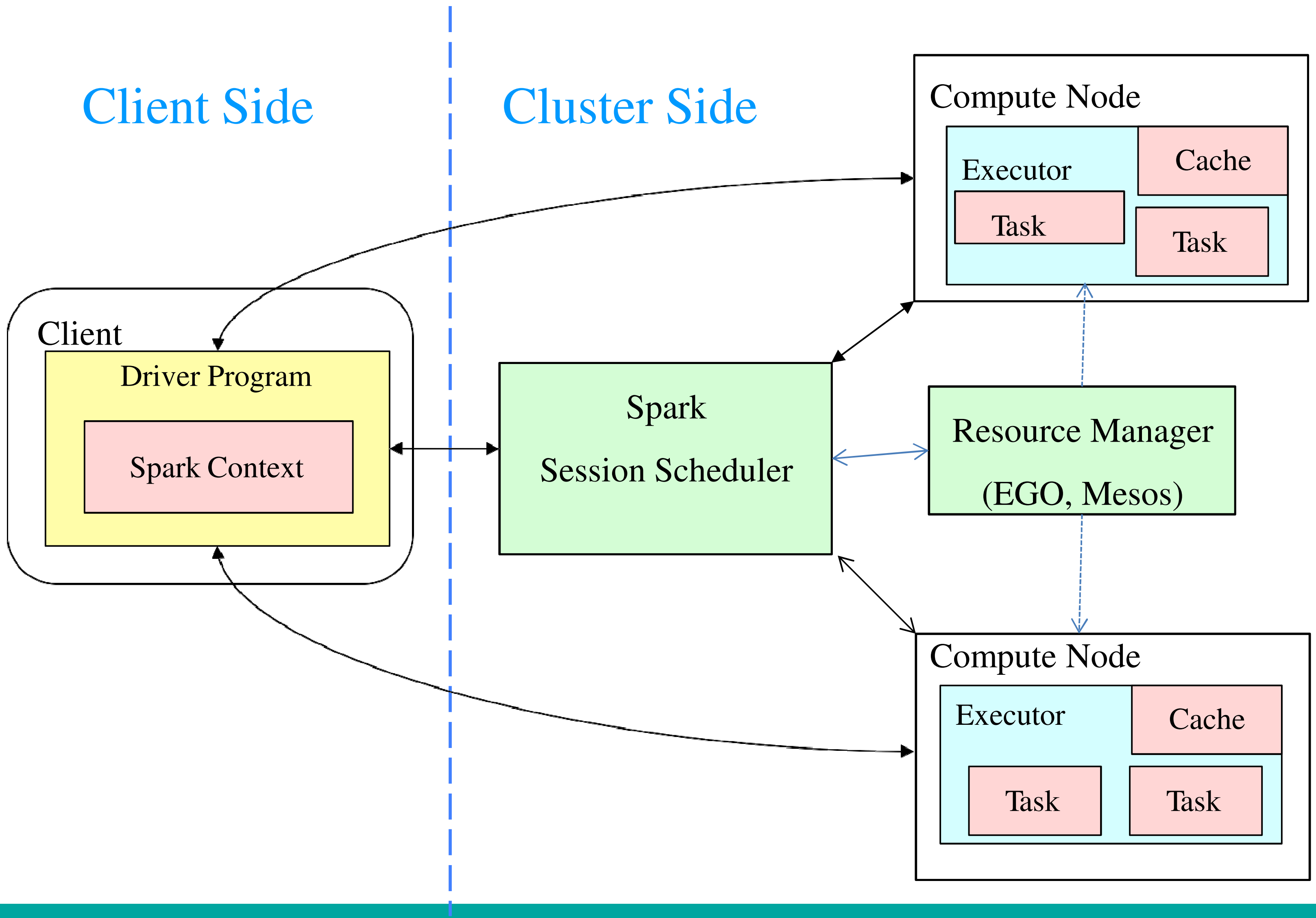
execute tasks

store and serve blocks

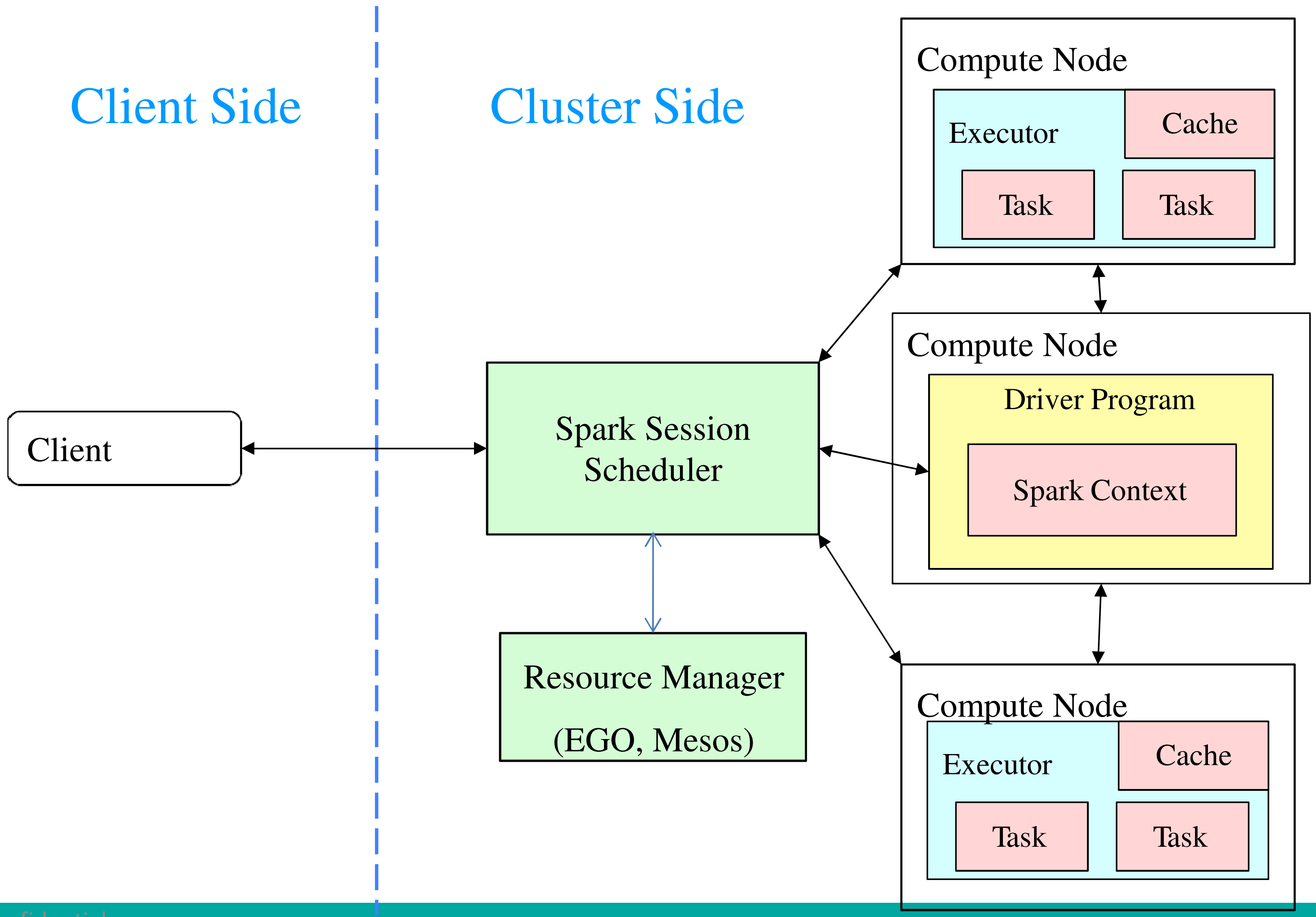
Advanced Spark Session Scheduler


Spark driver
Spark Context
Spark Env
RDD DAG
DAG Scheduler
Task Scheduler
Schedule Backend
Existing OSS code base
Conductor schedule backend
Spark Session Scheduler
Rest Job submission Handler
Job Submission Handler for Mesos Dispatcher
Driver Handler
GUI servlet
Persist engine
scheduling policy (plug-able)
driver/executor delegator
resource allocator (plug-able)
Conductor: advanced hierarchy, FS,FIFIO
Conductor:EGO, Mesos advance adaptor
Conductor
based open source protocol

