SCHEDULING

Scheduling

Scheduling Policies

The default scheduler contains a number of predicates and priorities; however, these can be changed via a scheduler policy file.

A short version is shown below:

```
"kind" : "Policy",
"apiVersion" : "v1",
"predicates" : [
        {"name" : "MatchNodeSelector", "order": 6},
        {"name" : "PodFitsHostPorts", "order": 2},
        {"name" : "PodFitsResources", "order": 3},
        {"name" : "NoDiskConflict", "order": 4},
        {"name" : "PodToleratesNodeTaints", "order": 5},
        {"name" : "PodFitsHost", "order": 1}
        ],
"priorities" : [
        {"name" : "LeastRequestedPriority", "weight" : 1},
        {"name" : "BalancedResourceAllocation", "weight" : 1},
        {"name" : "ServiceSpreadingPriority", "weight" : 2},
        {"name" : "EqualPriority", "weight" : 1}
        ],
"hardPodAffinitySymmetricWeight" : 10
}
```

Typically, you will configure a scheduler with this policy using the **`--policy-config-file`** parameter and define a name for this scheduler using the **`--scheduler-name`** parameter. You will then have two schedulers running and will be able to specify which scheduler to use in the pod specification.

With multiple schedulers, there could be conflict in the Pod allocation. Each Pod should declare which scheduler should be used. But, if separate schedulers determine that a node is eligible because of available resources and both attempt to deploy, causing the resource to no longer be available, a conflict would occur. The current solution is for the local kubelet to return the Pods to the scheduler for reassignment. Eventually, one Pod will succeed and the other will be scheduled elsewhere.