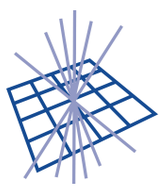
Container, System and Application Monitoring with Prometheus

Gareth Roy

HEPSYSMAN 16/1/2018

University of Glasgow



GridPP

UK Computing for Particle Physics

Large-scale cluster management at Google with Borg

Abhishek Verma† Luis Pedrosa‡ Madhukar Korupolu
David Oppenheimer Eric Tune John Wilkes

Google Inc.

Abstract

Google's Borg system is a cluster manager that runs hundreds of thousands of jobs, from many thousands of different applications, across a number of clusters each with up to tens of thousands of machines.

It achieves high utilization by combining admission control, efficient task-packing, over-commitment, and machine sharing with process-level performance isolation. It supports high-availability applications with runtime features that minimize fault-recovery time, and scheduling policies that reduce the probability of correlated failures. Borg simplifies life for its users by offering a declarative job specification language, name service integration, real-time job monitoring, and tools to analyze and simulate system behavior.

We present a summary of the Borg system architecture and features, important design decisions, a quantitative analysis of some of its policy decisions, and a qualitative examination of lessons learned from a decade of operational experience with it.

1. Introduction

The cluster management system we internally call Borg admits, schedules, starts, restarts, and monitors the full range of applications that Google runs. This paper explains how.

Borg provides three main benefits: it (1) hides the details of resource management and failure handling so its users can focus on application development instead; (2) operates with very high reliability and availability, and supports applications that do the same; and (3) lets us run workloads across tens of thousands of machines effectively. Borg is not the

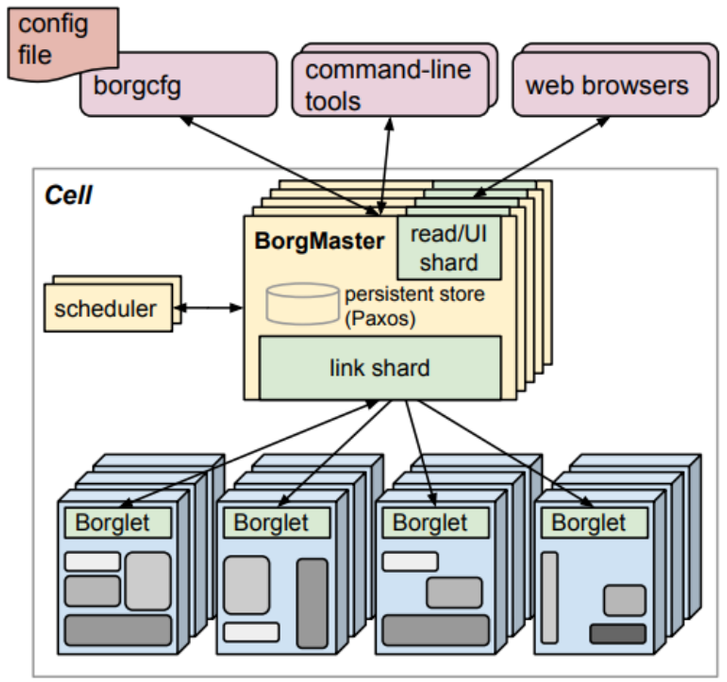


Figure 1: The high-level architecture of Borg. *Only a tiny fraction of the thousands of worker nodes are shown.*

cluding with a set of qualitative observations we have made from operating Borg in production for more than a decade.

2. The user perspective

Borg's users are Google developers and system administrators (site reliability engineers or SREs) that run Google's applications and services. Users submit their work to Borg in the form of *jobs*, each of which consists of one or more *tasks* that all run the same program (binary). Each job runs in one Borg *cell*, a set of machines that are managed as a unit. The remainder of this section describes the main features exposed in the user view of Borg.

Google Borg Monitoring (Borgmon)

- Each service and task at google exports metrics via HTTP.
- Easy to scrape via scripts or other automation tools.
- Google built a service called Sigma that would scrape, process and display detailed information on every job run.

Almost every task run under Borg contains a built-in HTTP server that publishes information about the health of the task and thousands of performance metrics (e.g., RPC latencies). Borg monitors the health-check URL and restarts tasks that do not respond promptly or return an HTTP error code. Other data is tracked by monitoring tools for dashboards and alerts on service level objective (SLO) violations.

