Data Analytics Platform

Efficient Scheduling Of High Performance Batch Computing For Analytics Workloads With Volcano

Krzysztof Adamski / Tinco Boekestijn

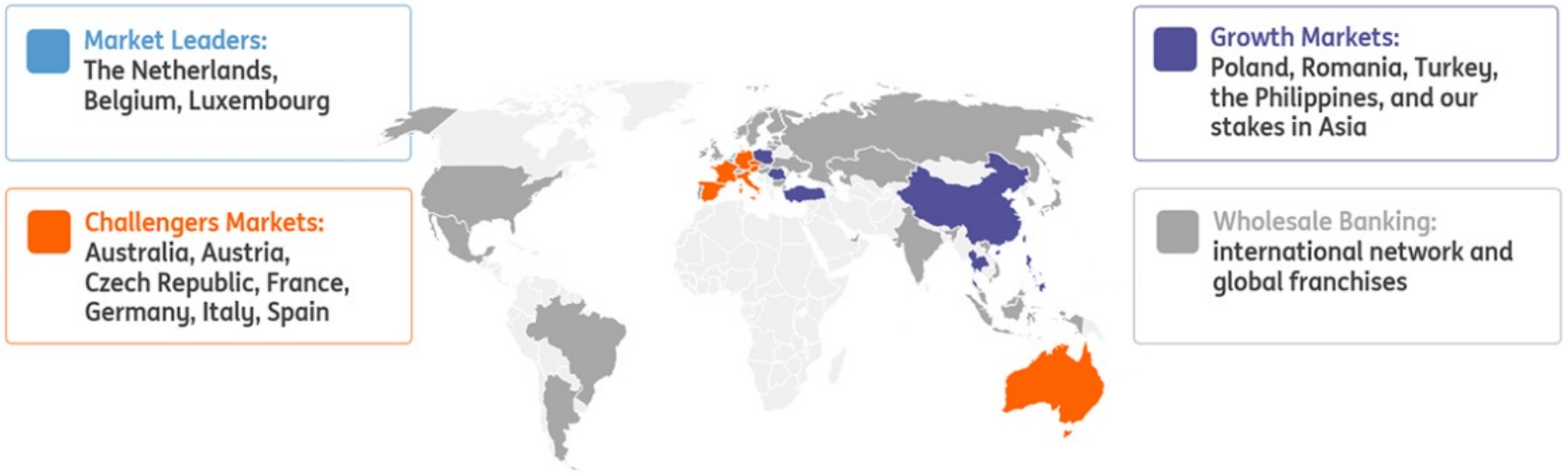
KubeCon + CloudNativeCon
North America 2022

do your thing


Our Story

ING: Who are we?

- Operational in **47 countries**
- Total Assets of **USD 1.1 trillion**
- **53.2 Million** clients globally
- **52, 000** employees

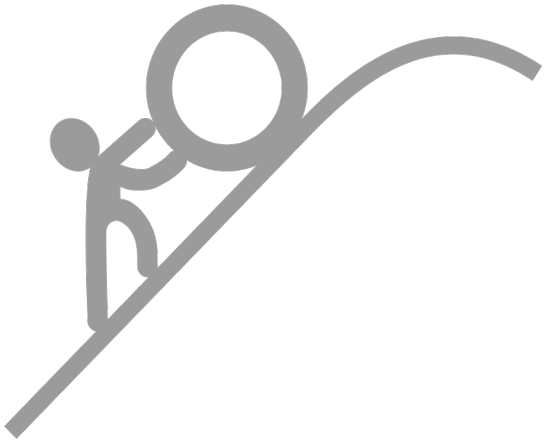


Our Purpose : To empower people to stay a step ahead in life and in business.

To facilitate and finance society's shift to a low-carbon future and pioneer innovative forms of finance to support a better world.

Challenges in the Banking industry

- Adoption of **new technology**
- **Regulatory** requirements vary globally
- **Data silos** – global and local restrictions
- Need a strong focus on **data security**
- **Compliant** innovation



North Star Metric

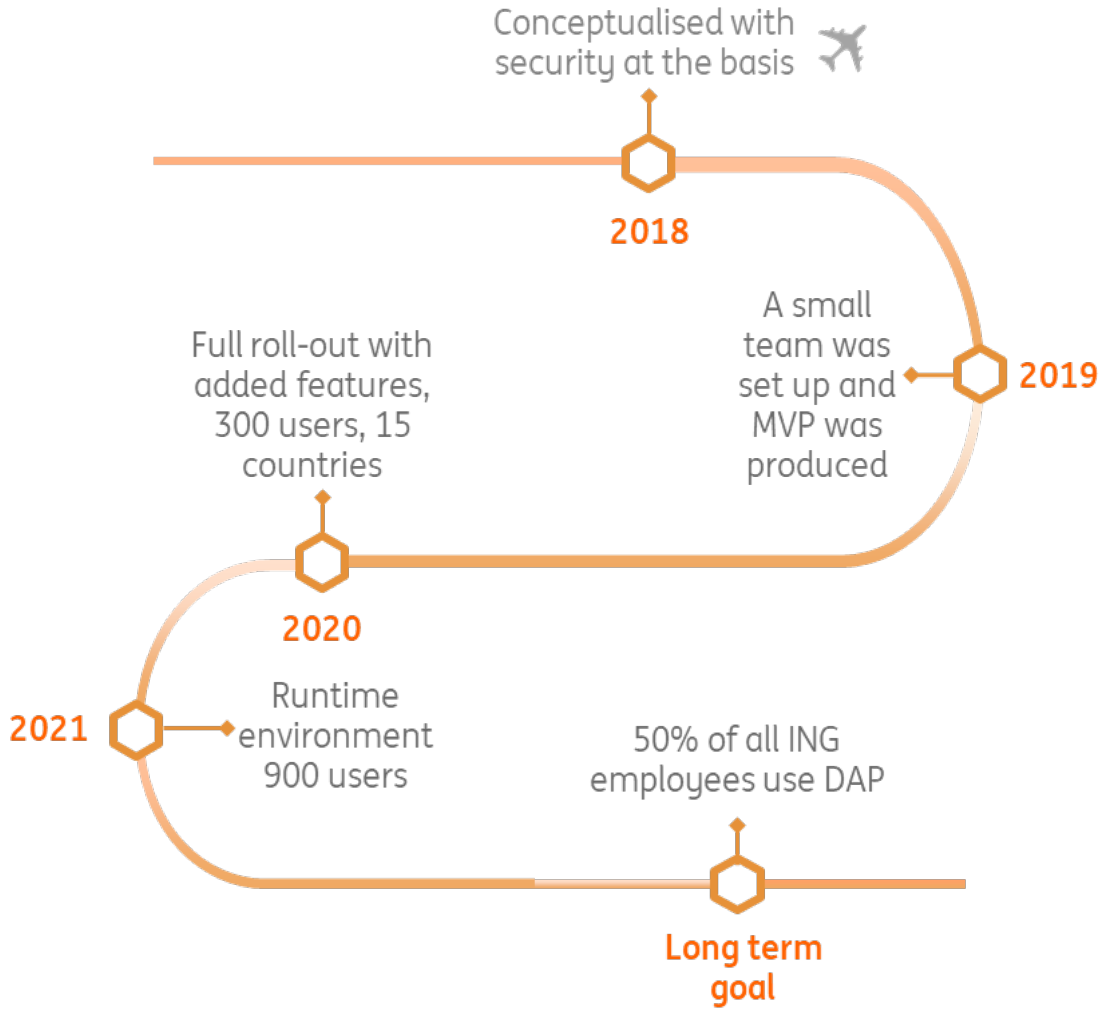
Become a data-driven company

Provide end-to-end analytics to 50% of ING employees globally in a secure, self-service platform