Spark Scheduler - Deploy: client mode / spark-shell



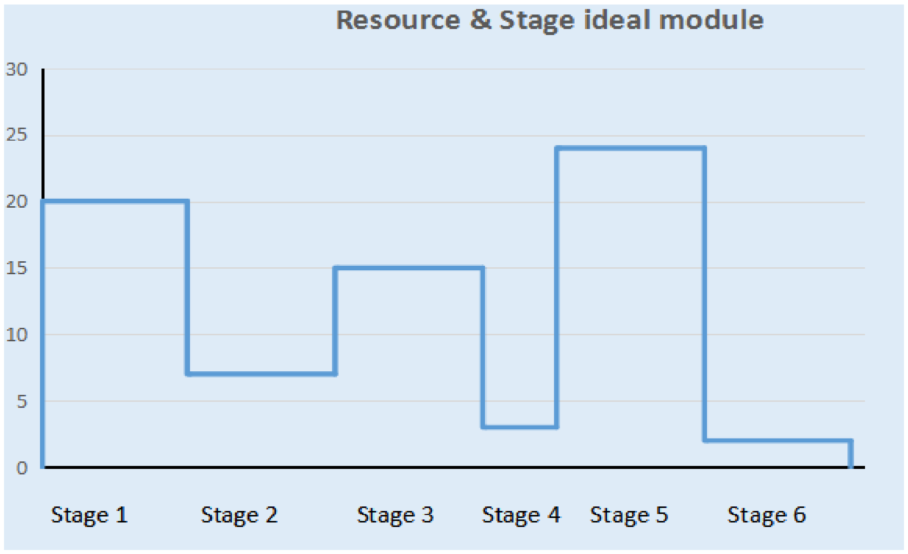
Spark Scheduler - Deploy: cluster mode

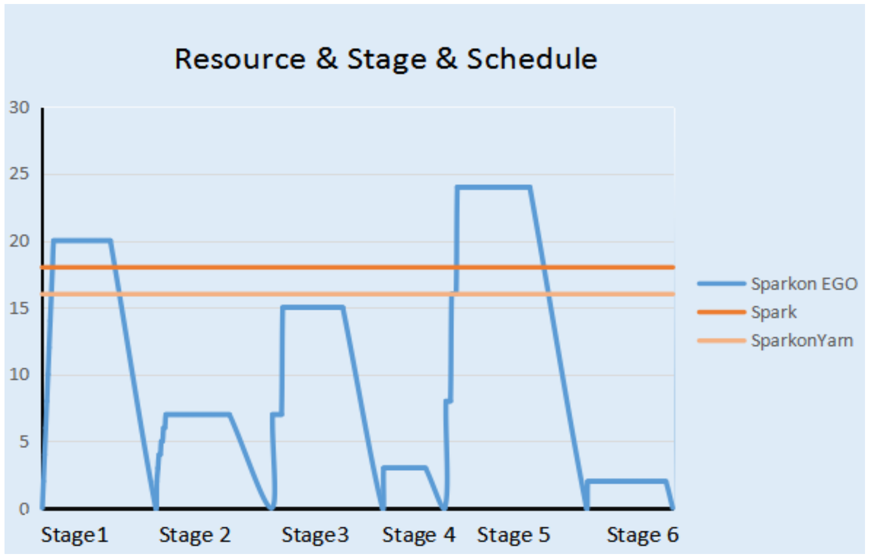


Spark Session Scheduler Advantages - Fine-grained scheduling

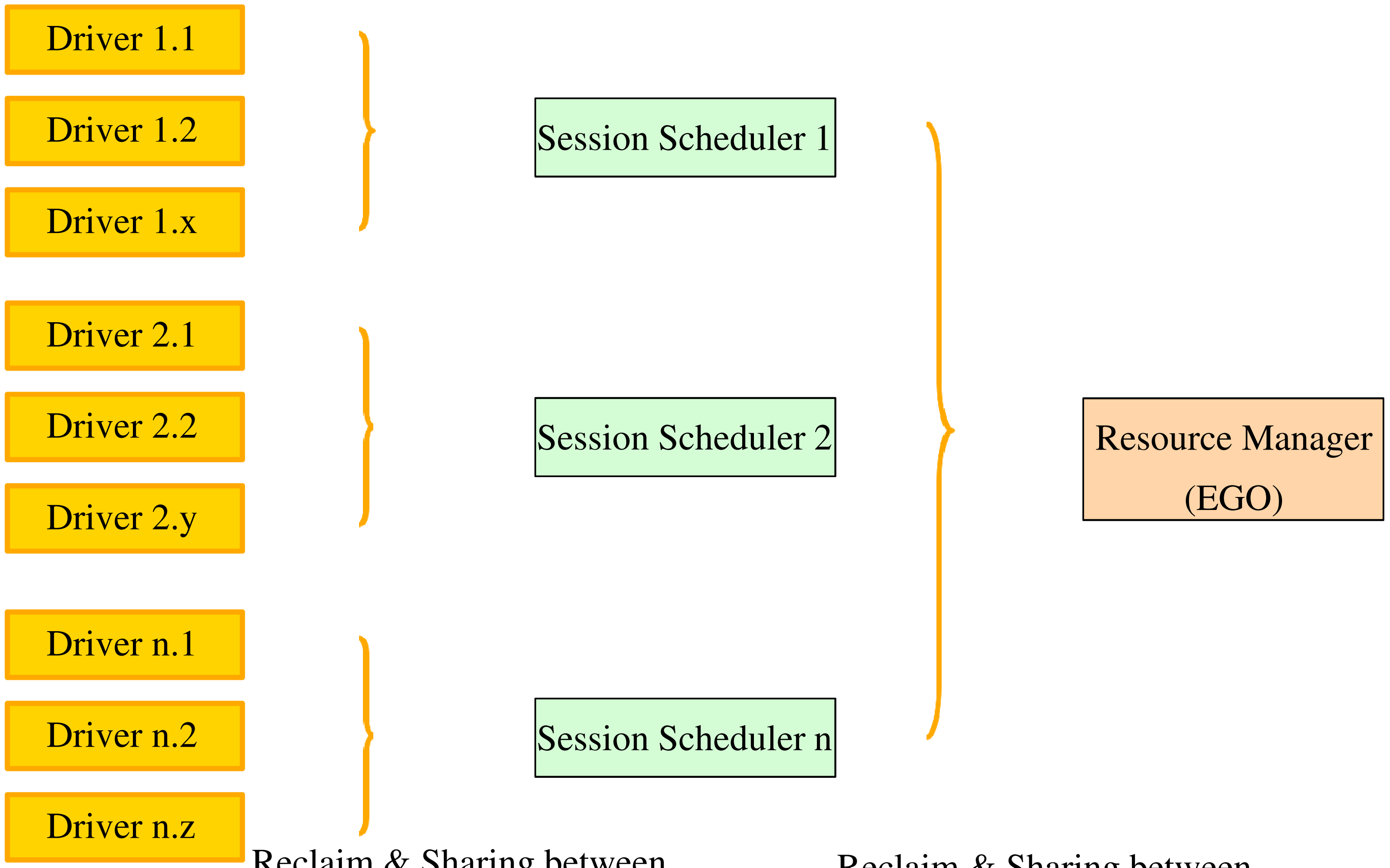
- Opensource Spark (Standalone, YARN, Mesos) assign resources by executor.
- In Conductor with Spark, resources are allocated on demand at task level.
- Fine-grained scheduling also achieves better data locality.

