Science Gateways Architecture, Fall 2017

Gourav Shenoy,
Software Development Engineer - IBM Watson
Cambridge, MA

Agenda

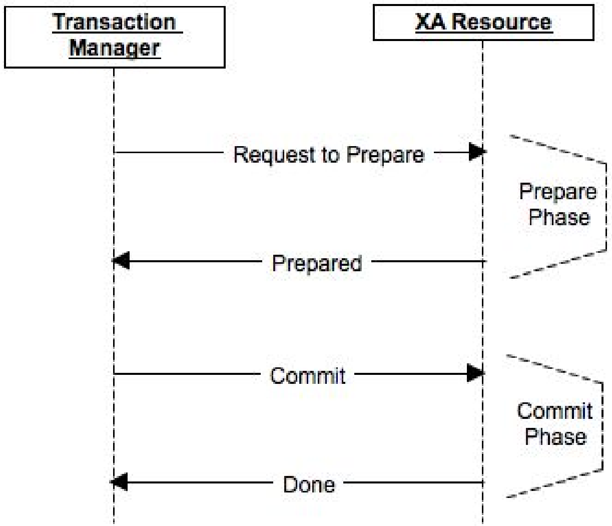
- Spring - Summer 2017
  - Event Driven Data Replication for Microservices
  - Profile Service SDKs
  - Distributed Task Execution
    - Using messaging infrastructure
    - Using Apache Helix
- IBM Watson (Health)
  - Watson for Real World Evidence (RWE)
  - Hyperledger Fabric - Blockchain Technology
- AMA

Data Management for Microservices

- Micro-service architecture - separation of concerns and independence.
- Manage data across different micro-services without compromising key attributes.
- Consider maintainability and portability of both database and micro-service.
- Database-per-microservice? Or shared database for micro-services?

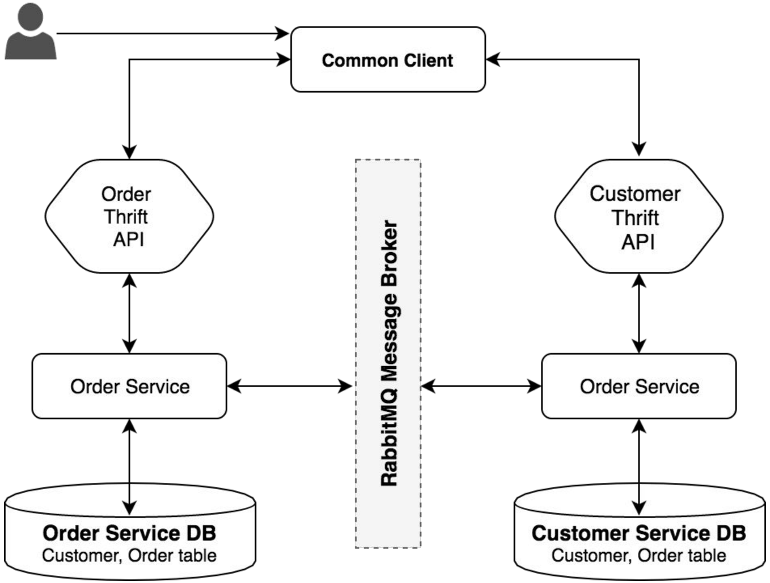
- Two solutions evaluated:
  - 2-Phase commits
  - Event driven data replication
- Evaluations based on CAP theorem.

2-Phase Commit

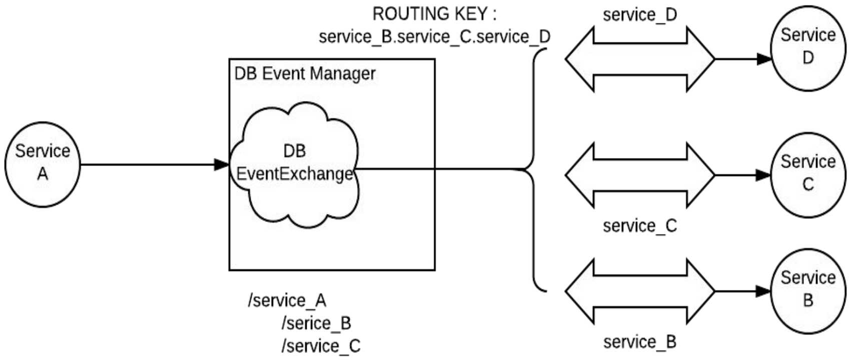


- Advantages:
  - Multiple databases - treat them as transactions.
  - Prepare phase - check if commits are allowed.
- Disadvantages:
  - Transaction manager is a single point of failure.
  - Blocking protocol - manager will block till it receives the messages from the services.
  - Rollbacks can be expensive.