DAP : Our journey

- **1100** users, **17** countries
- Total no of projects running on DAP: **453**
- Average daily users: **160**
- No. of total hours monthly on DAP:

  ~10,000 hours
- MoM avg user growth : **8.1 %**

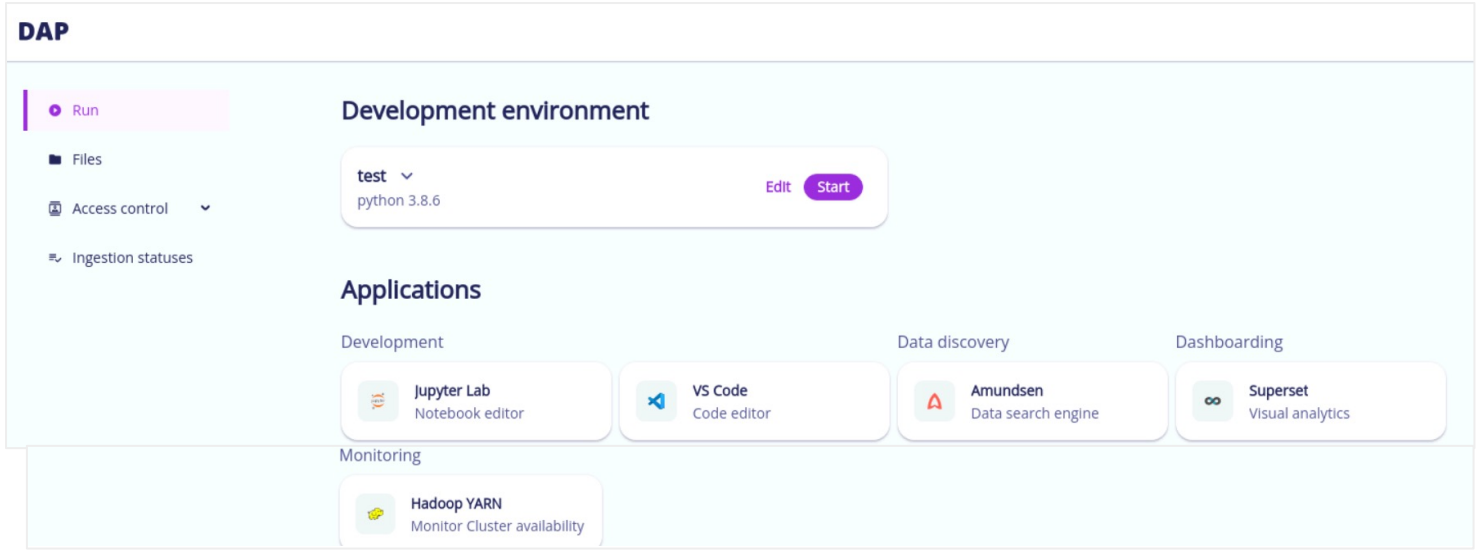



The Product

DAP : The Product

Data Analytics Platform aims to address all analytics needs in a single, highly secure, **self-service** platform.

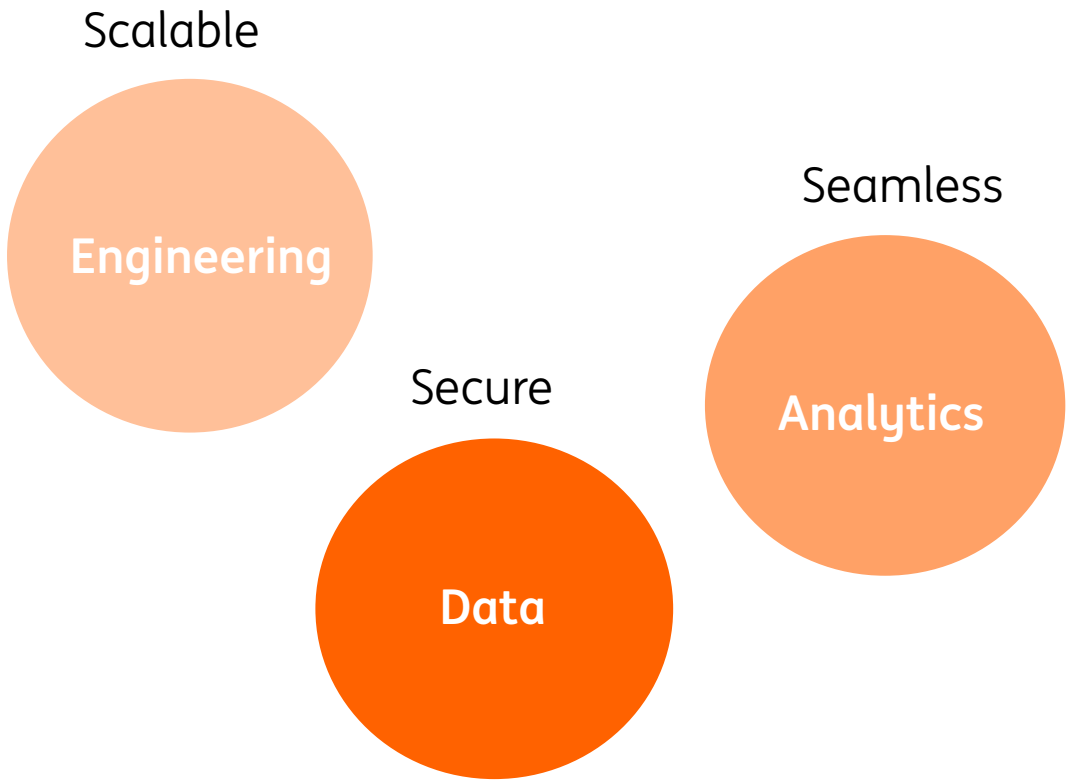
- **Modern** open-source tooling
- Significant **computation** power
- **Strictest** security & compliance measures
- All disciplines of **analytics in one place**
- Address Global and **local needs**



Self-service platform

Scalable data processing frameworks

- ML pipelines building, sharing, testing, evaluation, deployment, and scaling
- Reuse, create, share and productionize machine learning pipelines (predictive/prescriptive).



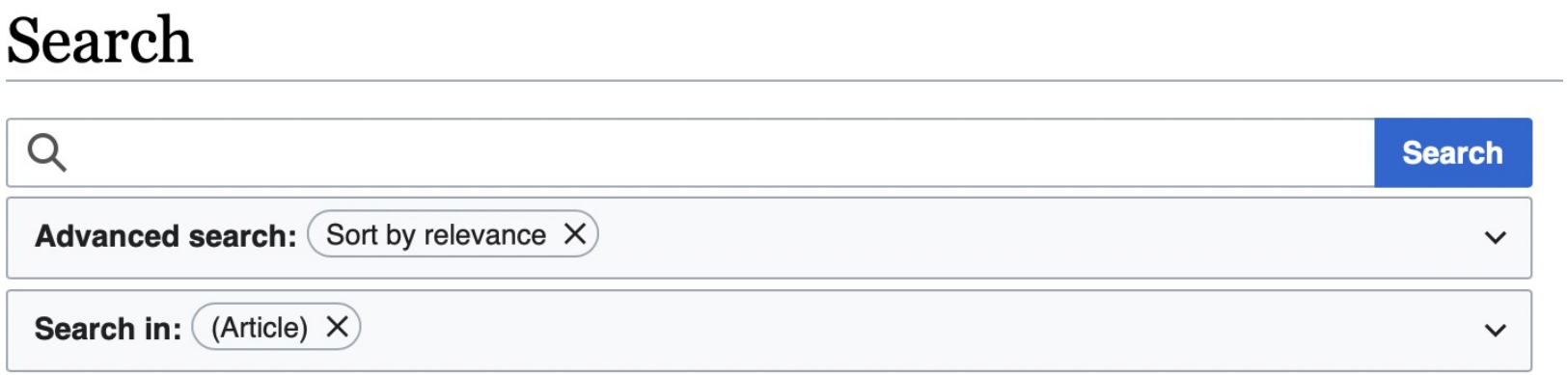
Seamless data science toolbox

- Data cleaning, sorting, merging, analysis, visualization, model creation & sharing
- Reuse, create, and publish/share analytical models (descriptive/diagnostic)

