

k8s cadvisor cpu seconds to cpu cores 697 views



ishulakshmi@gmail.com

to Prometheus Users

Hi All,

I have a question on understanding cpu usage of a pod that is running in k8s cluster.

Cadvisor is only reporting the cpu seconds container_cpu_usage_seconds_total. While I want to know how to convert to cpu cores(as the pods are scheduled in terms of cpu cores) and thus alerting for high cpu usage.

I quite could not get, how to convert the cpu seconds reported to cpu cores.

I have read <https://www.robustperception.io/understanding-machine-cpu-usage> but cadvisor is not reporting idle time. so anyone who solved and let me know could be great help.

Thanks
Eswar



Brian Candler

to Prometheus Users

`rate(container_cpu_usage_seconds_total[2m])` # choose your interval as appropriate to cover at least 2 sampling periods

The rate of increase of container_cpu_usage_seconds_total gives you the number of CPU-seconds consumed per second.

If this value is 1, then you are effectively using 1 whole core (or something equivalent, e.g. 1/4 each of 4 cores). If the value is 2, then you are effectively using 2 whole cores. And so on.

You haven't shown any examples of your metric: you say there's no "idle" label, but not what other labels you have. Therefore if there are multiple metrics for the same pod, you may need to sum over them. See: <https://www.robustperception.io/rate-then-sum-never-sum-then-rate>

...

December 30, 2015

Understanding Machine CPU usage

High CPU load is a common cause of issues. Let's look at how to dig into it with [Prometheus](#) and the [Node exporter](#).

On a [Node exporters' metrics](#) page, part of the output is:

```
# HELP node_cpu Seconds the cpus spent in each mode.
# TYPE node_cpu counter
node_cpu_seconds_total{cpu="0",mode="guest"} 0
node_cpu_seconds_total{cpu="0",mode="idle"} 2.03442237e+06
node_cpu_seconds_total{cpu="0",mode="iowait"} 3522.37
node_cpu_seconds_total{cpu="0",mode="irq"} 0.48
node_cpu_seconds_total{cpu="0",mode="nice"} 515.56
node_cpu_seconds_total{cpu="0",mode="softirq"} 953.06
node_cpu_seconds_total{cpu="0",mode="steal"} 0
node_cpu_seconds_total{cpu="0",mode="system"} 6605.46
node_cpu_seconds_total{cpu="0",mode="user"} 23343.01
node_cpu_seconds_total{cpu="1",mode="guest"} 0
```

```
node_cpu_seconds_total{cpu="1",mode="idle"} 2.03471439e+06
node_cpu_seconds_total{cpu="1",mode="iowait"} 3633.5
node_cpu_seconds_total{cpu="1",mode="irq"} 0.58
node_cpu_seconds_total{cpu="1",mode="nice"} 542.05
node_cpu_seconds_total{cpu="1",mode="softirq"} 880.49
node_cpu_seconds_total{cpu="1",mode="steal"} 0
node_cpu_seconds_total{cpu="1",mode="system"} 6581.92
node_cpu_seconds_total{cpu="1",mode="user"} 23171.06
```

This metric comes from `/proc/stat` and tell us how many seconds each CPU spent doing each type of work:

- user: The time spent in userland
- system: The time spent in the kernel
- iowait: Time spent waiting for I/O
- idle: Time the CPU had nothing to do
- irq&softirq: Time servicing interrupts
- guest: If you are running VMs, the CPU they use
- steal: If you are a VM, time other VMs "stole" from your CPUs

These modes are mutually exclusive. A high iowait means that you are disk or network bound, high user or system means that you are CPU bound.

These are counters, so to calculate the per-second values we use the `rate` function in the [expression browser](#):

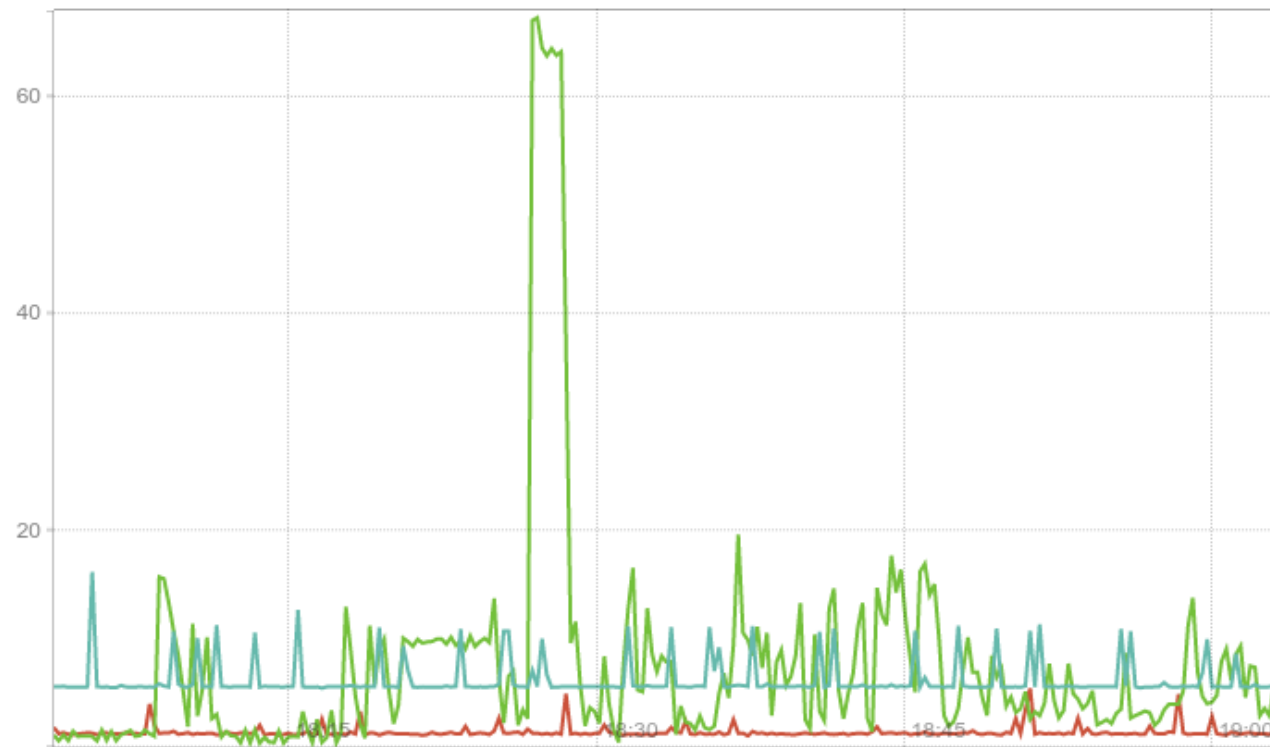
```
rate(node_cpu_seconds_total{job="node"}[1m])
```

We can aggregate this to get the overall value across all CPUs for the machine:

```
sum by (mode, instance)
(rate(node_cpu_seconds_total{job="node"}[1m]))
```

As these values always sum to one second per second for each cpu, the per-second rates are also the ratios of usage. We can use this to calculate the percentage of CPU used, by subtracting the idle usage from 100%:

```
100 - (avg by (instance)
(rate(node_cpu_seconds_total{job="node",mode="idle"}[1m])) *
100)
```



CPU Used % across several machines