Stage	S1	S2	S3	S4	S5	S6
Task No	20	7	15	3	24	2





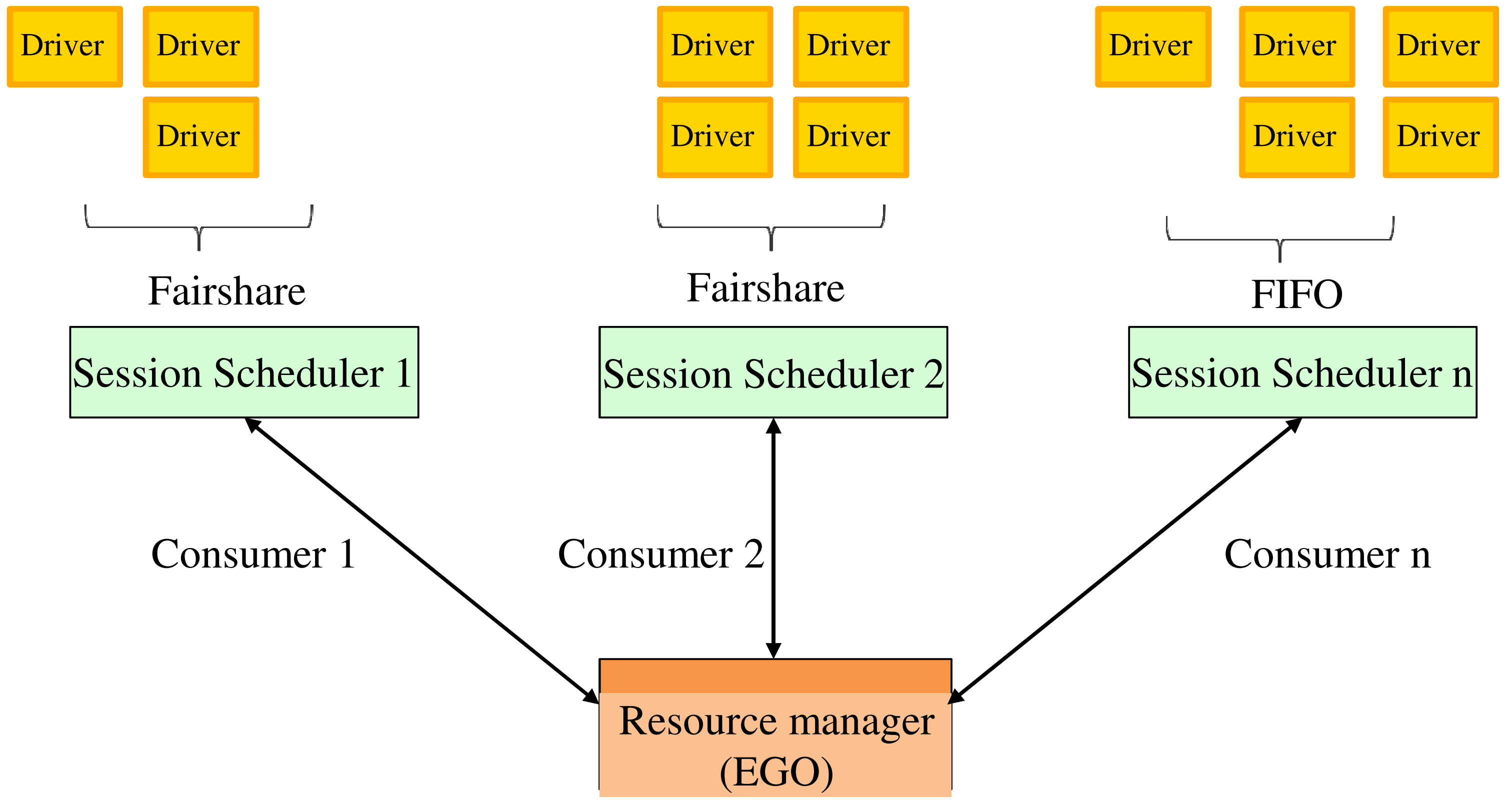
Spark on EGO Advantages - Reclaim & Sharing