Secure connection of data producers with consumers company-wide

- Managed, need-based data access
- Discovery and data exploration
- Digitized access management

Looking for the answer



Source: wikipedia.org

Tech landscape

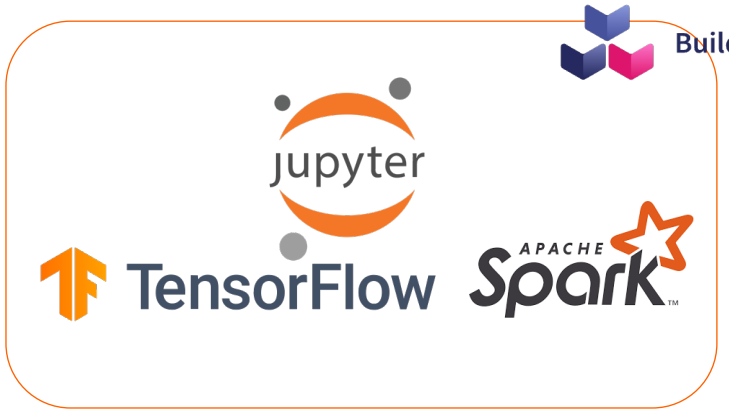
Cloud Native

DAP forms a layer on top of cloud native services providing:

- Data security
- Regulatory compliance
- Standardization
- Curated experience
- Tailored user journeys


On the journey

User interfaces



- Platform Entry-point
- Project Management

Data Science in a box
(Advanced analytics toolbox)

- Data Discovery
- Metadata engine

- SQL+BI toolset
- Dashboarding

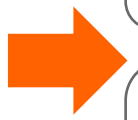
Evolution

Query Engines: Apache Spark, trino

Cache

Compute: hadoop YARN

Storage: hadoop HDFS



Query Engines: Apache Spark, trino

Cache: ALLUXIO

Compute: kubernetes

Storage: 

Key challenges

1. **Job management**
   - Pod level scheduling, no awareness of upper-level applications.
   - Lack of fine-grained lifecycle management.
   - Lack of task dependencies, job dependencies.

2. **Scheduling**
   - Lack of job based scheduling, e.g. job ordering, job priority, job pre-emption, job fair-share, job reservation.
   - Not enough advanced scheduling algorithms, E.g. CPU topology, task-topology, I0-Awareness, backfill.
   - Lack of support to resource sharing mechanism between jobs, queues, namespaces.

3. **Multi-framework support**
   - Insufficient support for frameworks e.g. Tensorflow, Pytorch.
   - Complexity managing every framework deployments (resource planning, sharing).

