

# Safe overload predicate:

```
[ info ] 2019/04/12 17:20:53 predicate.go:15: ===> SafeOverloadPredicate: applying SafePredicate for pod test-pod on node worker-node2 ...
[ info ] 2019/04/12 17:20:53 predicate.go:30: Checking cpu fit based on usage history :
meanFreeCPU = 400; stdFreeCPU = 100
freeAvg = 400.000000; freeStdev = 100.000000
capacity = 1000.000000; demand = 200.000000
usedAvg = 600.000000; usedStdev = 100.000000
BetaDistribution: alpha = 12.000000; beta = 3.000000; mean = 0.800000; var = 0.010000; m1 = 0.800000; m2 = 0.650000; m3 = 0.535294;
mu = 0.800000; sigma = 0.100000; risk(0.900000) = 0.158360      (<= 0.300000) accepted!
Checking memory fit based on usage history :
meanFreeMemory = 800000000; stdFreeMemory = 100000000
forecastedFreeMemory = 900000000; safeForecastWeight = 0.200000; weightedFreeMemory = 820000000.000000
freeAvg = 782.012939 MB; freeStdev = 95.367432 MB
capacity = 1396.113281 MB; demand = 200.000000 MB
usedAvg = 614.100342; usedStdev = 95.367432
BetaDistribution: alpha = 29.795476; beta = 21.301247; mean = 0.583119; var = 0.004666; m1 = 0.583119; m2 = 0.344694; m3 = 0.206410;
mu = 0.583119; sigma = 0.068309; risk(0.900000) = 0.000000      (<= 0.300000) accepted!
okCPU = true; okMemory = true; okOverall = true;
```

Annotations on the log output:

- timeStamp**: Points to the timestamp `2019/04/12 17:20:53`.
- podName**: Points to the pod name `test-pod`.
- nodeName**: Points to the node name `worker-node2`.
- Fits on CPU**: A blue bracket groups the CPU-related log entries.
- Fits on Memory**: A green bracket groups the memory-related log entries.

# Safe overload priority:

```
[ info ] 2019/04/12 17:20:53 priority.go:24: ==> SafePriority: calculating priority for pod test-pod on node worker-node1 ...
[ info ] 2019/04/12 17:20:53 priority.go:53: Calculating priority based on usage history ...
CPU usage statistics:
meanFreeCPU = 400; stdFreeCPU = 100
freeAvg = 400.000000; freeStdev = 100.000000
capacity = 1000.000000; demand = 200.000000
usedAvg = 600.000000; usedStdev = 100.000000
BetaDistribution: alpha = 12.000000; beta = 3.000000; mean = 0.800000; var = 0.010000; m1 = 0.800000; m2 = 0.650000; m3 = 0.535294;
mu = 0.800000; sigma = 0.100000; risk(0.900000) = 0.158360; riskFraction = 0.527867
CPUScore = 2
Memory usage statistics:
meanFreeMemory = 800000000; stdFreeMemory = 100000000
forecastedFreeMemory = 900000000; safeForecastWeight = 0.200000; weightedFreeMemory = 820000000.000000
freeAvg = 782.012939 MB; freeStdev = 95.367432 MB
capacity = 1396.113281 MB; demand = 200.000000 MB
usedAvg = 614.100342; usedStdev = 95.367432
BetaDistribution: alpha = 29.795476; beta = 21.301247; mean = 0.583119; var = 0.004666; m1 = 0.583119; m2 = 0.344694; m3 = 0.206410;
mu = 0.583119; sigma = 0.068309; risk(0.900000) = 0.000000; riskFraction = 0.000000
memoryScore = 9
overallScore = 2
```

Annotations on the log output:

- timeStamp**: Points to the timestamp [ info ] 2019/04/12 17:20:53.
- podName**: Points to the pod name test-pod.
- nodeName**: Points to the node name worker-node1.
- overallScore**: Points to the overall priority score 2.

Groupings indicated by curly braces:

- CPU score**: Groups CPU usage statistics and BetaDistribution parameters.
- Memory score**: Groups Memory usage statistics and BetaDistribution parameters.

# Safe balance priority:

```
[ info ] 2019/04/12 17:20:53 priority.go:24: ==> SafePriority: calculating priority for pod test-pod on node worker-node1 ...
[ info ] 2019/04/12 17:20:53 priority.go:53: Calculating priority based on usage history ...
CPU usage statistics:
meanFreeCPU = 600; stdFreeCPU = 200
forecastedFreeCPU = 1200; safeForecastWeight = 0.200000; weightedFreeCPU = 720.000000
freeAvg = 720.000000; freeStdev = 200.000000
capacity = 1000.000000; demand = 200.000000
usedAvg = 280.000000; usedStdev = 200.000000
BetaDistribution: alpha = 2.515200; beta = 2.724800; mean = 0.480000; var = 0.040000; m1 = 0.480000; m2 = 0.270400; m3 = 0.168634;
mu = 0.480000; sigma = 0.200000; risk(0.900000) = 0.010095; riskFraction = 0.033649
CPUScore = 8
Memory usage statistics:
meanFreeMemory = 800000000; stdFreeMemory = 200000000
freeAvg = 762.939453 MB; freeStdev = 190.734863 MB
capacity = 1396.113281 MB; demand = 200.000000 MB
usedAvg = 633.173828; usedStdev = 190.734863
BetaDistribution: alpha = 7.097218; beta = 4.795282; mean = 0.596781; var = 0.018665; m1 = 0.596781; m2 = 0.374812; m3 = 0.245438;
mu = 0.596781; sigma = 0.136618; risk(0.900000) = 0.003974; riskFraction = 0.013246
memoryScore = 8
overallScore = 8
```

Annotations:

- timeStamp**: Points to the timestamp in the log.
- podName**: Points to the pod name in the log.
- nodeName**: Points to the node name in the log.
- CPU score**: Groups the CPU usage statistics and BetaDistribution details.
- Memory score**: Groups the Memory usage statistics and BetaDistribution details.
- overallScore**: Points to the overall score at the bottom of the log.