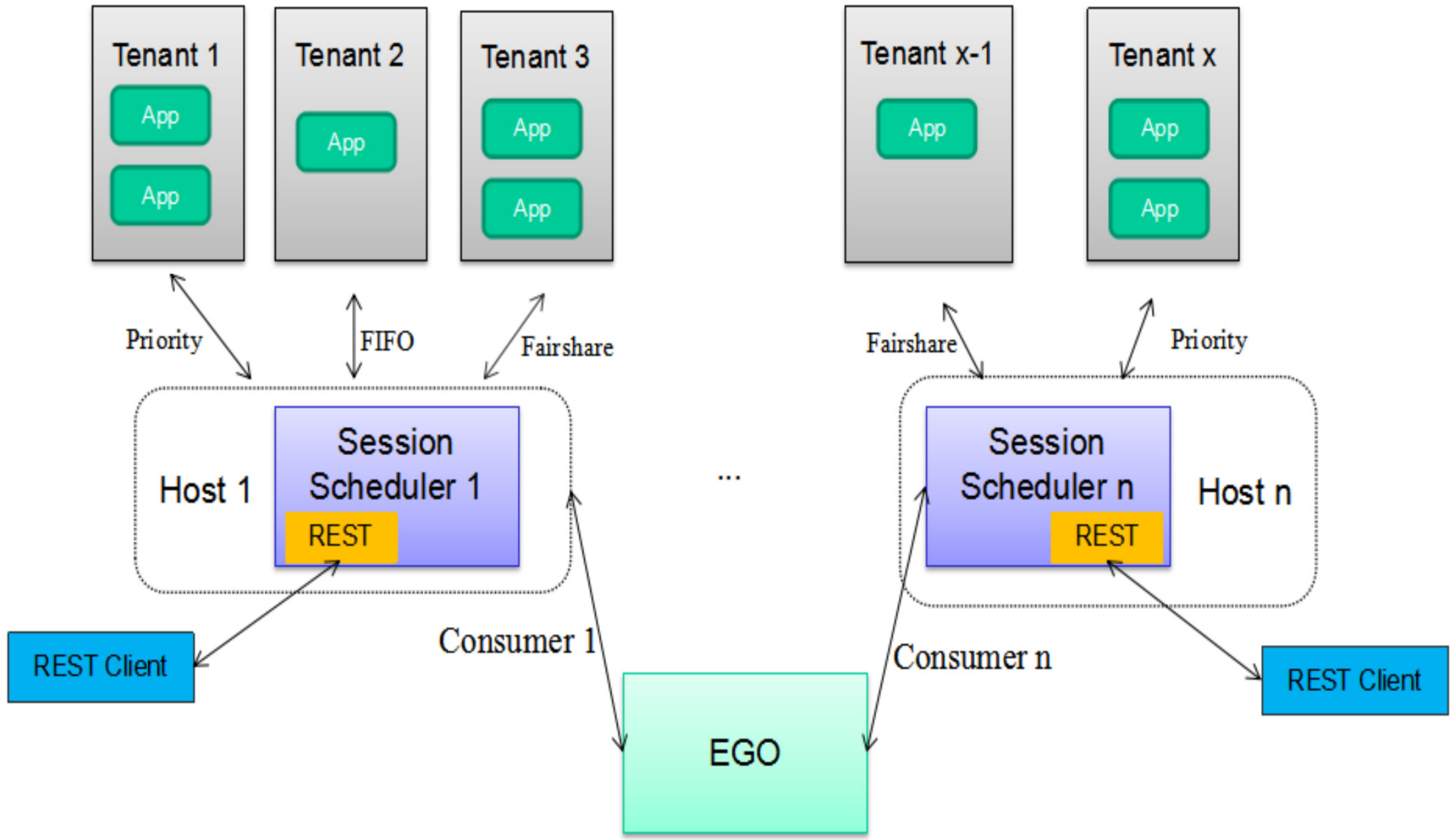
Reclaim & Sharing between Applications

Reclaim & Sharing between EGO consumers

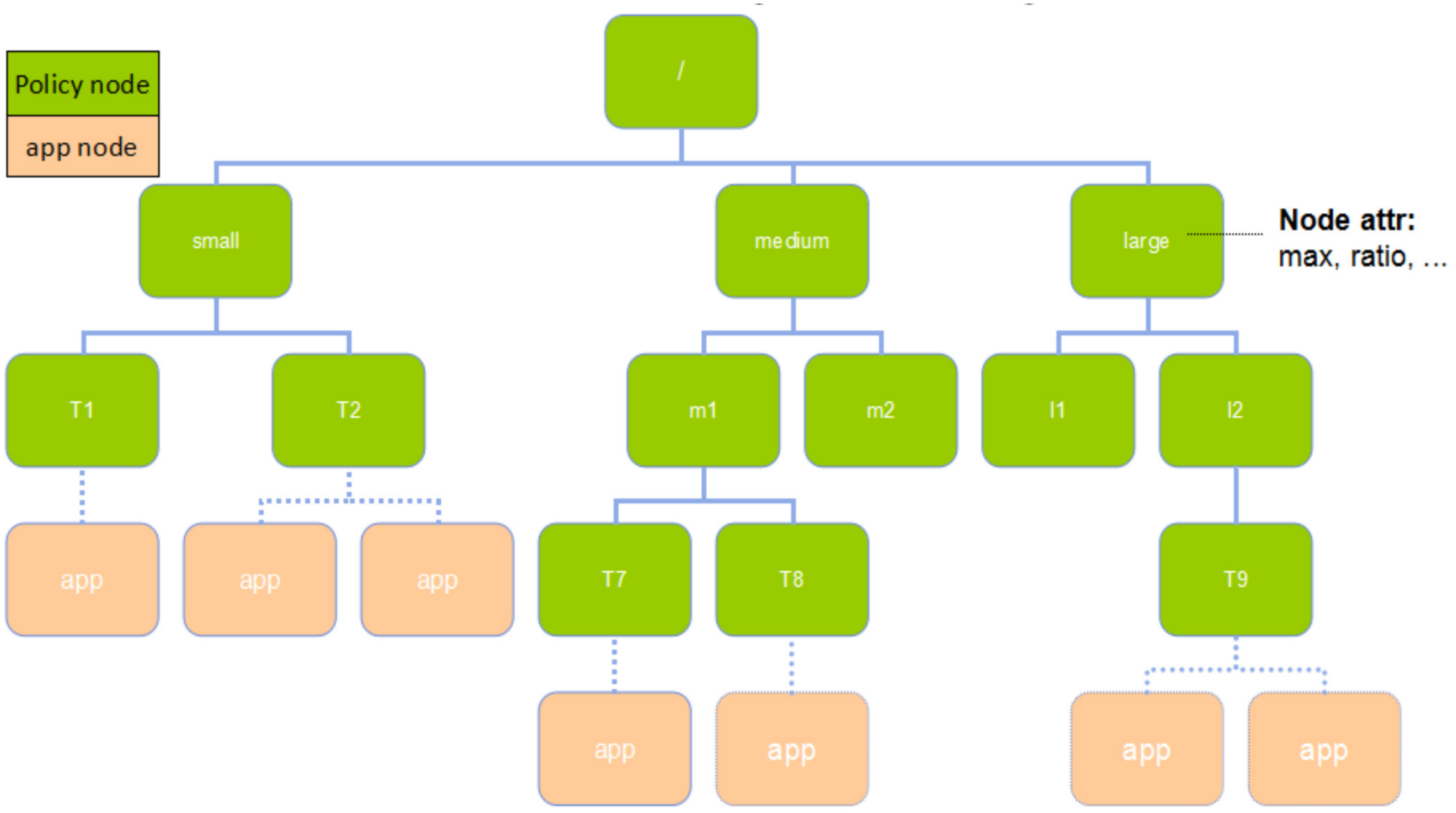
Spectrum Conductor with Spark: multi-tenant & multiple-resource schedule policy



Flexible scheduling policies (Spark service plans)



Flexible hierarchical policy in Spark Session scheduler

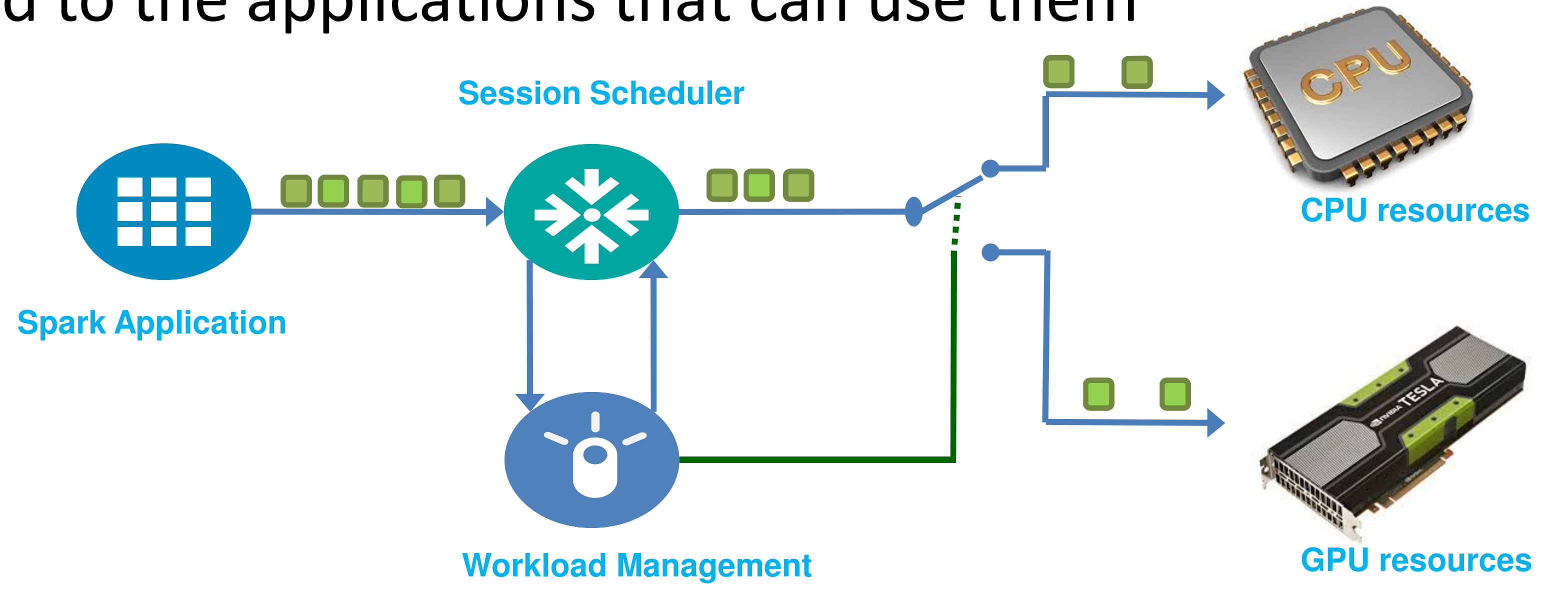


A tag is a path : /root/small/T1

Faster Time to Results – GPU Support

- Accelerate Spark applications with GPUs
  - Presented at Spark Summit San Francisco June 2016

- Conductor scheduler interfaces with Spark scheduler to ensure that GPU resources are assigned to the applications that can use them



Agenda

1	Bluemix Spark Cloud Technical Challenges
2	Bluemix Spark Cloud Architecture & Implementation
3	IBM – Spark & Mesos Community

IBM is all-in on its commitment to Spark

"It's like Spark just got blessed by the enterprise rabbi."