On the trail for a solution

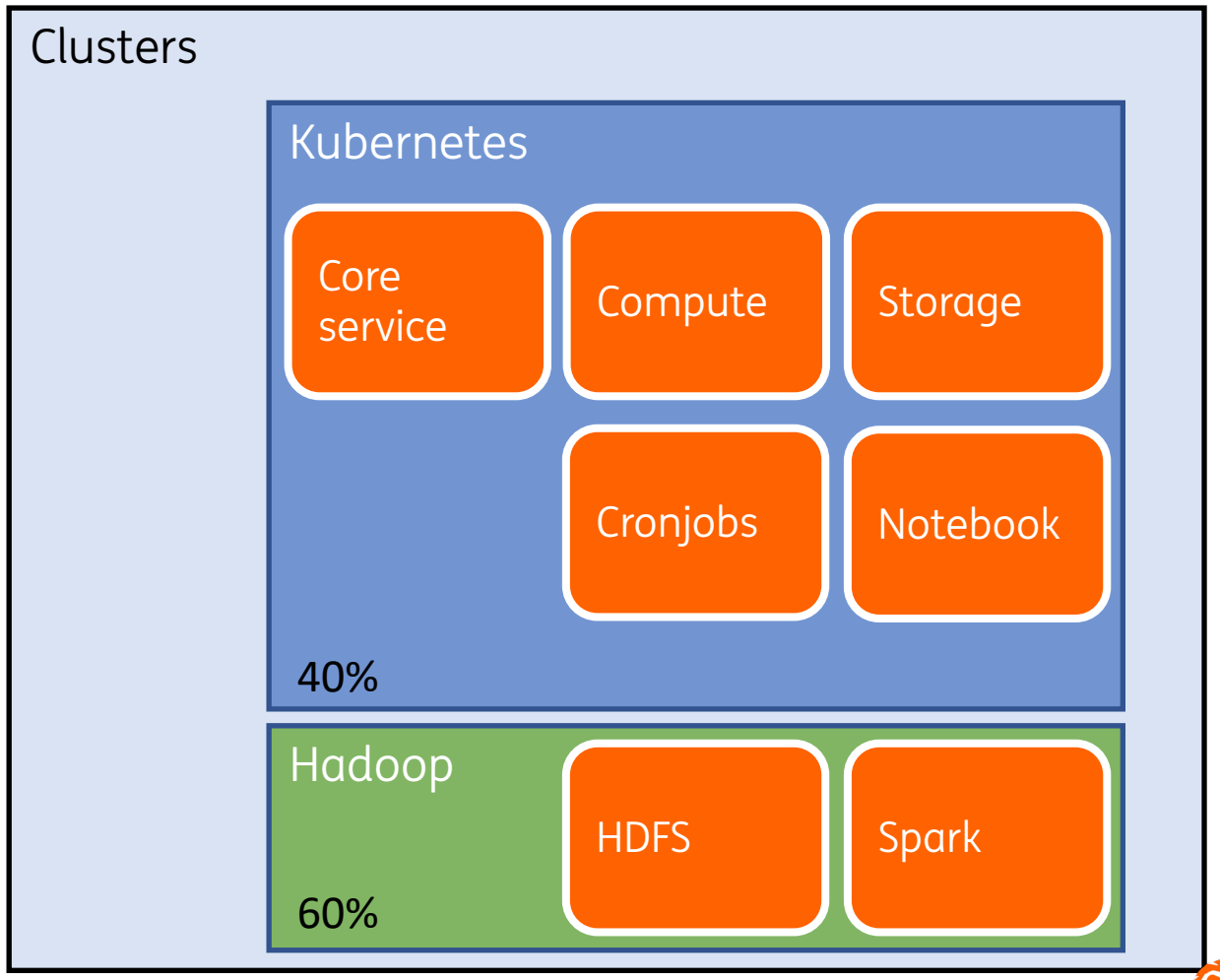
Static allocation

On-premise means having a fixed cluster

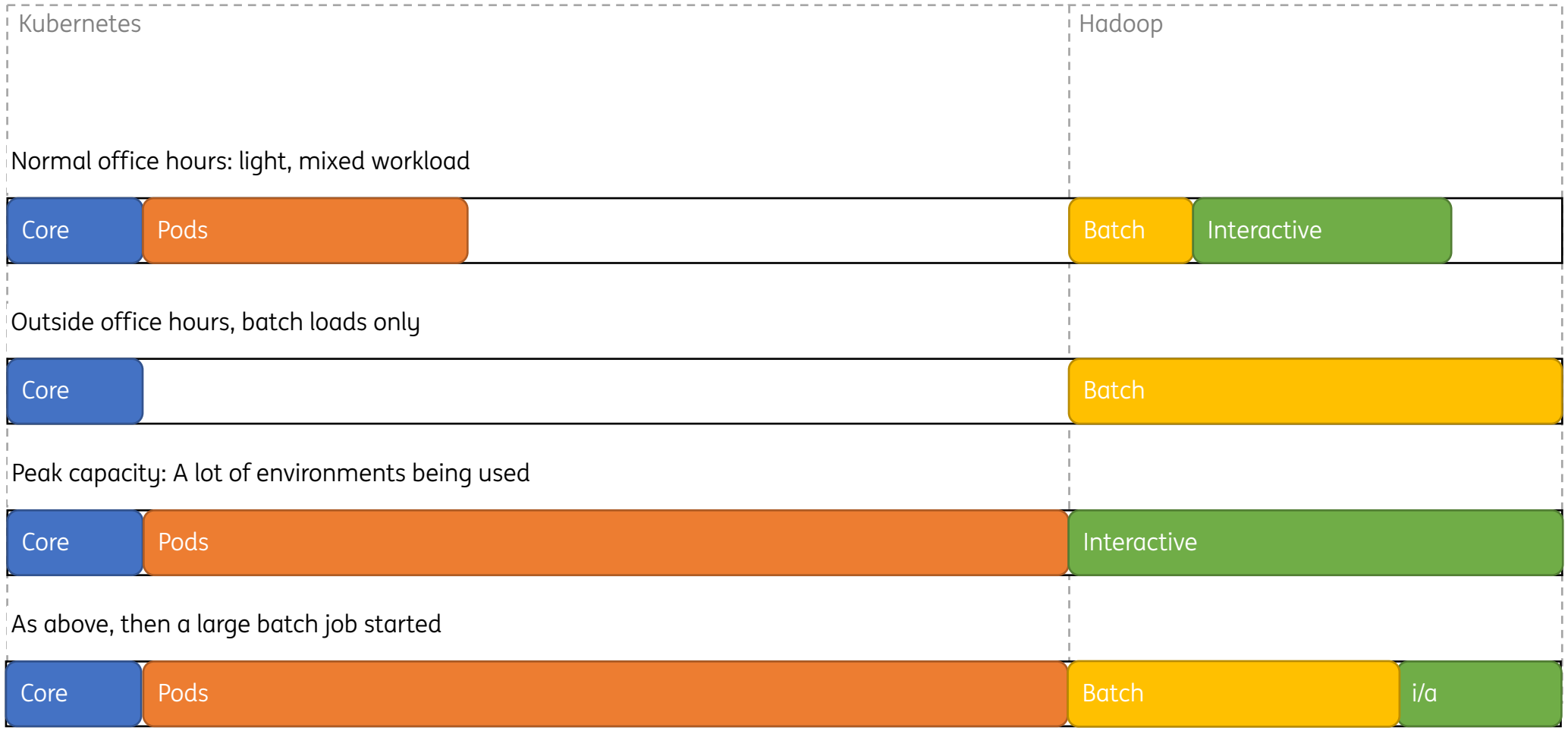
Scheduling is split between Hadoop and Kubernetes

We aim for only Kubernetes

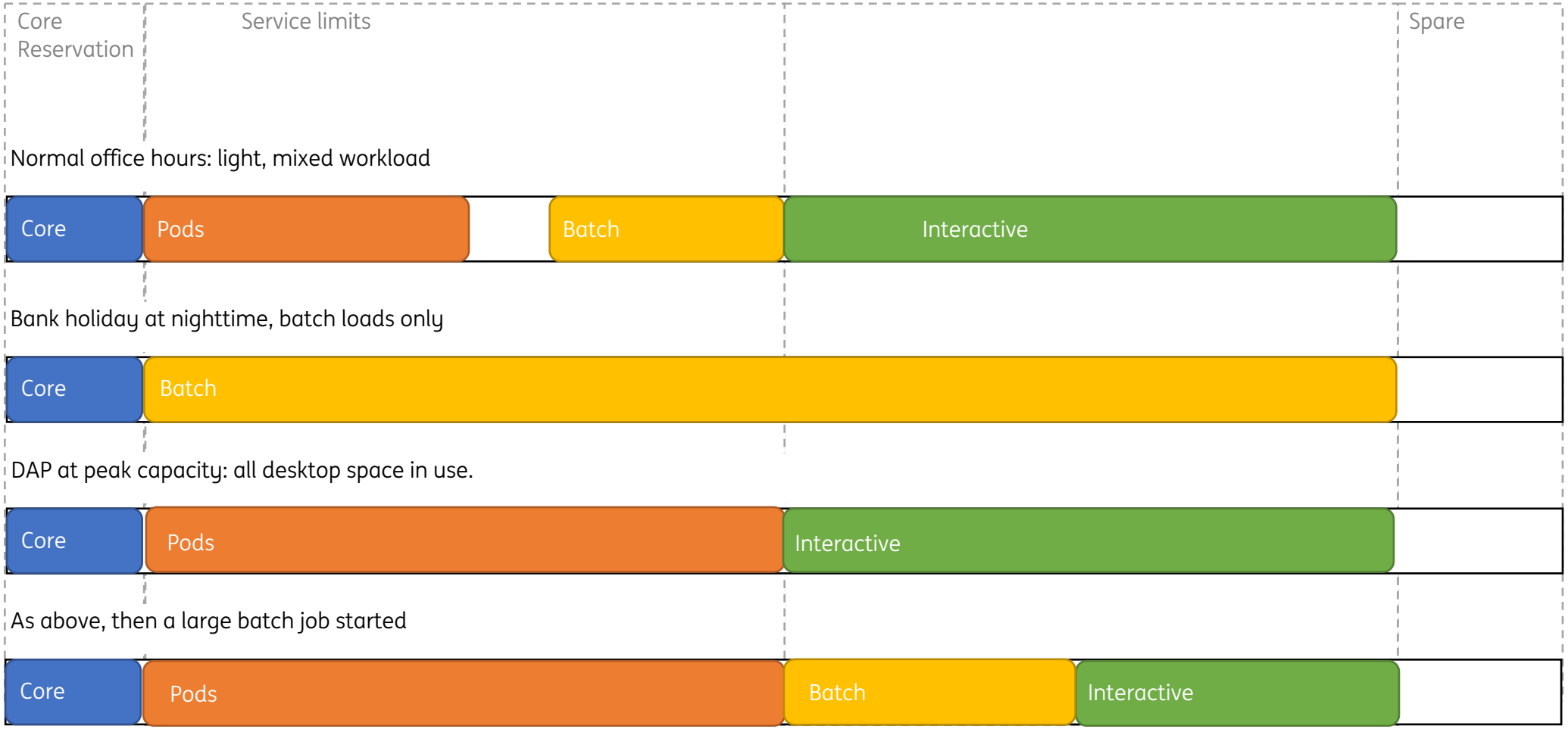
** Like the rest of the world*



Kubernetes + YARN



Kubernetes (with Volano)



Batch job scheduler for Kubernetes

Enables greater control over (task) scheduling:

- Job Queues with weighted priority
- Able to commit above Queue limits if cluster has spare capacity
- Able to preempt pods when more pods comes in
- Configurable strategies to deal with competing workloads
- (Not unlike YARN)

https://volcano.sh/

Since 3.3.0 does Spark officially have support for Volcano*

* YuniKorn is now also supported since 3.3.1

Volcano


MindSpore
TensorFlow
Kubeflow
OPEN MPI
PyTorch
PaddlePaddle
HOROVOD
APACHE Spark
MicroServices
Volcano
CRD + Controllers + Scheduler
Kubernetes

Volcano

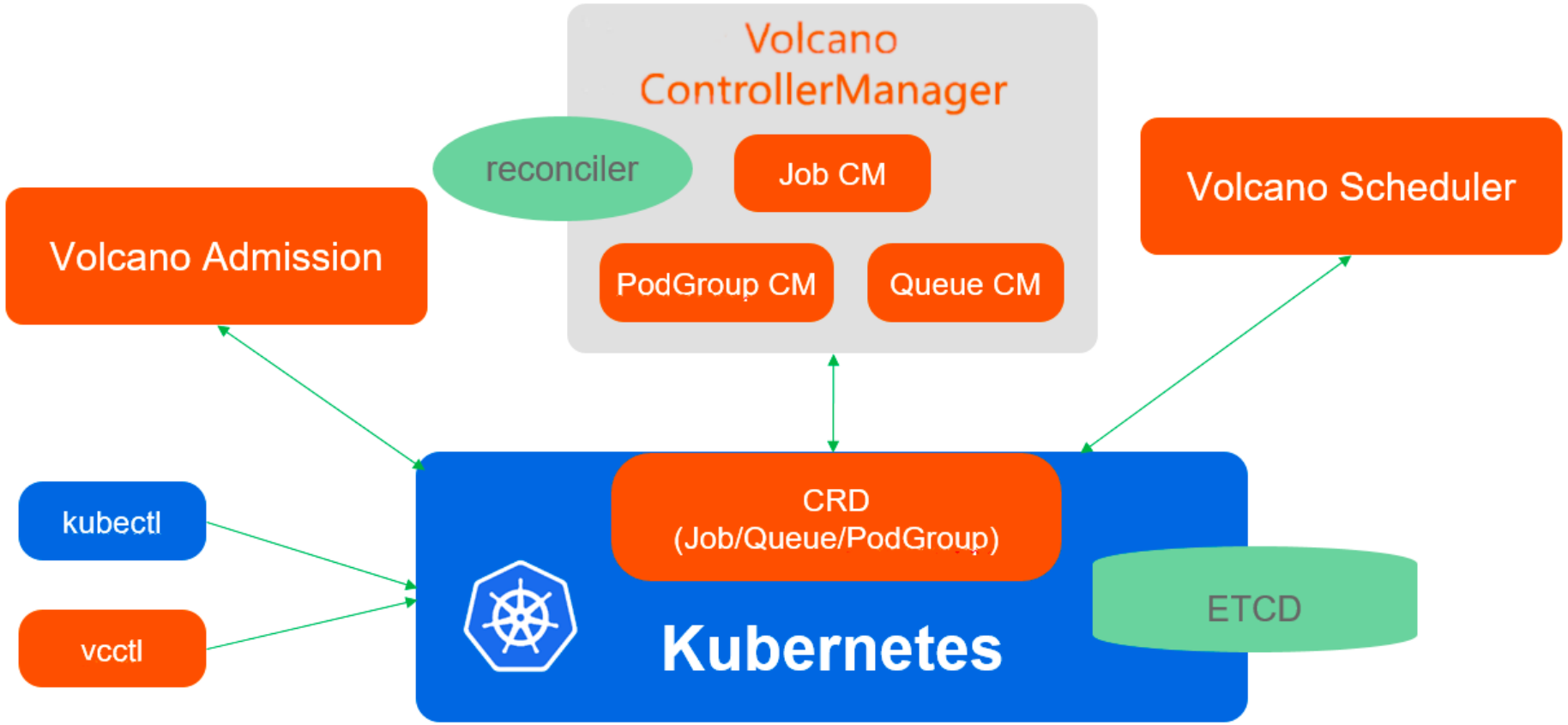
```yaml
# volcano-scheduler.conf

actions: "reclaim, enqueue, allocate, backfill"
tiers:
- plugins:
  - name: priority
  - name: gang
  - name: conformance
- plugins:
  - name: overcommit
  - name: drf
  - name: predicates
  - name: proportion
  - name: nodeorder
  - name: binpack
```

```yaml
apiVersion: batch.volcano.sh/v1alpha1
kind: Job
metadata:
  name: user1-job
spec:
  minAvailable: 1
  schedulerName: volcano
  maxRetry: 5
  queue: root-user1
  tasks:
    - replicas: 4
      name: "default-nginx"
      policies:
        - event: TaskCompleted
          action: CompleteJob
      template:
        spec:
          schedulerName: volcano
          containers:
            - image: busybox
              imagePullPolicy: IfNotPresent
              name: busybox
```

```yaml
apiVersion: scheduling.volcano.sh/v1beta1
kind: Queue
metadata:
  name: root
  annotations:
    "volcano.sh/hierarchy": "root/user1"
    "volcano.sh/hierarchy-weights": "1/1"
spec:
  weight: 1
```

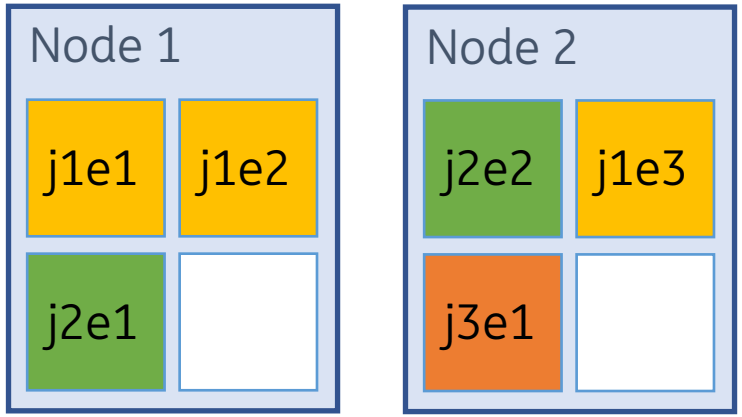
Volcano architecture



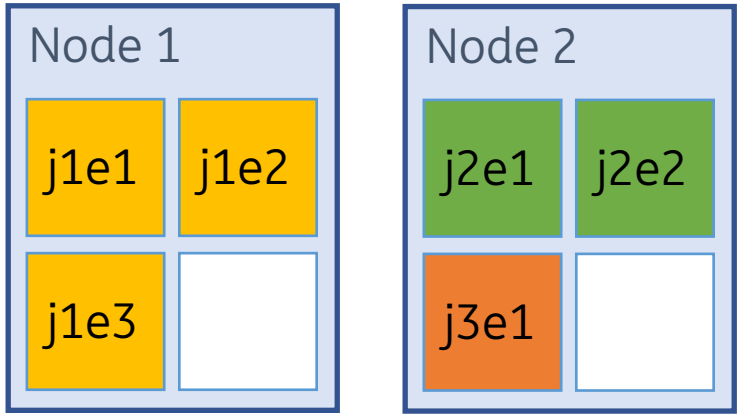
Balancing

Redundancy and locality

Anti-affinity: best for redundancy



Affinity: best for locality



- Multiple spark applications should be scheduled with anti-affinity,
- Executors within a single application should be scheduled with affinity.

