

cloudera

Where will my pod land?

Peter Ableda

Cloudera Data Science Workbench - Use case

The screenshot displays the Cloudera Data Science Workbench interface. At the top, there's a navigation bar with 'Documentation' and 'Projects' tabs. Below this, a dashboard shows '2 sessions running' and '0 jobs running'. Two progress indicators show '2.00 vCPU' and '4.00 GiB' usage out of 4 vCPU and 15.17 GiB respectively.

The 'Projects' section lists several projects:

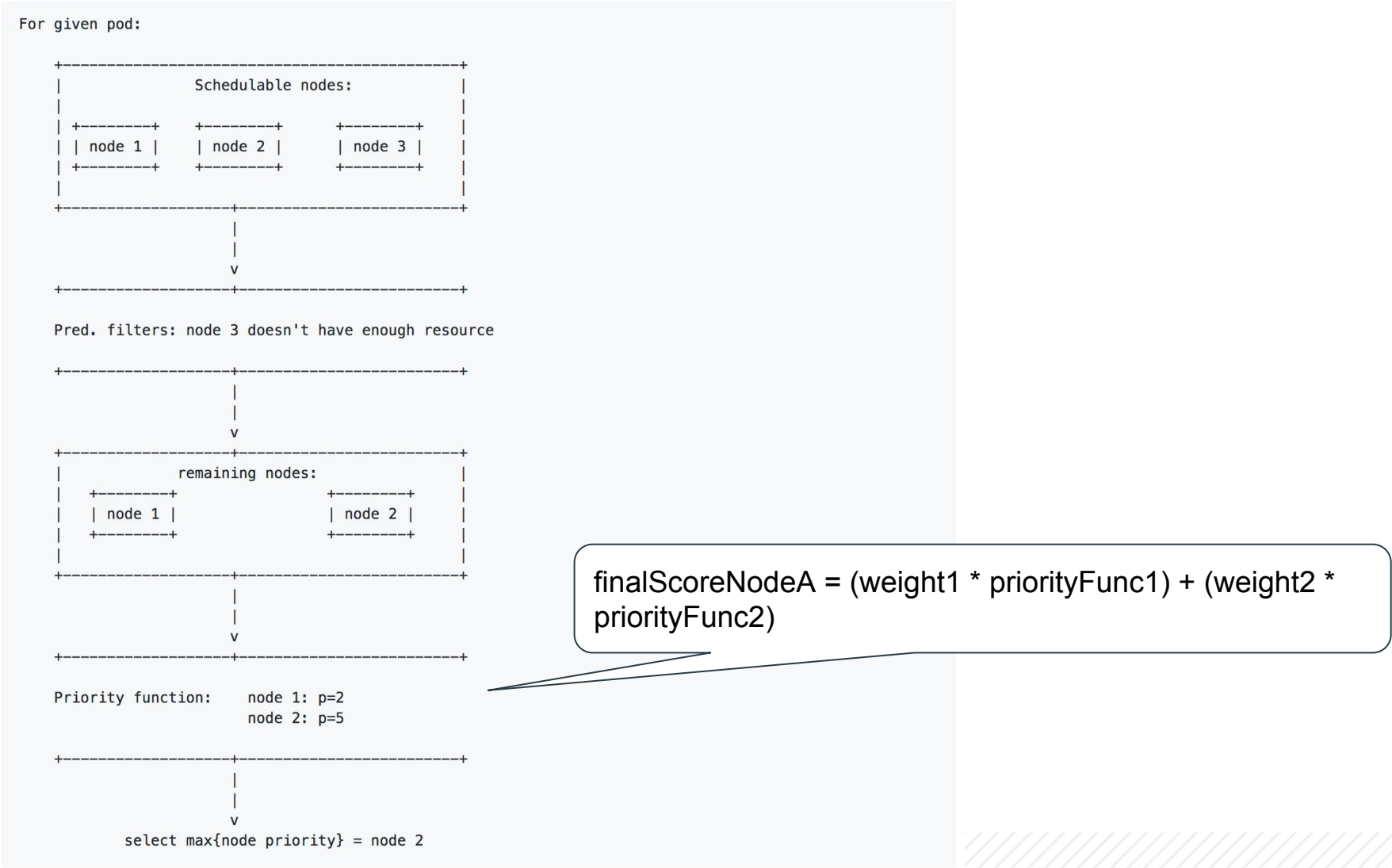
- Python Template Project** by Edith Potter. Last worked on just now.
- Python Visualizations** by Lucy Milton. Last worked on just now.
- Air Quality San Francisco** by Lucy Milton. Last worked on just now.
- GitHub Data** by Lucy Milton. Last worked on 1 minute ago.
- Yahoo Search Data** by Lucy Milton. Last worked on 2 minutes ago.
- PySpark Tests**