Ben Horowitz
Andreessen Horowitz

Contribute to the Core

Launch Spark Technology Cluster (STC), 300 engineers

Open source SystemML

Partner with databricks

Foster Community

Educate 1M+ data scientists and engineers via online courses

Sponsor AMPLab, creators and evangelists of Spark

Infuse the Portfolio

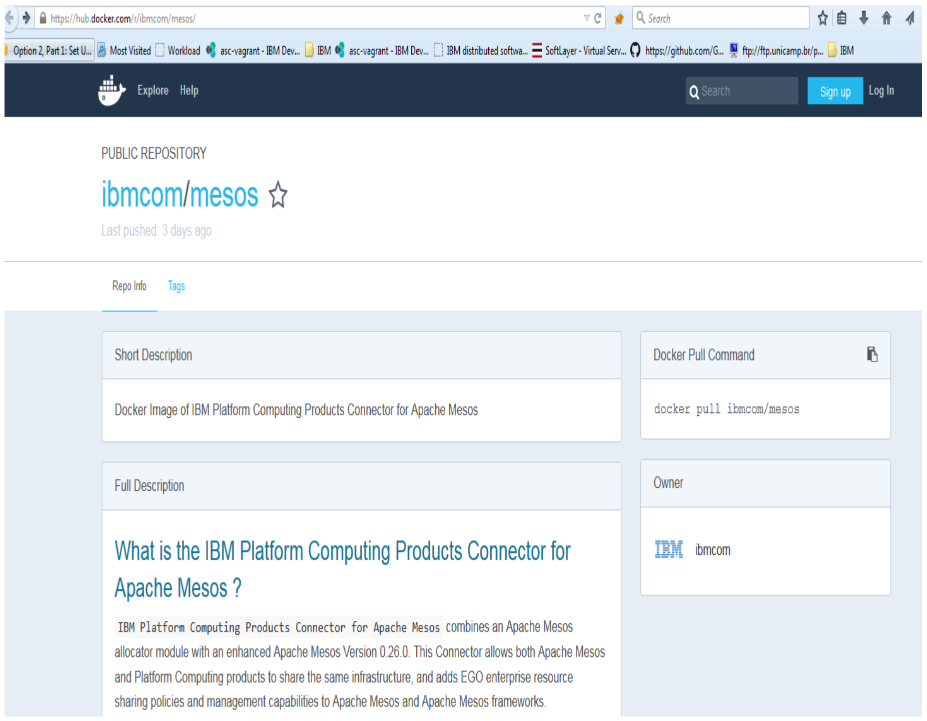
Integrate Spark throughout portfolio

3,500 employees working on Spark-related topics

Spark however customers want it – standalone, platform or products

Mesos OSS Community Activities

- Active development with Mesos community – 11 IBM Developers.
- 100+ JIRAs delivered or in progress
- Leading or participating in several work streams: POWER Support, Optimistic Offers, Container Support, GPU Support, Swarm and Kubernetes integration
- Relationship with Mesosphere – weekly calls, on-site developer presence
- Attendance at MesosCon 2016 with sponsorship and booth
- ***Technical Preview of Mesos with IBM Value-Add on Docker Hub – Both x86 and POWER images***



PUBLIC REPOSITORY

ibmcom/mesos

Last pushed: 3 days ago

Short Description

Docker Image of IBM Platform Computing Products Connector for Apache Mesos

Docker Pull Command

```
docker pull ibmcom/mesos
```

Full Description

What is the IBM Platform Computing Products Connector for Apache Mesos ?

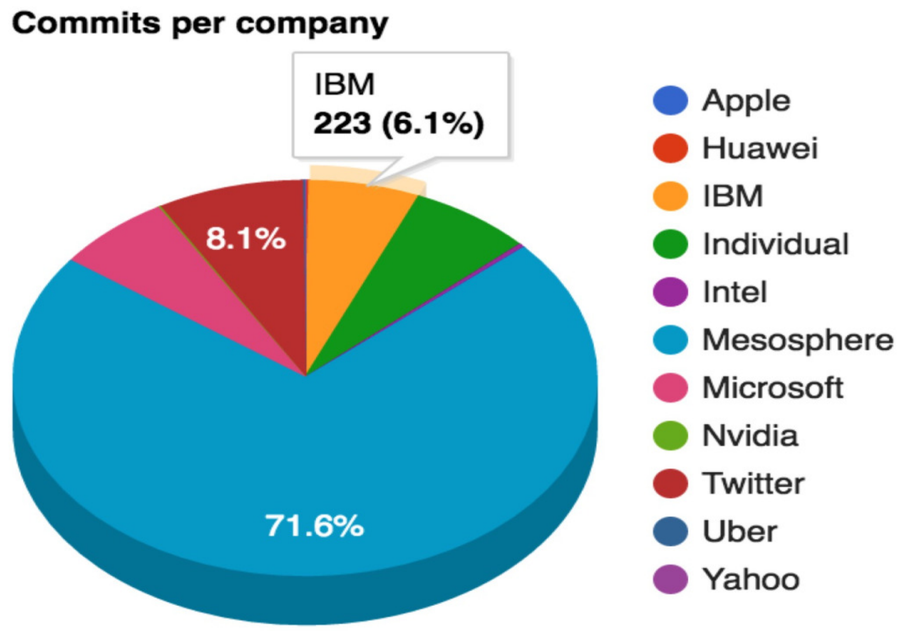
`IBM Platform Computing Products Connector for Apache Mesos` combines an Apache Mesos allocator module with an enhanced Apache Mesos Version 0.26.0. This Connector allows both Apache Mesos and Platform Computing products to share the same infrastructure, and adds EGO enterprise resource sharing policies and management capabilities to Apache Mesos and Apache Mesos frameworks.

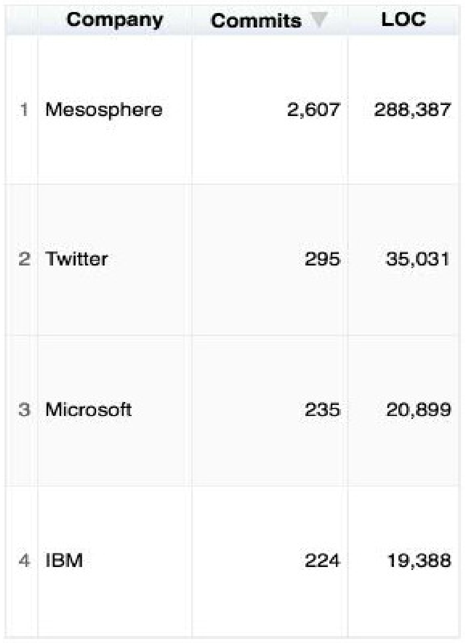
Owner

IBM ibmcom

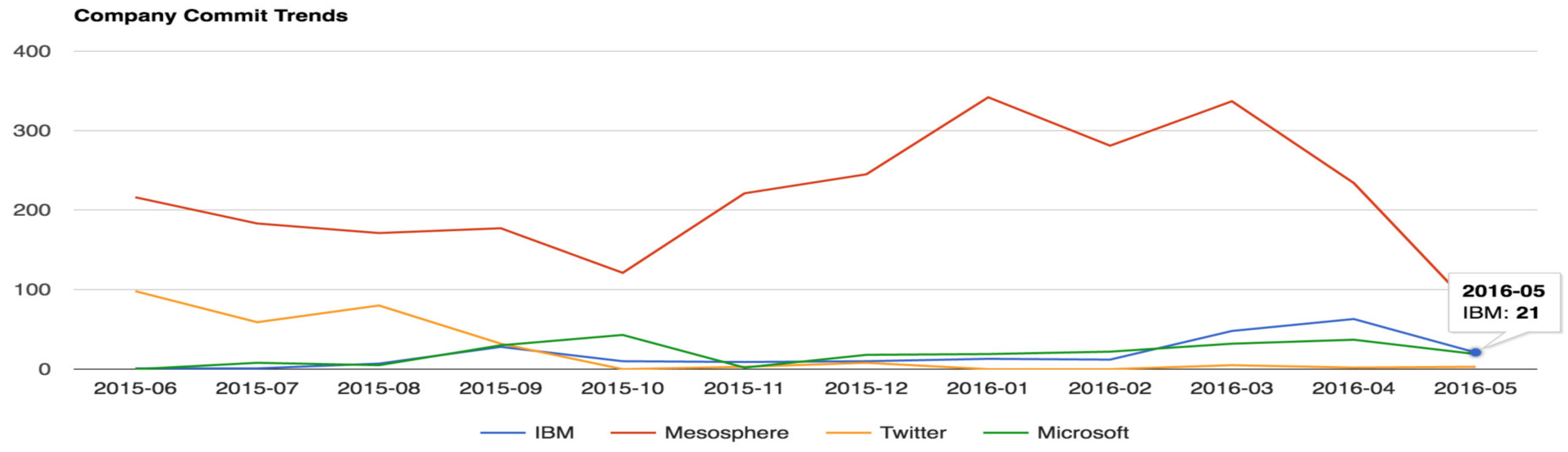
IBM Committed Mesos Patches

Note: The data is counted from Jun.2015 until now.