Plugins

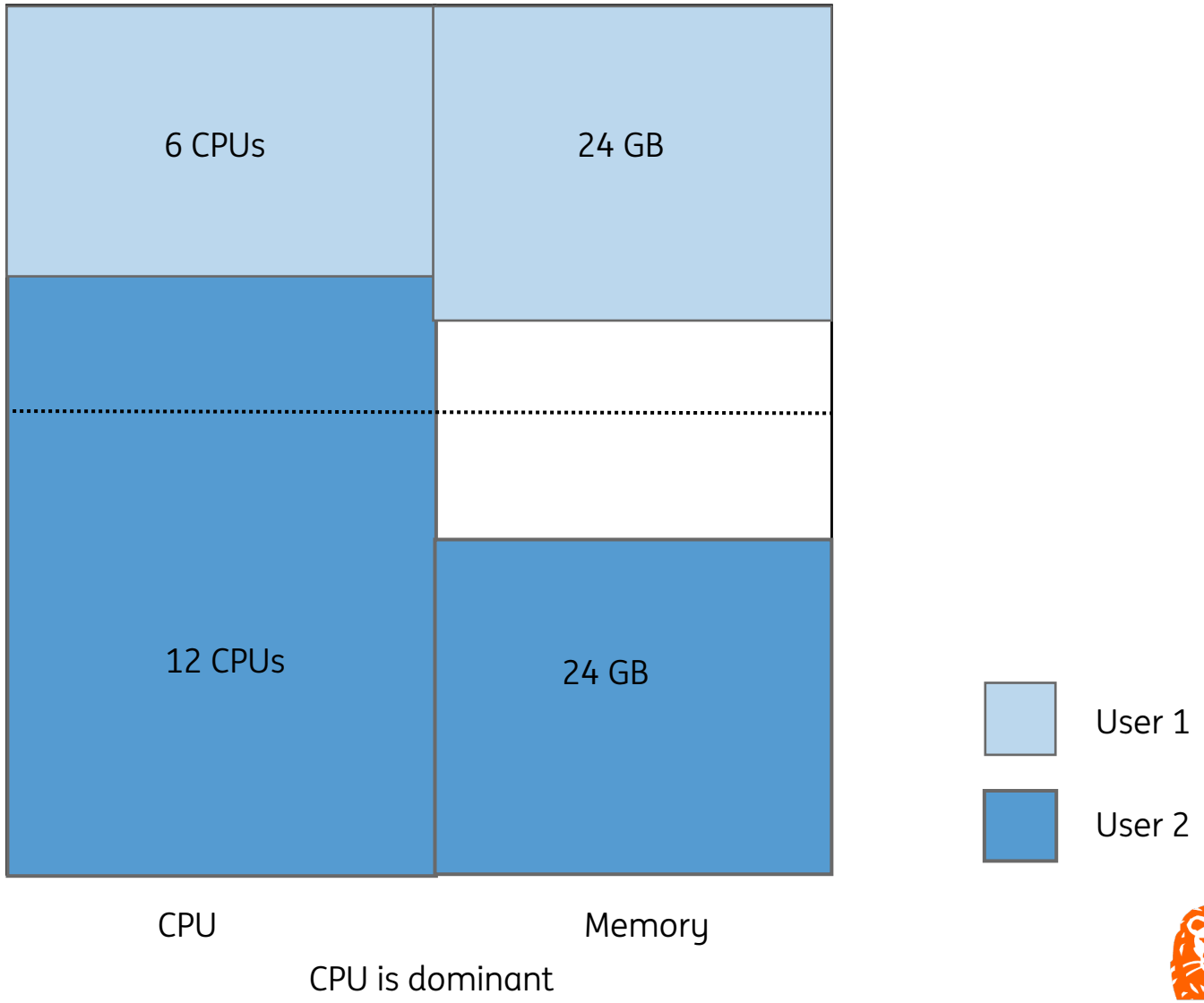
master ▾ volcano / pkg / scheduler / plugins / Go to file Add file ▾ ···

xilinxing and waiterQ merge #2506(add volume binding plugin) into scheduler/plugins/nodeorder; ... 6e0beac 7 days ago History

..		
binpack	Modify format verification by gofmt.	13 days ago
cdp	add cooldownn protection plugin	3 months ago
conformance	enable eight golangci-lint linters and fix lint errors	15 months ago
drf	feat(webhook): add task level dependson webhook	10 months ago
extender	Support JobReady for extender plugin.	4 months ago
gang	fix(scheduler): fix jobStarvingFn logic	5 months ago
nodeorder	merge #2506(add volume binding plugin) into scheduler/plugins/nodeorder;	7 days ago
numaaware	Modify format verification by gofmt.	13 days ago
overcommit	Modify format verification by gofmt.	13 days ago
predicates	merge #2506(add volume binding plugin) into scheduler/plugins/nodeorder;	7 days ago
priority	fix the CI failure	10 months ago
proportion	fix: proportion metrics accuracy	4 months ago
rescheduling	filter the rescheduling strategies which contain victim fucntions	4 months ago
resourcequota	fix 2488, scheduler panic	2 months ago
sla	Modify format verification by gofmt.	13 days ago
task-topology	Modify format verification by gofmt.	13 days ago
tdm	update preempt and TDM to match VictimTasks function	6 months ago
usage	update the codes according to the review comments	6 months ago