A service called Sigma provides a web-based user interface (UI) through which a user can examine the state of all their jobs, a particular cell, or drill down to individual jobs and tasks to examine their resource behavior, detailed logs, execution history, and eventual fate. Our applications generate voluminous logs; these are automatically rotated to avoid running out of disk space, and preserved for a while after the task's exit to assist with debugging. If a job is not running Borg provides a "why pending?" annotation, together with guidance on how to modify the job's resource requests to better fit the cell. We publish guidelines for "conforming" resource shapes that are likely to schedule easily.

Borg records all job submissions and task events, as well as detailed per-task resource usage information in Infrastore, a scalable read-only data store with an interactive SQL-like interface via Dremel [61]. This data is used for usage-based charging, debugging job and system failures, and long-term capacity planning. It also provided the data for the Google cluster workload trace [80].

Prometheus

- Prometheus is an open source version of the Borgmon idea.
- Pull metric collection rather than push (push is available via a gateway).
- Easy and simple to scale by adding Prometheus servers.
- Uses a very simple exposition format.
- Designed for “right now” monitoring, with a default retention of 15 days
- Single go executable (or available via Docker)

Running with Docker

- Example running via Docker and Docker-compose
- Run a Prometheus endpoint as well as cAdvisor for monitoring local containers and Grafana for optional visualisation
- Persistence is via Docker volumes which extend beyond life of the container.
- Show how to monitor, containers, systems and legacy applications

```
version: '3'
services:
  cadvisor:
    image: google/cadvisor
    ports:
      - 8080:8080
    volumes:
      - /:/rootfs:ro
      - /var/run:/var/run:rw
      - /sys:/sys:ro
      - /var/lib/docker/:/var/lib/docker:ro

  prometheus:
    image: prom/prometheus
    ports:
      - 9090:9090
    volumes:
      - ./files/prometheus.yml:/etc/prometheus/prometheus.yml
      - prometheus_data:/prometheus

  grafana:
    image: grafana/grafana
    ports:
      - 3000:3000
    volumes:
      - grafana_data:/var/lib/grafana

volumes:
  prometheus_data:
  grafana_data:
~
~
~
~
~
~
~
~
~
~
~
~
docker-compose.yml                                    1,1
"docker-compose.yml" 30L, 568C
```

cAdvisor

Isolation

CPU

Shares 1024 *shares*

Allowed Cores 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31

Memory

Reservation unlimited

Limit unlimited

Swap Limit unlimited

Usage

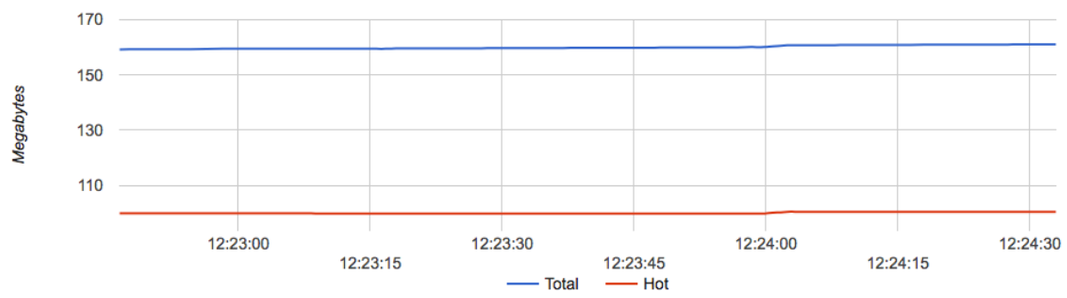
Overview

Processes

User	PID	PPID	Start Time	CPU %	MEM %	RSS	Virtual Size	Status	Running Time	Command
65534	3,488	3,470	11:38	1.20	0.10	110.20 MiB	147.57 MiB	Ssl	00:00:34	prometheus