	Company	Commits	LOC
1	Mesosphere	2,607	288,387
2	Twitter	295	35,031
3	Microsoft	235	20,899
4	IBM	224	19,388

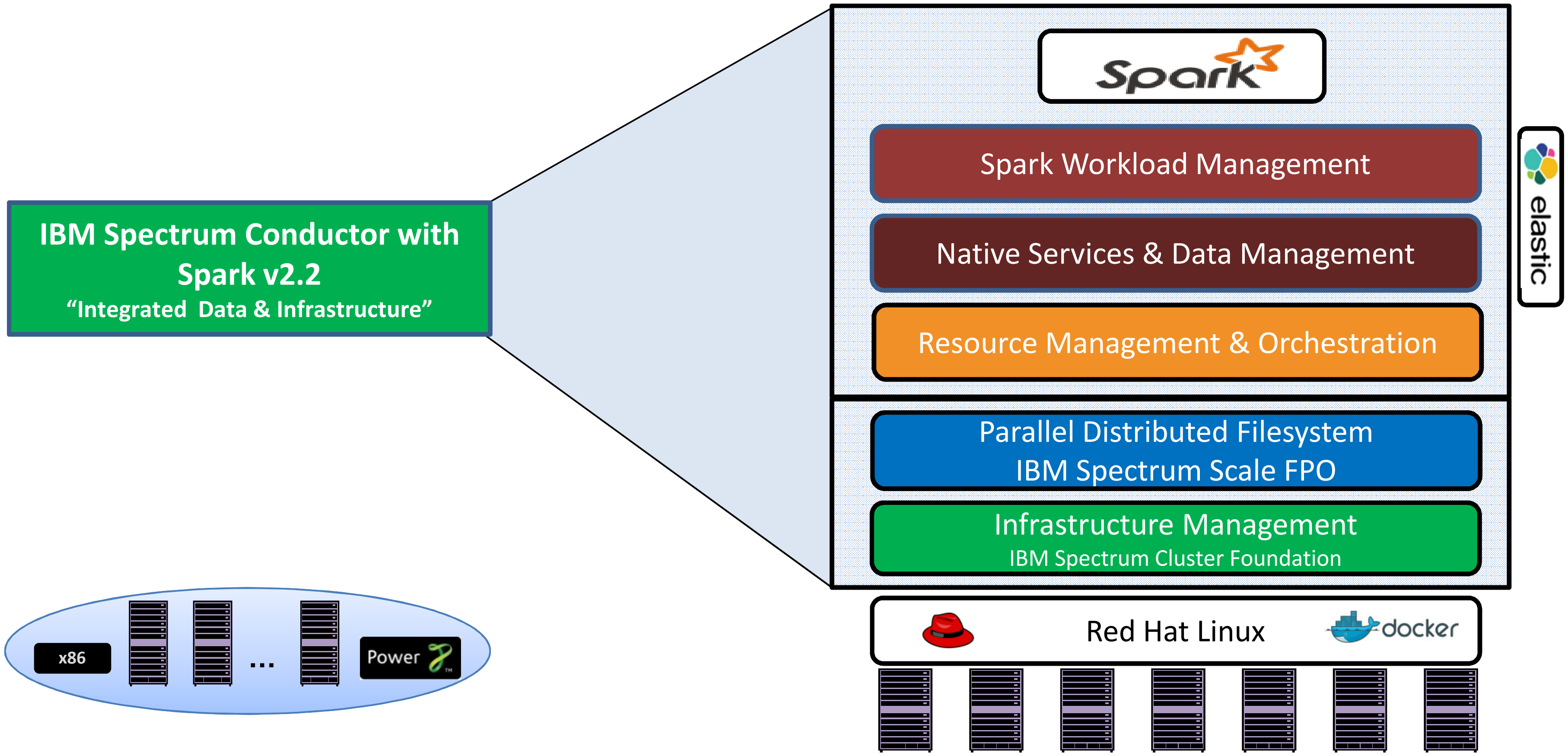


Appendix

1. **IBM Analytics for Apache Spark**
   https://console.ng.bluemix.net/catalog/services/apache-spark
2. **IBM Data Science Experience**
   http://datascience.ibm.com/
3. **IBM DataWorks (Access, combine, transform – ACT)**
   http://www.ibm.com/analytics/us/en/technology/cloud-data-services/dataworks/

Backup

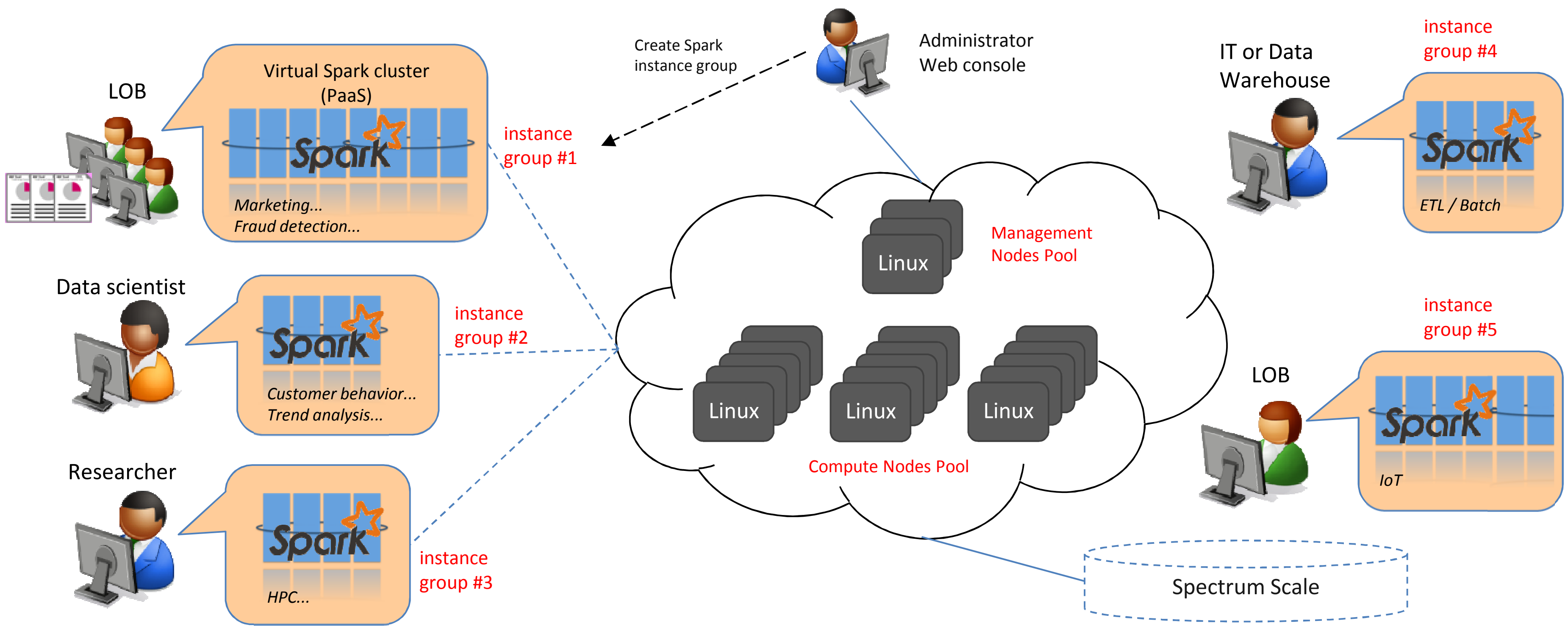
Apache Spark Cloud on-prem – Spectrum Conductor with Spark