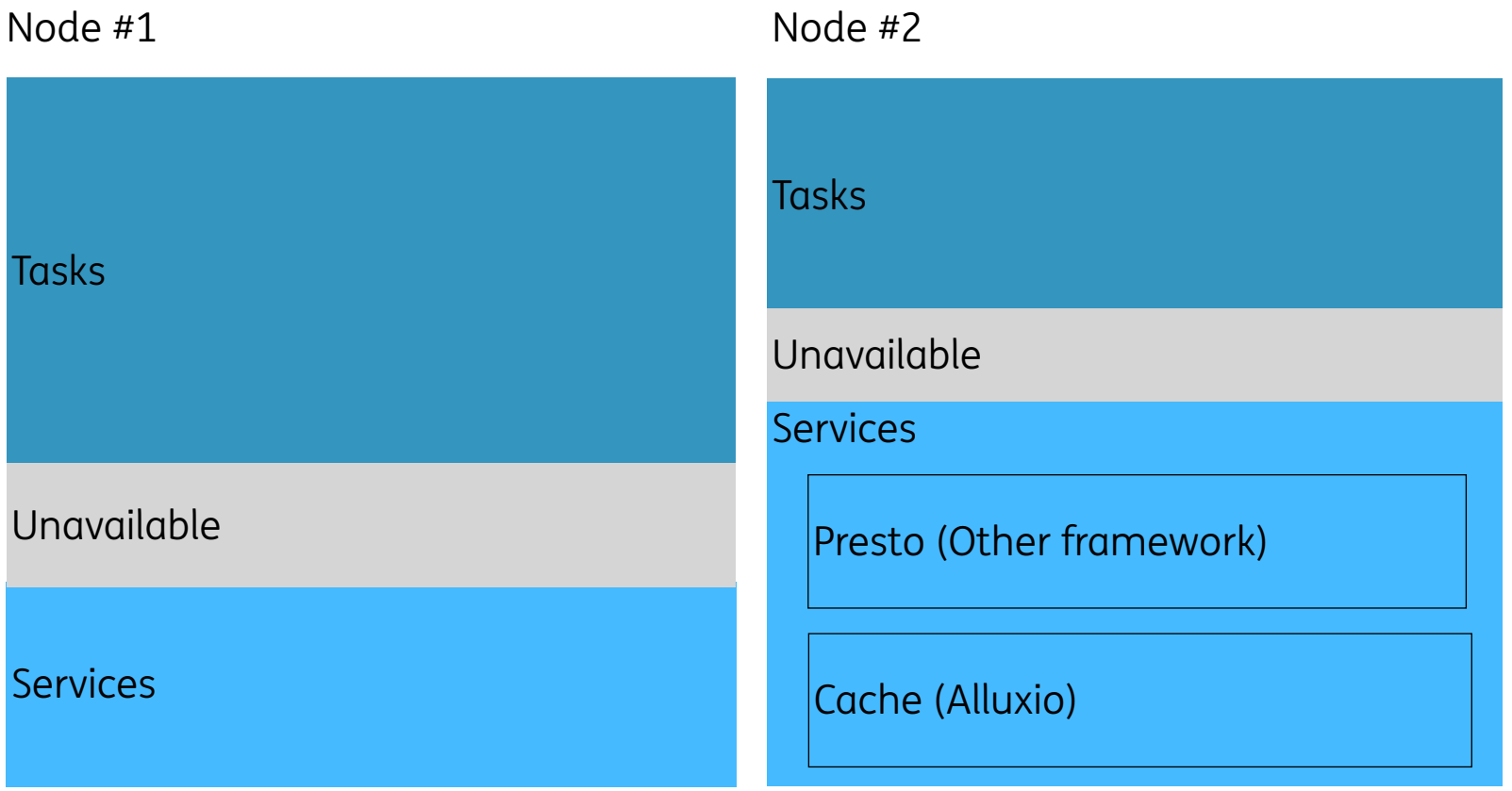
DRF

1. Everyone has a weighted claim (Queues)
2. You can overcommit on unused resources from another process
3. Calculations are done on the dominant resource
4. Other resources are preempted once a user is claiming its fair share




Stack Up

Avoid resource starvation



Resource calculation

R = Total
C = Consumed
A = Available

A = (T - C) * Factor

We choose like a factor between (0.5 and 0.8)

Spark setup

```
from pyspark.sql import SparkSession

spark = SparkSession.builder \
    .config("spark.kubernetes.executor.scheduler.name", "volcano") \
    .config("spark.kubernetes.executor.label.volcano.sh/queue-name", "root-user1") \
    .config("spark.kubernetes.executor.label.volcano.sh/group-name", "user1") \
    .getOrCreate()
```

```
Execution started at 2022-10-25 15:05:31
```

