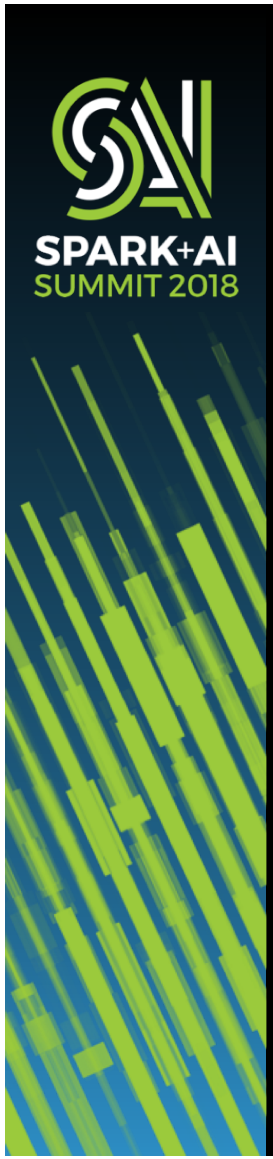


Dynamic Class-Based Spark Workload Scheduling and Resource Management Using YARN

Lakshmisha Nanjappachar and Manish Singh
Mitylytics Inc

#ExpSAIS15

Agenda

- Introduction
- Journey
- State of the art
- Capacity scheduler in Yarn resource manager
- Mitylytics solution to scheduling
- conclusion

Introduction

- Multi-tenant Spark cluster
- Job prioritization
- Ad-hoc jobs
- Lack of policy driven control
- Lack of SLA
- EMR/Dataproc/HdInsight defaults
- Lack of fine granularity controls

Journey

- Development, test and Deployment Job characterization
- Customer issues with Multi-tenanted clusters in the cloud or on-premise.
- Data collected from several large enterprise clusters
- Fine grain scheduling using feedback loop and machine learning
- Run-time resource management

State of the art

- Manual intervention
  - Kill ad-hoc jobs
  - Kill badly behaving jobs
  - Restart machines
  - Restart cluster
- Change YARN code
- Increase capacity
- Reactive solutions

Capacity Scheduler in Yarn Resource Manager

- Yarn is widely deployed including
  - AWS EMR
  - GCP DataProc
  - Azure HdInsight
- Default configuration utilizes single queue with the dominant resource specification.
- Customized scheduler requires continuous manual tuning.

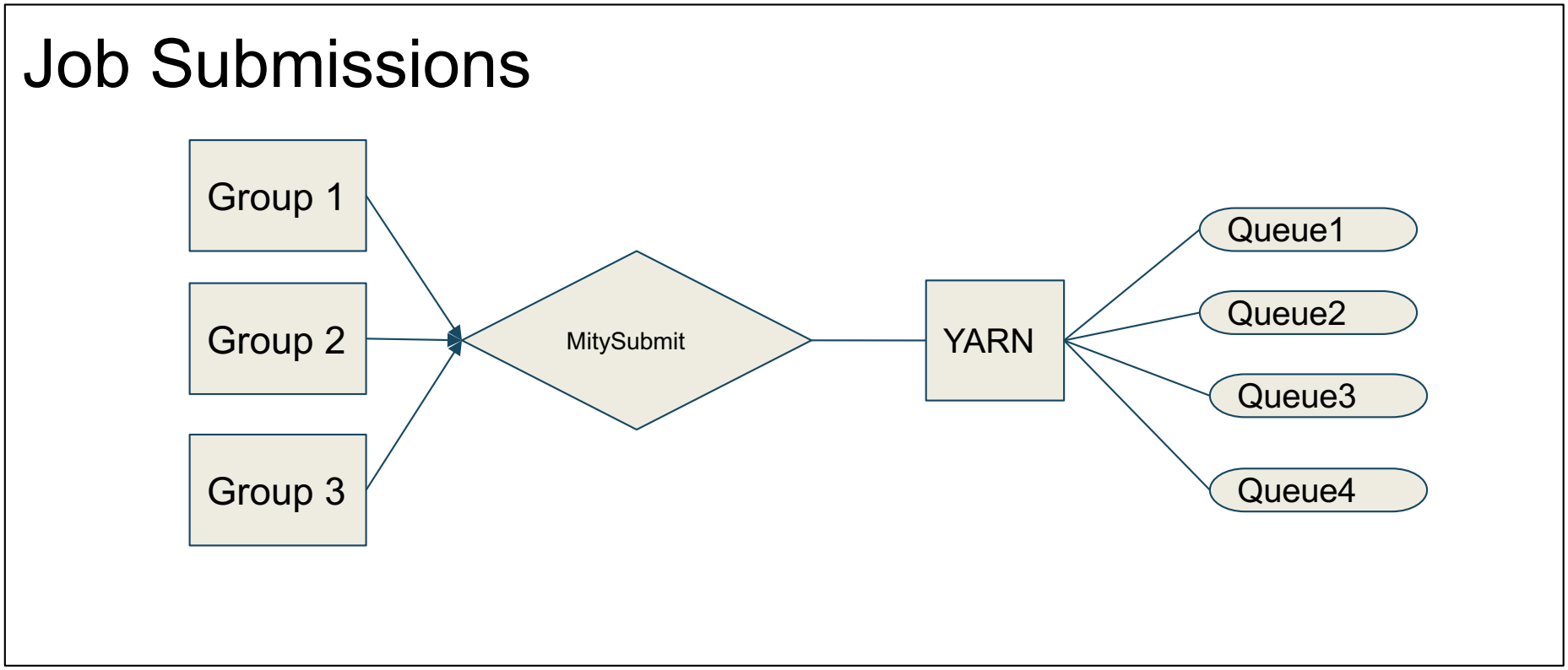
Mitylytics solution to Scheduling

- Be proactive
- Segregate jobs
- Bring dynamic resource allocation into the game
- Ensure SLAs of all jobs
- No code changes
- Drop-in solution

How does it work

- Patented Intelligent Scheduling
- Effective run-time management of resource consumption
- Patented dynamic scheduling
- All workloads welcome!
- Any workload anywhere

Workings


Job Submissions
Group 1
Group 2
Group 3
MitySubmit
YARN
Queue1
Queue2
Queue3
Queue4

MitySubmit Configuration

```
"queue_priorities":[
    {
      "queue":"gold"
    },
    {
      "queue":"silver"
    },
    {
      "queue":"bronze"
    },
    {
      "queue":"default"
    }
  ],
  "user_policy":[
    {
      "user":"manish",
      "queue":"gold"
    },
    {
      "user":"misha",
      "queue":"silver"
    },
    {
      "user":"mitylytics",
      "queue":"bronze"
    }
  ],
```

Queue & User Policy

```
"queue_priorities":[
    {
      "queue":"gold"
    },
    {
      "queue":"silver"
    },
    {
      "queue":"bronze"
    },
    {
      "queue":"default"
    }
  ],
  "user_policy":[
    {
      "user":"manish",
      "queue":"gold"
    },
    {
      "user":"misha",
      "queue":"silver"
    },
    {
      "user":"mitylytics",
      "queue":"bronze"
    }
  ],
```

Job Category Policy (example from The ML)

Demo

- Default Policy
- Job Category Policy
- Utilization
- SLA adherence

Conclusion

Intelligent and automated scheduling of YARN jobs can be done based on workload and infrastructure analysis