Memory

Total Usage

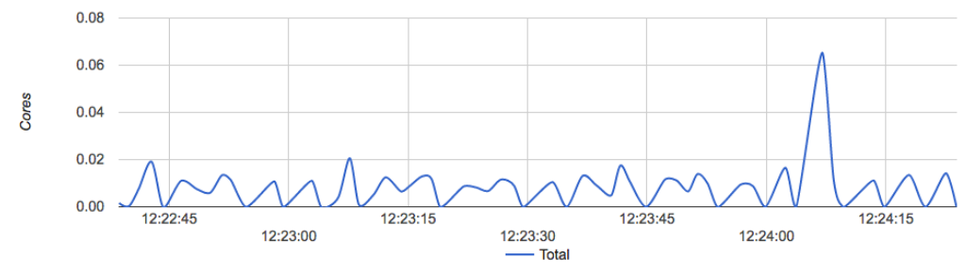


Usage Breakdown

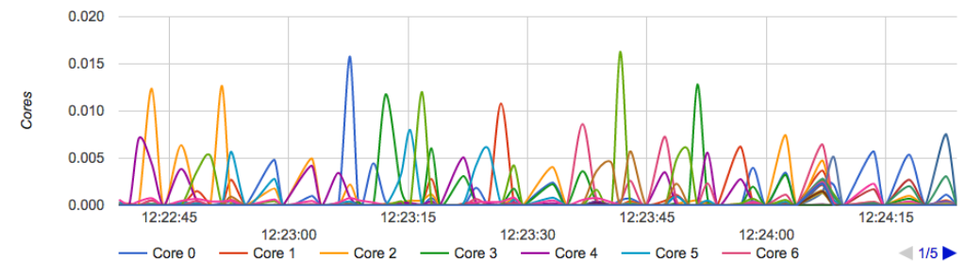
161.10 MiB / 62.64 GiB (0%)

CPU

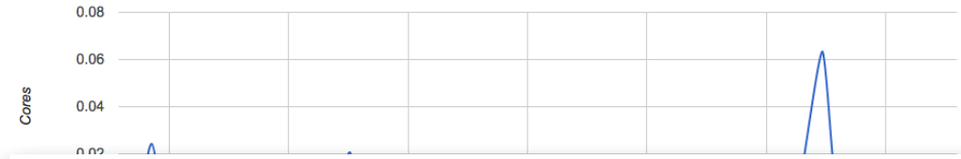
Total Usage



Usage per Core



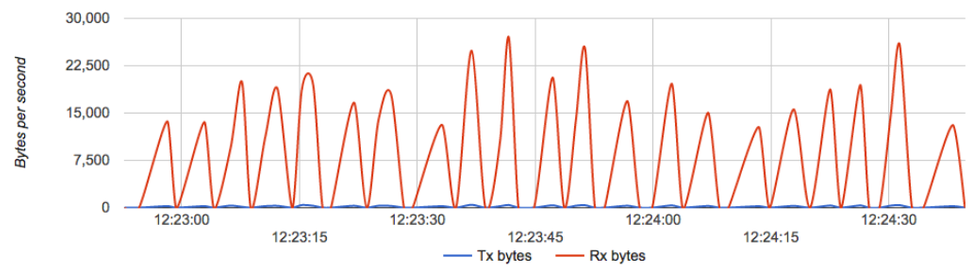
Usage Breakdown



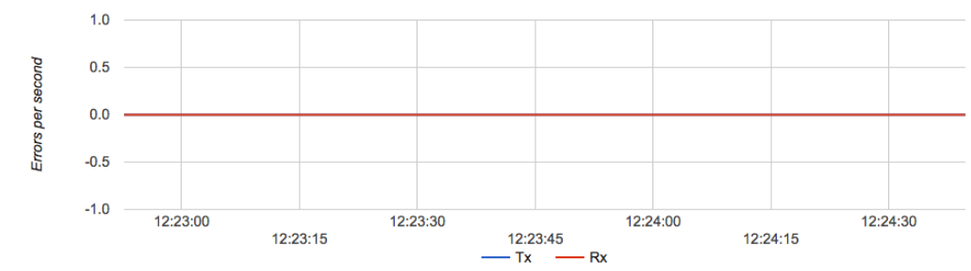
Network

Interface: **eth0**

Throughput



Errors



cAdvisor - Metrics (http://hostname:8080/metrics)

container_cpu_system_seconds_total
container_cpu_usage_seconds_total
container_cpu_user_seconds_total
container_fs_inodes_free
container_fs_inodes_total
container_fs_io_current
container_fs_io_time_seconds_total
container_fs_io_time_weighted_seconds_total
container_fs_limit_bytes
container_fs_read_seconds_total
container_fs_reads_merged_total
container_fs_reads_total
container_fs_sector_reads_total
container_fs_sector_writes_total
container_fs_usage_bytes
container_fs_write_seconds_total
container_fs_writes_merged_total
container_fs_writes_total
container_last_seen
container_memory_cache
container_memory_failcnt
container_memory_failures_total
container_memory_rss
container_memory_swap
container_memory_usage_bytes
container_memory_working_set_bytes
container_network_receive_bytes_total
container_network_receive_errors_total
container_network_receive_packets_dropped_total
container_network_receive_packets_total
container_network_transmit_bytes_total
container_network_transmit_errors_total
container_network_transmit_packets_dropped_total
container_network_transmit_packets_total
container_scrape_error
container_spec_cpu_period
container_spec_cpu_shares
container_spec_memory_limit_bytes
container_spec_memory_swap_limit_bytes
container_start_time_seconds
container_tasks_state