Event driven data replication

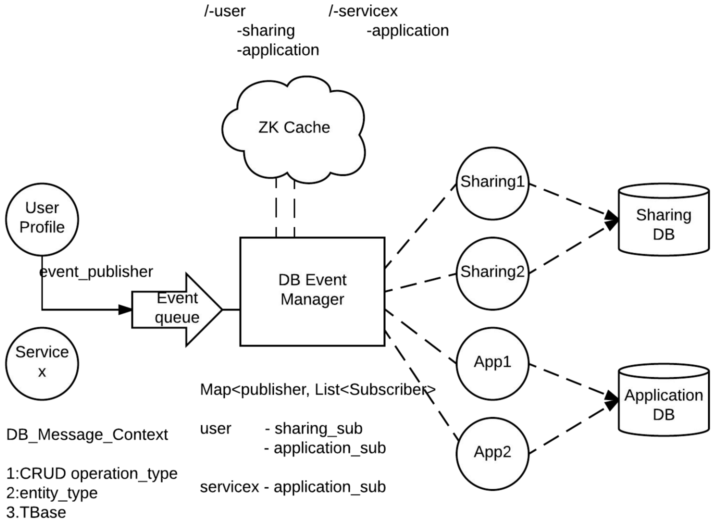


The Concept



Airavata Implementation

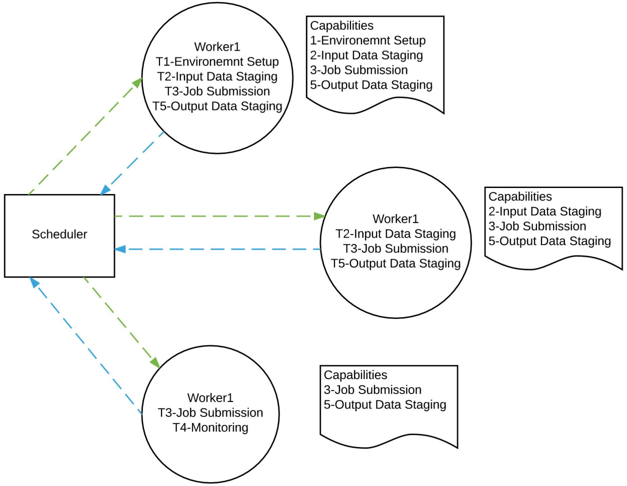
Service SDKs with Data Replication



- Break **registry** micro-service into smaller independent micro-services:
  - Profile-Service, Sharing-Registry
- Services should remain updated with the latest data.
  - **db-event-manager** module
  - Publisher-subscriber model.
  - Zookeeper for distributed coordination.

Distributed Task Execution - Push vs Pull

Push Mechanism



Scheduler

Worker1
T1-Environemnt Setup
T2-Input Data Staging
T3-Job Submission
T5-Output Data Staging

Capabilities
1-Environemnt Setup
2-Input Data Staging
3-Job Submission
5-Output Data Staging

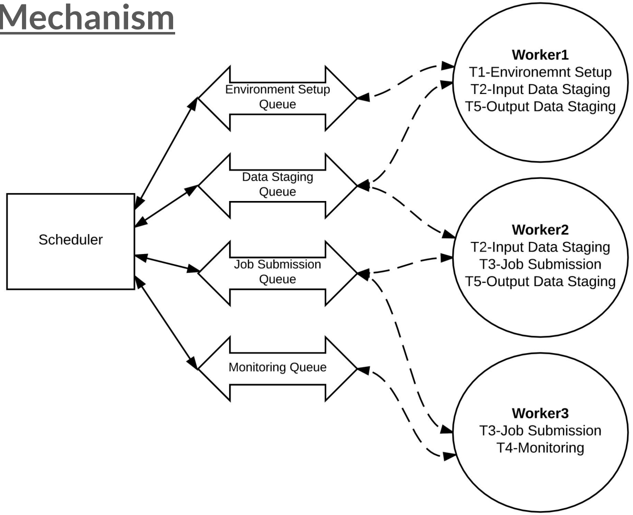
Worker1
T2-Input Data Staging
T3-Job Submission
T5-Output Data Staging