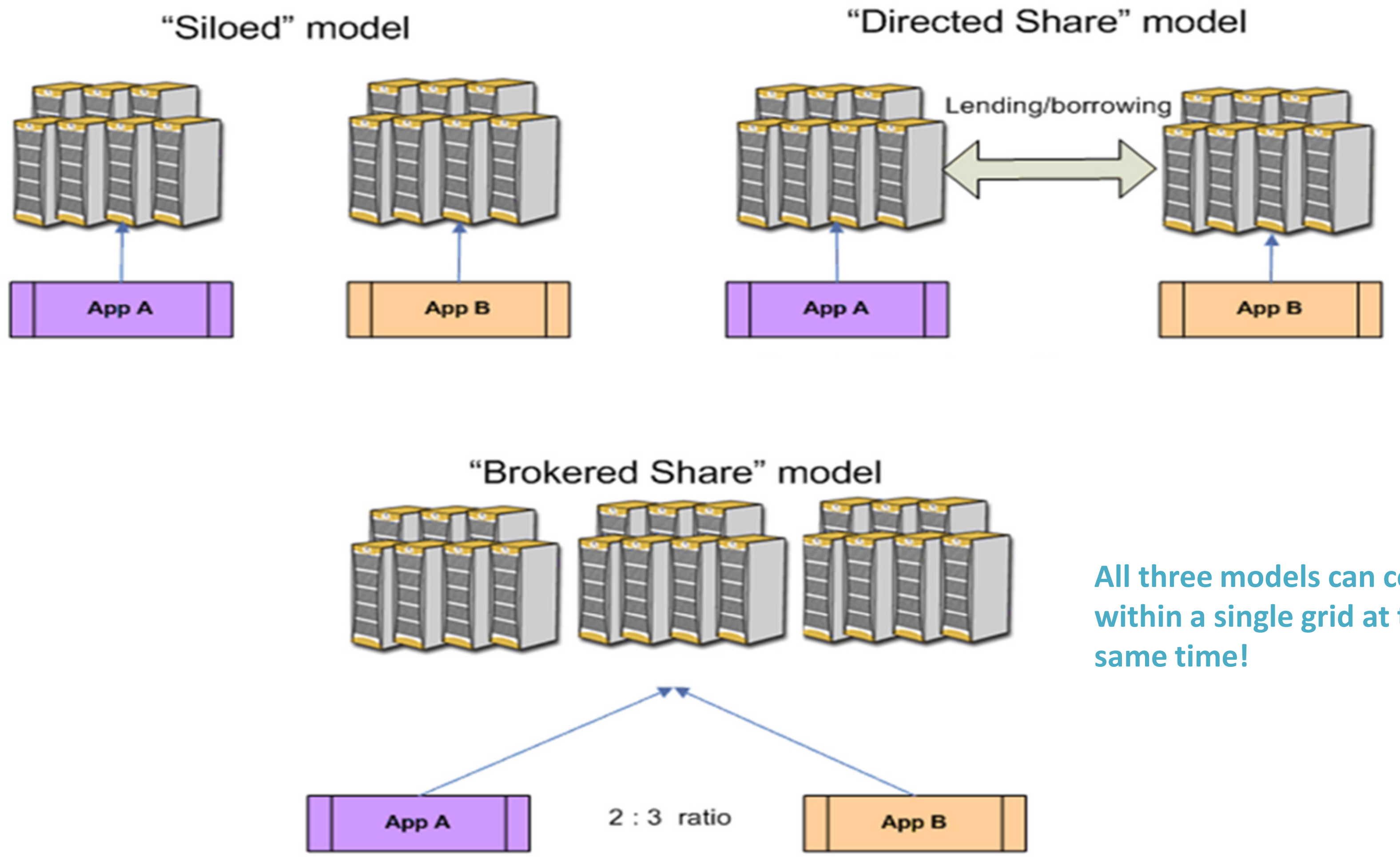
IBM Spectrum Conductor with Spark v2.2
"Integrated Data & Infrastructure"
x86
...
Power
Spark
Spark Workload Management
Native Services & Data Management
Resource Management & Orchestration
Parallel Distributed Filesystem
IBM Spectrum Scale FPO
Infrastructure Management
IBM Spectrum Cluster Foundation
Red Hat Linux
docker
elastic

Spark Shared Services Model – Private Spark Cloud

- Physical view: **Spectrum Conductor with Spark** installed on each Linux Server
- Logical view: Users (groups) have their own Spark cluster and they are isolated, protected, secured by Spark Instance Groups – Managed by SLA

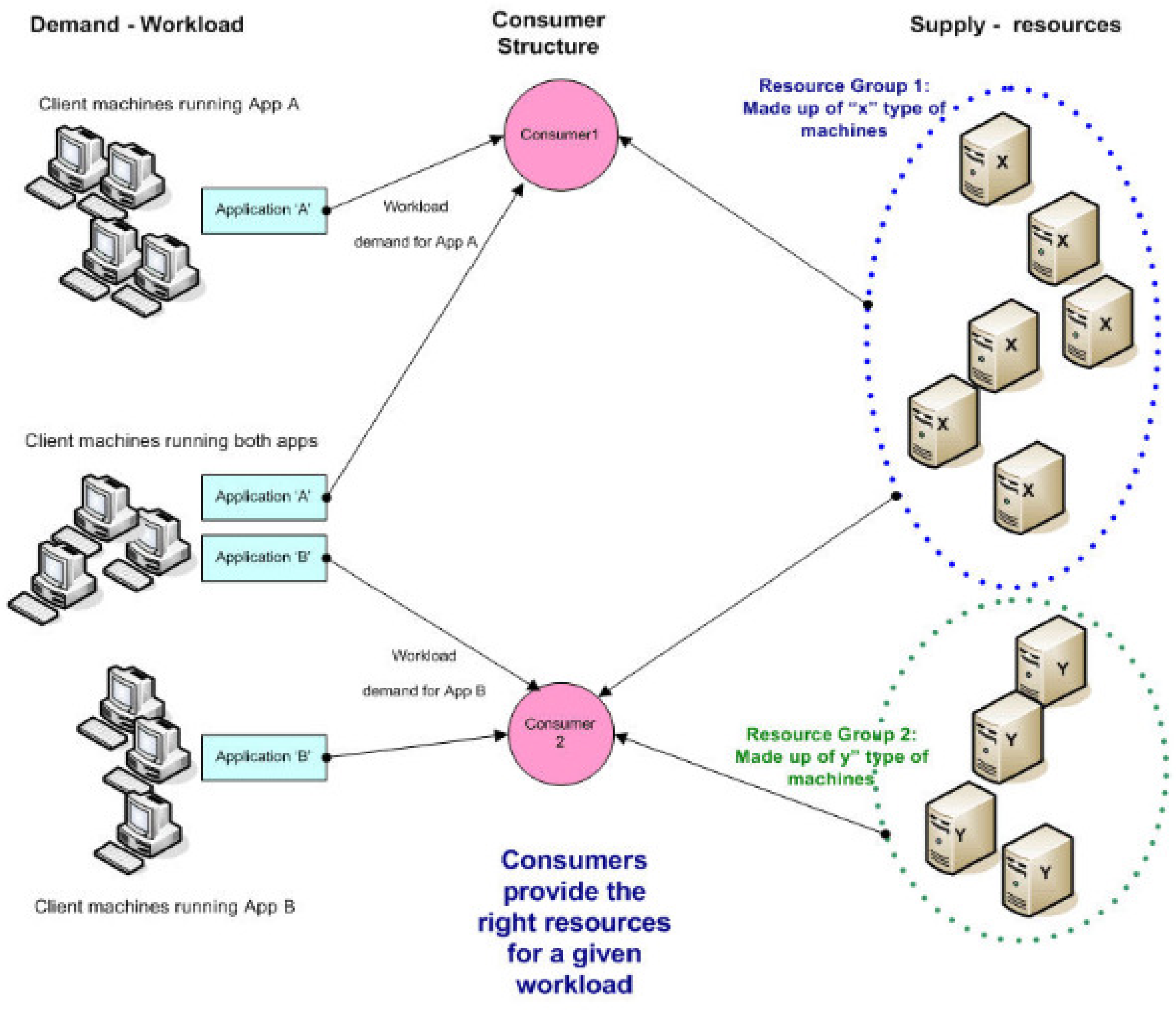


Resource Sharing Logic
Dynamic Runtime Sharing

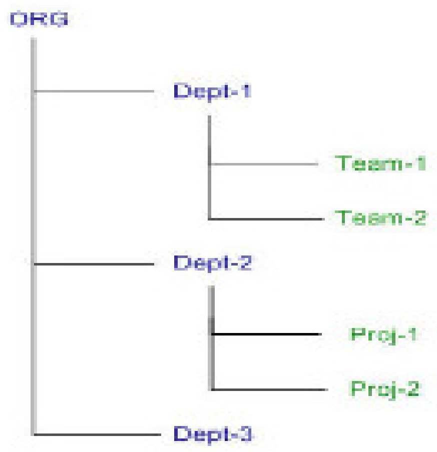


All three models can co-exist within a single grid at the same time!