Prometheus Config

- Simply YAML file.
- This example we only specify scraping rules.
- Sources of machines to scrape can be:
    - azure
    - consul
    - dns
    - ec2
    - openstack
    - file
    - gce
    - kubernetes
    - marathon
    - nerve
    - serverset
    - triton
    - static

```yaml
global:
  scrape_interval: 15s

scrape_configs:
  - job_name: 'prometheus'
    scrape_interval: 5s
    static_configs:
      - targets: ['localhost:9090']

  - job_name: 'cadvisor'
    scrape_interval: 5s
    static_configs:
      - targets: ['cadvisor:8080']
```

```
files/prometheus.yml                                          1,1
"files/prometheus.yml" 13L, 260C
```

Prometheus

Prometheus Alerts Graph Status Help

Targets

cadvisor (1/1 up)

Endpoint	State	Labels	Last Scrape	Error
http://cadvisor:8080/metrics	UP	instance="cadvisor:8080"	615ms ago	

host (1/1 up)

Endpoint	State	Labels	Last Scrape	Error
http://kube003.beowulf.cluster:9100/metrics	UP	instance="kube003.beowulf.cluster:9100"	1.824s ago	

prometheus (1/1 up)

Endpoint	State	Labels	Last Scrape	Error
http://localhost:9090/metrics	UP	instance="localhost:9090"	4.389s ago	

Prometheus Alerts Graph Status Help

sum(rate(container_cpu_usage_seconds_total{container_label_com_docker_compose_config_hash=~".*"}[30s])) by (name)

Load time: 200ms
Resolution: 14s
Total time series: 3

Execute - insert metric at cursor -

Graph Console

1h Until Res. (s) stacked

0.1
0.05
0
11:45 12:00 12:15 12:30

{name="prometheus_prometheus_1"}
{name="prometheus_grafana_1"}
{name="prometheus_cadvisor_1"}

Remove Graph

sum(container_memory_usage_bytes{container_label_com_docker_compose_config_hash=~".*"}) by (name)

Load time: 94ms
Resolution: 14s
Total time series: 3

Execute - insert metric at cursor -

Graph Console

1h Until Res. (s) stacked

300M
200M
100M
0
11:45 12:00 12:15 12:30

{name="prometheus_prometheus_1"}
{name="prometheus_grafana_1"}
{name="prometheus_cadvisor_1"}

Remove Graph

sum(rate(container_network_receive_bytes_total{container_label_com_docker_compose_config_hash=~".*"}[30s])) by (name)

Load time: 143ms
Resolution: 14s
Total time series: 3

Execute - insert metric at cursor -

Graph Console

1h Until Res. (s) stacked

15k
10k

Demo Container Dashboard Last 1 hour

Container CPU usage

0.15
0.10
0.05
0
11:45 11:50 11:55 12:00 12:05 12:10 12:15 12:20 12:25 12:30 12:35 12:40

prometheus_cadvisor_1 prometheus_grafana_1 prometheus_prometheus_1

Container Memory Usage

334 MiB
286 MiB
238 MiB
191 MiB
143 MiB
95 MiB
48 MiB
0 B
11:45 11:50 11:55 12:00 12:05 12:10 12:15 12:20 12:25 12:30 12:35 12:40

prometheus_cadvisor_1 prometheus_grafana_1 prometheus_prometheus_1

Container Network (Recieve)

13 K
10 K
8 K
5 K
3 K
0
11:45 11:50 11:55 12:00 12:05 12:10 12:15 12:20 12:25 12:30 12:35 12:40

prometheus_cadvisor_1 prometheus_grafana_1 prometheus_prometheus_1

Container Network (Transmit)

400 K
300 K
200 K
100 K
0
11:45 11:50 11:55 12:00 12:05 12:10 12:15 12:20 12:25 12:30 12:35 12:40

prometheus_cadvisor_1 prometheus_grafana_1 prometheus_prometheus_1

Container Network (In, Out)

400 K
300 K
200 K
100 K
0
11:45 11:50 11:55 12:00 12:05 12:10 12:15 12:20 12:25 12:30 12:35 12:40

prometheus_cadvisor_1 (out) prometheus_grafana_1 (out) prometheus_prometheus_1 (out) prometheus_cadvisor_1 (in) prometheus_grafana_1 (in)
prometheus_prometheus_1 (in)