Capabilities
2-Input Data Staging
3-Job Submission
5-Output Data Staging

Worker1
T3-Job Submission
T4-Monitoring

Capabilities
3-Job Submission
5-Output Data Staging

Pull Mechanism



Scheduler

Environment Setup Queue

Data Staging Queue

Job Submission Queue

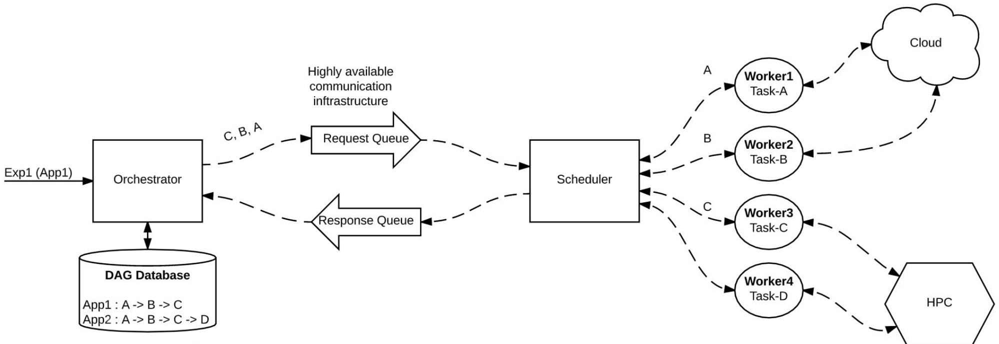
Monitoring Queue

Worker1
T1-Environemnt Setup
T2-Input Data Staging
T5-Output Data Staging

Worker2
T2-Input Data Staging
T3-Job Submission
T5-Output Data Staging

Worker3
T3-Job Submission
T4-Monitoring

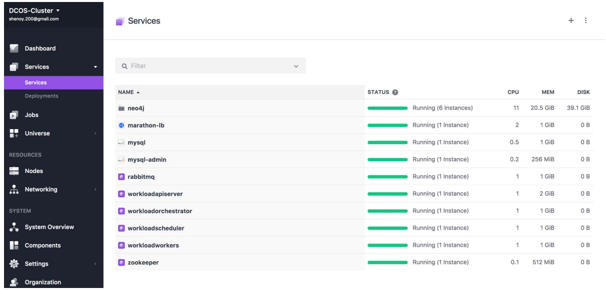
Distributed Task Execution - Messaging Infra



*** A, B, C, D are tasks*

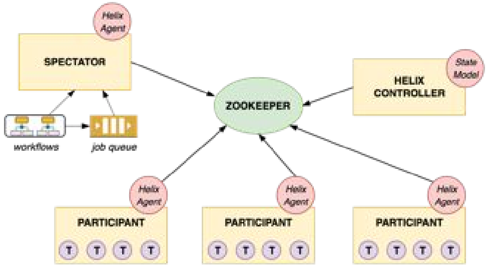
Conceptual architecture

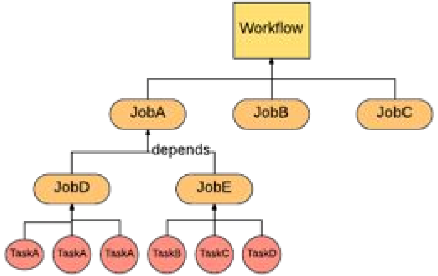
Prototype deployed on DCOS cluster



NAME	STATUS	CPU	MEM	DISK
neo4j	Running (6 Instances)	11	20.5 GiB	39.1 GiB
marathon-lb	Running (1 Instance)	2	1 GiB	0 B
mysql	Running (1 Instance)	0.5	1 GiB	0 B
mysql-admin	Running (1 Instance)	0.2	256 MiB	0 B
rabbitmq	Running (1 Instance)	1	1 GiB	0 B
workloadapiserver	Running (1 Instance)	1	2 GiB	0 B
workloadorchestrator	Running (1 Instance)	1	1 GiB	0 B
workloadscheduler	Running (1 Instance)	1	1 GiB	0 B
workloadworkers	Running (1 Instance)	1	1 GiB	0 B
zookeeper	Running (1 Instance)	0.1	512 MiB	0 B