Consumer Based Sharing Model


Demand - Workload
Consumer Structure
Supply - resources
Client machines running App A
Application 'A'
Workload demand for App A
Consumer1
Resource Group 1: Made up of "x" type of machines
X
Client machines running both apps
Application 'A'
Application 'B'
Workload demand for App B
Consumer 2
Resource Group 2: Made up of y" type of machines
Y
Application 'B'
Client machines running App B
Consumers provide the right resources for a given workload

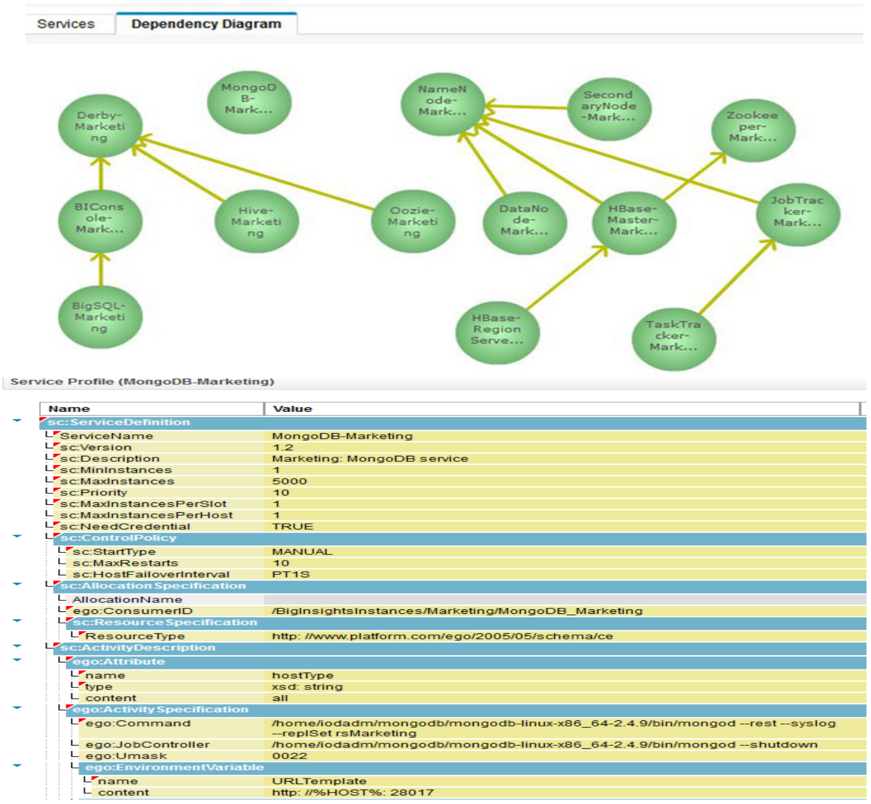
Individual User, project, department, LOB, or entire company



Consumers can be divided into lower level consumers, which may be sub divided. The lowest level consumer is the level at which the application is associated.

IBM Spectrum Conductor – Complex Service Management

Support for complex application services



Service Profile (MongoDB-Marketing)

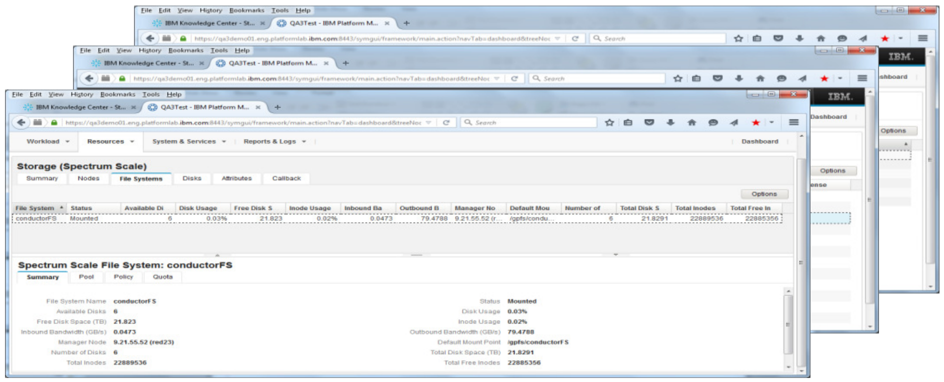
Name	Value
sc:ServiceDefinition	
ServiceName	MongoDB-Marketing
sc:Version	1.2
sc:Description	Marketing: MongoDB service
sc:MinInstances	1
sc:MaxInstances	5000
sc:Priority	10
sc:MaxInstancesPerSlot	1
sc:MaxInstancesPerHost	1
sc:NeedCredential	TRUE
sc:ControlPolicy	
sc:StartType	MANUAL
sc:MaxRestarts	10
sc:HostFailoverInterval	PT1S
sc:Allocation Specification	
AllocationName	
ego:ConsumerID	/BigInsightsInstances/Marketing/MongoDB_Marketing
sc:Resource Specification	
ResourceType	http: //www.platform.com/ego/2005/05/schema/ce
sc:ActivityDescription	
ego:Attribute	
name	hostType
type	xsd: string
content	all
ego:Activity Specification	
ego:Command	/home/iodadm/mongodb/mongodb-linux-x86_64-2.4.9/bin/mongod --rest --syslog --replSet rsMarketing
ego:JobController	/home/iodadm/mongodb/mongodb-linux-x86_64-2.4.9/bin/mongod --shutdown
ego:Umask	0022
ego:EnvironmentVariable	
name	URLTemplate
content	http: //%HOST%: 28017

A generalized service controller for complex long running application services

- Service and application definition
- Service life cycle management
- Complex service dependency
- HA, Persistency, virtual IP mgmt
- Elastic service pool
- Multiple triggers for grow/shrink
- Dynamic services deployment
- Resource sharing among long running services and tasks/jobs
- Stateful vs. stateless services
- API & scriptable interface

Spectrum Scale 4.2 Integration

- Multi-protocol support – Posix, HDFS, Object, etc.
- End-to-End, fully supported Spark solution with Spectrum Scale
    - No dependency to Hadoop / HDFS
- Benefit from all Spectrum Scale Enterprise data management features
    - Fileset based management
    - Information Lifecycle Management
    - Active file management
    - Retention and Immutability
- Integrated Spectrum Scale Monitoring



File Edit View History Bookmarks Tools Help
IBM Knowledge Center - St...
QA3Test - IBM Platform M...
https://qa3demo01.eng.platformlab.ibm.com:8443/symgui/framework/main.action?navTab=dashboard&treeNoc
Search
Workload
Resources
System & Services
Reports & Logs
Dashboard
Summary
Nodes
File Systems
Disks
Attributes
Callback
Options
Pool
Policy
Quota

Storage (Spectrum Scale)

File System	Status	Available Di	Disk Usage	Free Disk S	Inode Usage	Inbound Ba	Outbound B	Manager No	Default Mou	Number of	Total Disk S	Total Inodes	Total Free In
conductorFS	Mounted	6	0.03%	21.823	0.02%	0.0473	79.4788	9.21.55.52 (r...	/gpfs/condu...	6	21.8291	22889536	22885356

Spectrum Scale File System: conductorFS

File System Name	conductorFS	Status	Mounted
Available Disks	6	Disk Usage	0.03%
Free Disk Space (TB)	21.823	Inode Usage	0.02%
Inbound Bandwidth (GB/s)	0.0473	Outbound Bandwidth (GB/s)	79.4788
Manager Node	9.21.55.52 (red23)	Default Mount Point	/gpfs/conductorFS
Number of Disks	6	Total Disk Space (TB)	21.8291
Total Inodes	22889536	Total Free Inodes	22885356