The central area features a code editor with a file explorer on the left showing 'analysis.r', 'Boston.csv', and 'README.md'. The code in 'analysis.r' includes setup, data loading, and plotting instructions. To the right of the code editor, a console window shows the execution of R commands, including reading the 'Boston.csv' file and displaying its structure and a summary of the 'medv' variable.

Below the code editor, the 'Cluster Metadata' section provides an overview of the cluster's performance, including a 'Job History' graph showing duration over time.

Demo setup and affinity recap

Kubernetes Scheduling algorithm



Predicates

Filtering the nodes

- `PodFitsResources`: Check if the free resource (CPU and Memory) meets the requirement of the Pod. The free resource is measured by the capacity minus the sum of requests of all Pods on the node.
- `PodFitsHostPorts`: Check if any `HostPort` required by the Pod is already occupied on the node.
- `MatchNodeSelector`: Check if the labels of the node match the labels specified in the Pod's `nodeSelector` field and, as of Kubernetes v1.2, also match the `scheduler.alpha.kubernetes.io/affinity` pod annotation if present.
- `HostName`: Filter out all nodes except the one specified in the PodSpec's `NodeName` field.
- `CheckNodeUnschedulable`, `CheckNodeMemoryPressure`, `CheckNodeDiskPressure`, ...

Priorities

Ranking the nodes

- `LeastRequestedPriority`: The node is prioritized based on the fraction of the node that would be free if the new Pod were scheduled onto the node.
- `MostRequestedPriority`: Opposite of the above.
- `BalancedResourceAllocation`: This priority function tries to put the Pod on a node such that the CPU and Memory utilization rate is balanced after the Pod is deployed.
- `SelectorSpreadPriority`: Spread Pods by minimizing the number of Pods belonging to the same service, replication controller, or replica set on the same node.
- `ImageLocalityPriority`: Nodes are prioritized based on locality of images requested by a pod.
- `NodeAffinityPriority`: (Kubernetes v1.2) Implements `preferredDuringSchedulingIgnoredDuringExecution` node affinity.
- ...

Example scheduler conf

```
{ "kind" : "Policy", "apiVersion" : "v1",  
  "metadata" : {"name": "scheduler-policy-config"},  
  "predicates" : [  
    {"name" : "PodFitsHostPorts"},  
    {"name" : "PodFitsResources"},  
    {"name" : "NoDiskConflict"},  
    {"name" : "NoVolumeZoneConflict"},  
    {"name" : "MatchNodeSelector"},  
    {"name" : "HostName"}  
  ],
```

```
  "priorities" : [  
    {"name" : "LeastRequestedPriority", "weight" : 1},  
    {"name" : "BalancedResourceAllocation", "weight" : 1},  
    {"name" : "ServiceSpreadingPriority", "weight" : 1},  
    {"name" : "EqualPriority", "weight" : 1},  
    {"name": "NodeAffinityPriority", "weight": 10}  
  ]  
}
```

Configure Kube-scheduler demo

Other concerns

What other configurations make sense?

- “I want to let my workloads to burst CPU/Memory”
 - > Default config is good for this, you want your pods evenly distributed.
- “I would like to enable auto-scaling without the need to cordon/drain my nodes”
 - > You need to go from LeastRequestedPriority to MostRequestedPriority
- “I need to schedule big workloads but they can’t fit on any of the hosts. E.g. want 16vCPU for a Pod and each machines have 32vCPU but all nodes are 60% utilized.”
 - > You need to go from LeastRequestedPriority to MostRequestedPriority

THANK YOU