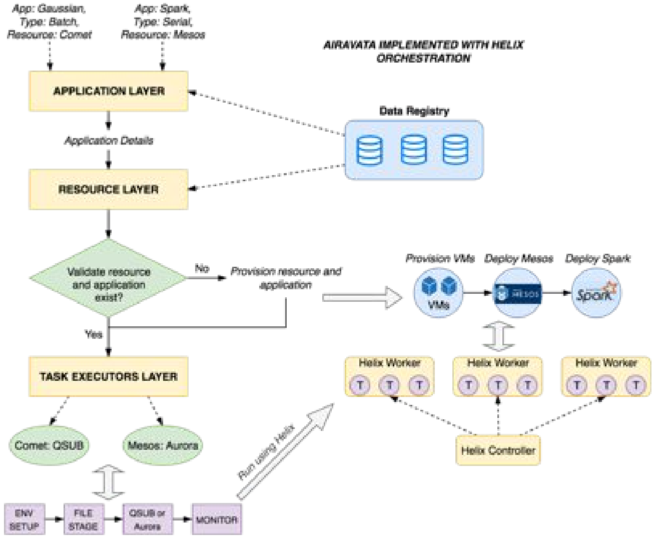
Apache Helix





- Helix is a generic cluster management framework.
- Building highly scalable and fault-tolerant distributed systems.
- Out-of-box APIs to perform Distr. Task Exe.
- Core components:
  - Participant - hosts task executors.
  - Spectator - observes participant nodes.
  - Controller - controls participant nodes, coordinates transitions in cluster.

Using Helix with Airavata



- Create Helix Tasks for each task executor (resources).
- Create Participant nodes (workers) with these resources.
- Define Online-Offline state model and appropriate state transitions.
- Define workflows (DAGs) for experiment executions.
- Define job queues for incoming experiment requests.

IBM Watson for Real World Evidence (RWE)

- Cloud-based SaaS offering for researchers, analysts, and data scientists to facilitate the discovery of relevant and valuable insights with greater efficiency and confidence.
- Using Watson Platform for Health, leverages multiple types of datasets to enable insights.
- Helps life-science companies make better, faster and evidence based decisions.
- Provides a way to centralize datasets and optimize analytics for improving the efficiency of testing hypotheses and making business decisions.

- Intended for use by analysts with programming expertise in R and/or Python.

IBM Watson RWE *(cont…)*

- The environment is preconfigured to connect to various datasets and analytic tools enabling users to focus on complex data analysis rather than spending precious time setting up inter-data connections and using disparate tooling options.
- Specific use cases of WRWE include such uses as identifying approved drugs for repurposing, comparing outcomes of competitive drugs, and mapping disease progression.
- Components of the offering:
  - Data pre-loaded and normalized (10 therapeutic datasets from IBM Explorys Life Sciences and public datasets).
  - Visual analytics to quickly find more insights.
  - Jupyter Notebooks to run queries and computations for deeper analysis of data.

My role in W-RWE

- Core developer and leading the infrastructure team.
    - Accounting and Identity Management
    - Watson health cloud platform - highly available, scalable and stable
    - Security of multiple components which are part of the offering.
- Devops
    - CI/CD - Gitlab, Jenkins, UrbanCode Deploy (and with Patterns).
    - Demo-ready system management.
- Lunch-and-Learns
    - Conduct internal sessions.
- Team bonding sessions - daily

IBM Watson Health Disruptor Hackathon

- Won first prize - Secure and Private Access to Client Datasets Using Blockchain Technology.
- Hyperledger fabric - built a prototype that makes use of smart contracts to enable clients to access their datasets on-demand from their private Watson for Real World Evidence (W-RWE) environment.
- Developed the necessary infrastructure in Watson Health cloud that automates secure connectivity from WHC to client's datacenter and securely transports encrypted datasets to their private environment in WHC.
- The smart contract guarantees that only the privileged users in client environment have access to unencrypted datasets.

Questions?