SparkSession - hive

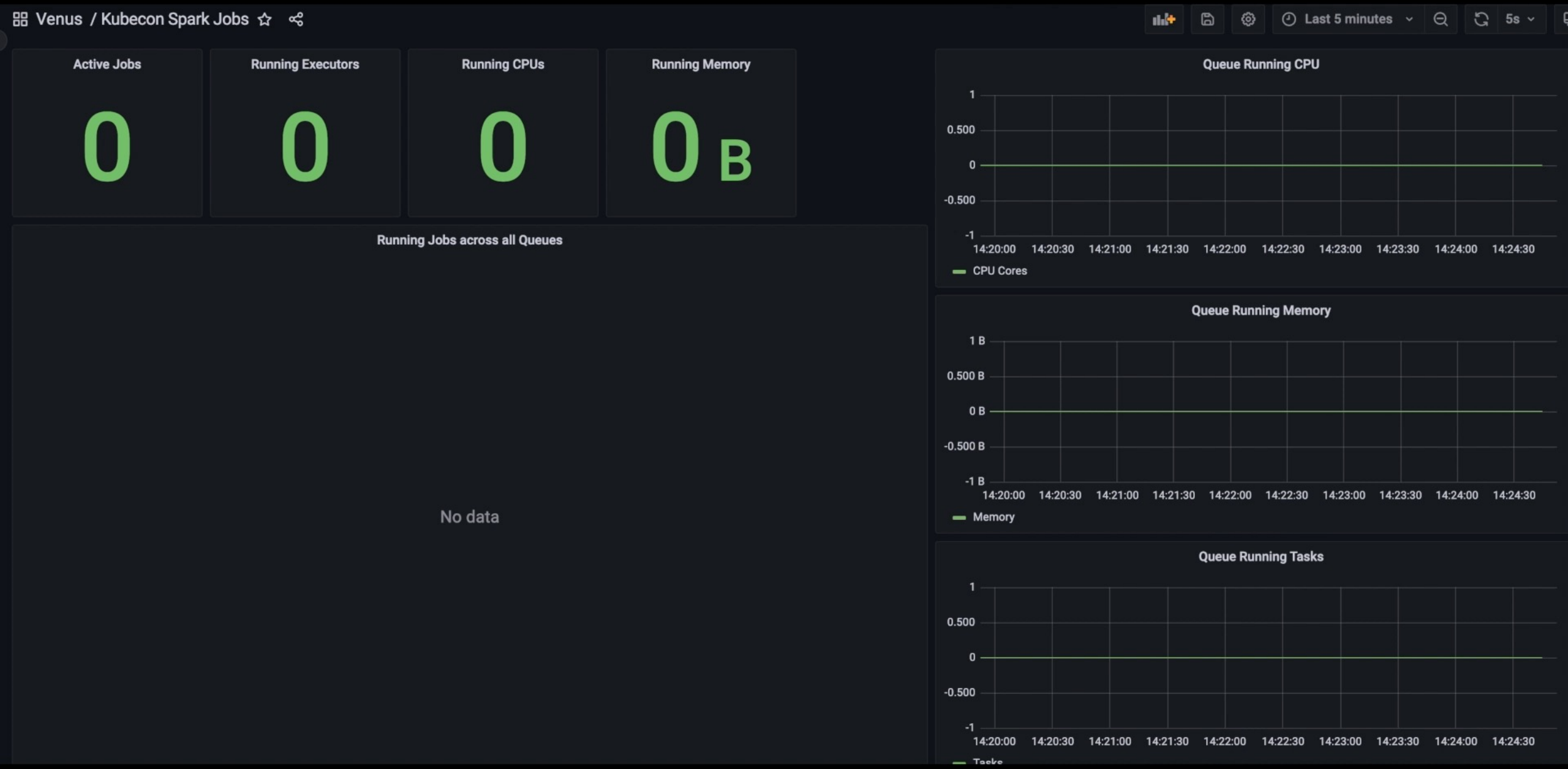
SparkContext

Spark UI

Version	`v3.3.0`
Master	

Notes

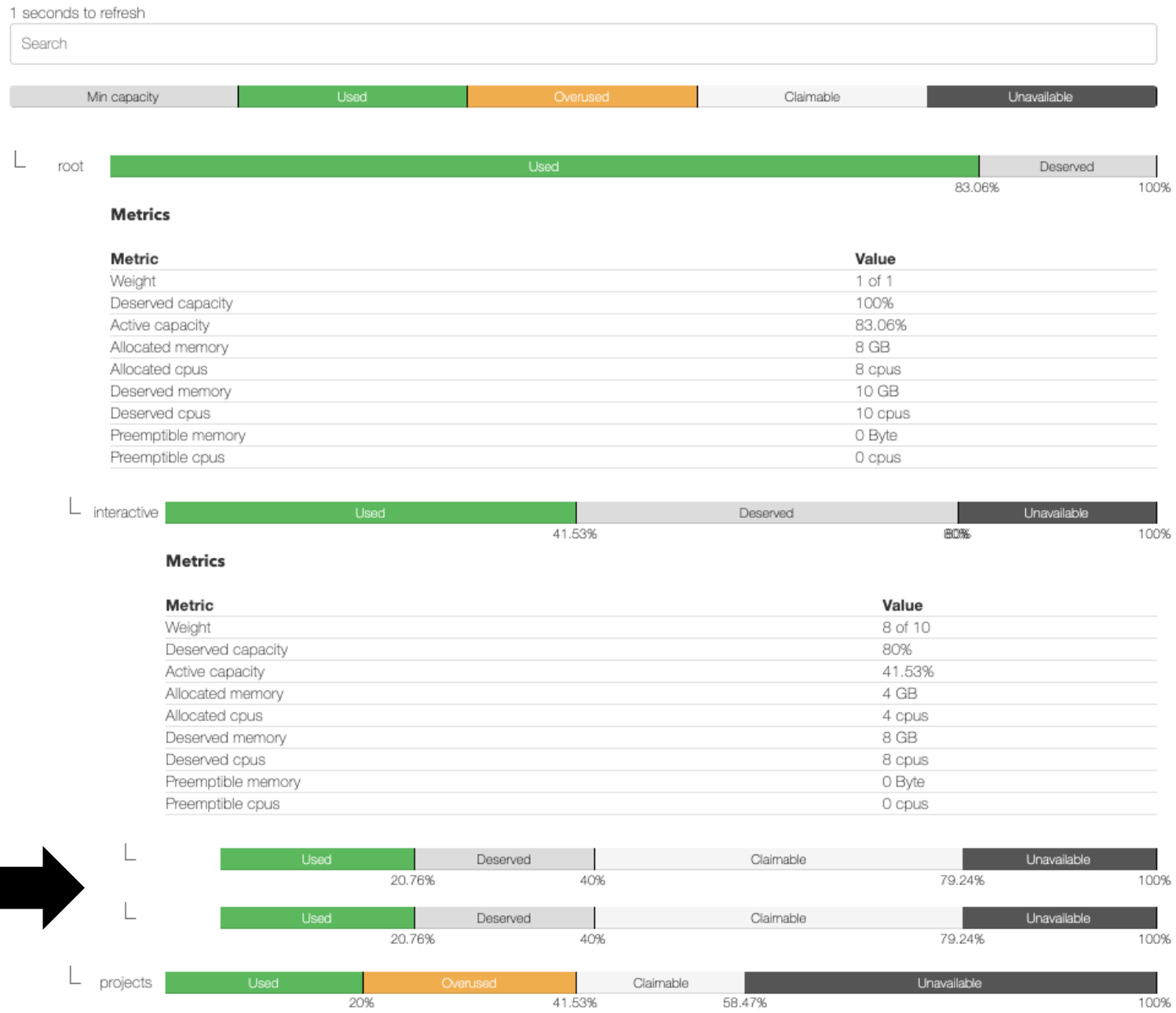
- Volcano Queue & Podgroup are created automatically
- Started pods are assigned to Volcano PodGroup
- OwnerReference & driver heartbeat for garbage collection
- MaxPendingPods for limiting allocations once the queues are full
- Enabled with dynamic allocation and minExecutors=1


Venus / Kubecon Spark Jobs
Last 5 minutes
5s
Active Jobs
0
Running Executors
0
Running CPUs
0
Running Memory
0 B
Running Jobs across all Queues
No data
Queue Running CPU
1
0.500
0
-0.500
-1
14:20:00
14:20:30
14:21:00
14:21:30
14:22:00
14:22:30
14:23:00
14:23:30
14:24:00
14:24:30
CPU Cores
Queue Running Memory
1 B
0.500 B
0 B
-0.500 B
-1 B
Memory
Queue Running Tasks
Tasks

Workload monitoring

DRF Dashboard

1. Replaces yarn UI
2. Add hierarchy

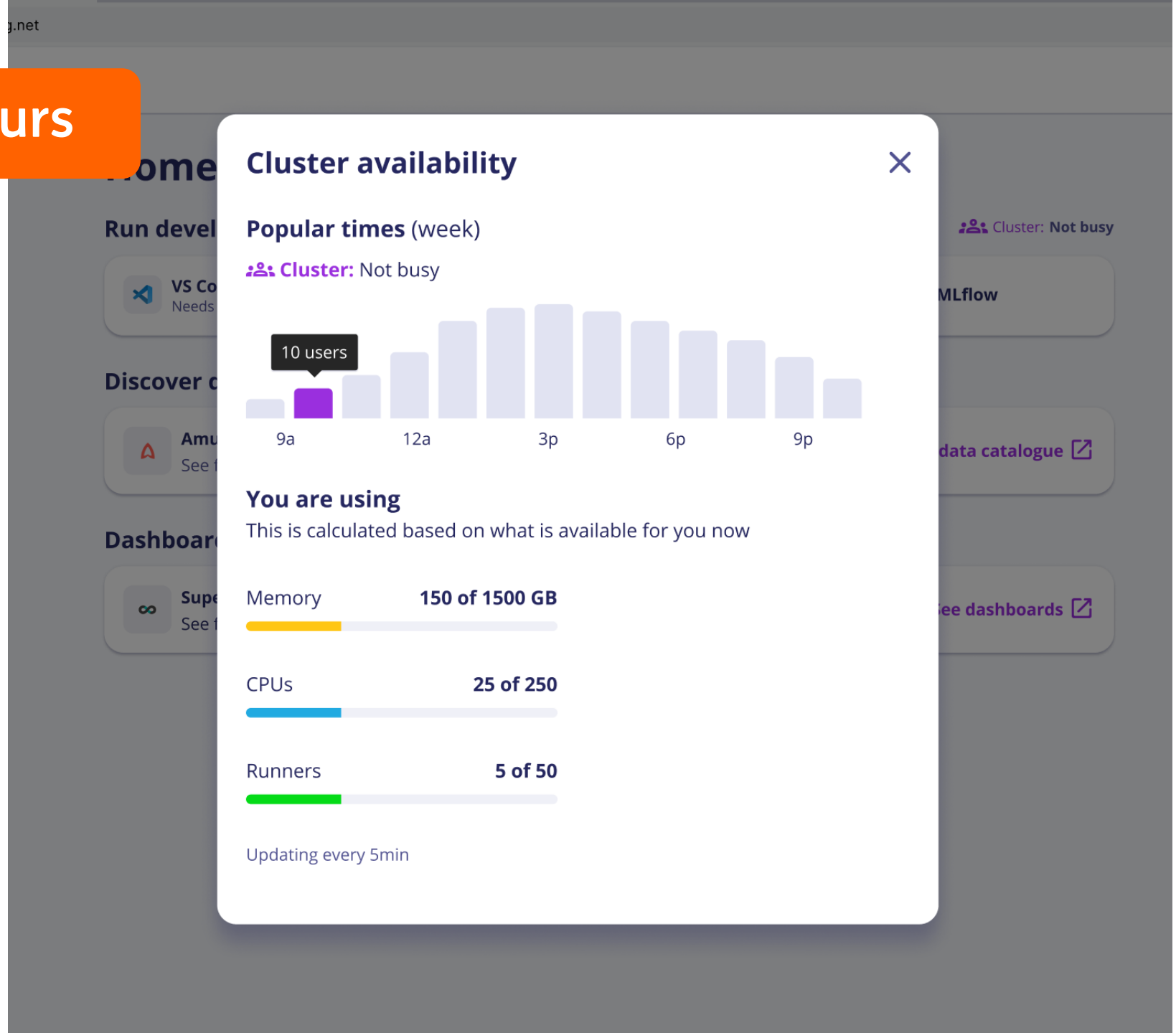


User 1

User 2

Avoid traffic jams

Cluster rush hours



Summary

All the love for the folks that created Volcano!

We would like to add our work to Volcano soon:

1. DRF Dashboard
2. Adding spare space on each node
3. Automatic Queue management
4. More Prometheus metrics
5. Updates to the Grafana dashboards
6. Updates to Kube-state-metrics
7. Limiting cluster roles

do your thing