Prometheus vs Grafana

node_exporter

- For system monitoring, Prometheus proved node_exporter
- Single golang binary that gathers a myriad of performance counters.
- Simple to run as a service (via systemd, init.d needs some work).
- Ansible script shown here.

```
- hosts: demo
  tasks:
  - name: Copy node_exporter binary to target systems
    copy:
      src: files/node_exporter
      dest: /usr/sbin/node_exporter
      owner: root
      group: root
      mode: 0755

  - name: Copy node_exporter service file to target systems
    copy:
      src: files/node_exporter.service
      dest: /etc/systemd/system/node_exporter.service
      owner: root
      group: root
      mode: 0644

  - name: Copy node_exporter environment file to target systems
    copy:
      src: files/node_exporter_env
      dest: /etc/sysconfig/node_exporter_env
      owner: root
      group: root
      mode: 0644

  - name: Add node_exporter user
    user: name=node_exporter group=users shell=/sbin/nologin cre

  - name: Create textfile directory
    file:
      path: /var/lib/node_exporter/textfile_collector
      state: directory
      mode: 0755

  - name: Enable and Start the node_exporter service
    service:
      name: node_exporter
      enabled: yes
      state: started
~
~
~
setup.yml                                                    1,1
"setup.yml" 40L, 1016C
```

node_exporter

```
INFO[0000] Starting node_exporter (version=0.15.1, branch=HEAD, revision=ba5da2c29ae7f6209a88cb58676ba5ba029ad785)
INFO[0000] Build context (go=go1.9.2, user=root@b73d73f4fc5e, date=20171107-17:50:51)  source="node_exporter.go:44"
INFO[0000] No directory specified, see --collector.textfile.directory  source="textfile.go:57"
INFO[0000] Enabled collectors:                           source="node_exporter.go:50"
INFO[0000]  - cpu                                        source="node_exporter.go:52"
INFO[0000]  - filesystem                                 source="node_exporter.go:52"
INFO[0000]  - meminfo                                    source="node_exporter.go:52"
INFO[0000]  - conntrack                                  source="node_exporter.go:52"
INFO[0000]  - mdadm                                      source="node_exporter.go:52"
INFO[0000]  - hwmon                                      source="node_exporter.go:52"
INFO[0000]  - netdev                                     source="node_exporter.go:52"
INFO[0000]  - filefd                                     source="node_exporter.go:52"
INFO[0000]  - loadavg                                    source="node_exporter.go:52"
INFO[0000]  - ipvs                                       source="node_exporter.go:52"
INFO[0000]  - edac                                       source="node_exporter.go:52"
INFO[0000]  - infiniband                                 source="node_exporter.go:52"
INFO[0000]  - xfs                                        source="node_exporter.go:52"
INFO[0000]  - bcache                                     source="node_exporter.go:52"
INFO[0000]  - textfile                                   source="node_exporter.go:52"
INFO[0000]  - uname                                      source="node_exporter.go:52"
INFO[0000]  - time                                       source="node_exporter.go:52"
INFO[0000]  - diskstats                                  source="node_exporter.go:52"
INFO[0000]  - entropy                                    source="node_exporter.go:52"
INFO[0000]  - wifi                                       source="node_exporter.go:52"
INFO[0000]  - timex                                      source="node_exporter.go:52"
INFO[0000]  - sockstat                                   source="node_exporter.go:52"
INFO[0000]  - stat                                       source="node_exporter.go:52"
INFO[0000]  - netstat                                    source="node_exporter.go:52"
INFO[0000]  - zfs                                        source="node_exporter.go:52"
INFO[0000]  - arp                                        source="node_exporter.go:52"
INFO[0000]  - vmstat                                     source="node_exporter.go:52"
INFO[0000] Listening on :9100                            source="node_exporter.go:76"
```

node_arp_entries
node_boot_time
node_context_switches
node_cpu
node_cpu_core_throttles_total
node_cpu_frequency_hertz
node_cpu_frequency_max_hertz
node_cpu_frequency_min_hertz
node_cpu_package_throttles_total
node_disk_bytes_read
node_disk_bytes_written
node_disk_io_now
node_disk_io_time_ms
node_disk_io_time_weighted
node_disk_reads_completed
node_disk_reads_merged
node_disk_read_time_ms
node_disk_sectors_read
node_disk_sectors_written
node_disk_writes_completed
node_disk_writes_merged
node_disk_write_time_ms
node_entropy_available_bits
node_exporter_build_info
node_filefd_allocated
node_filefd_maximum
node_filesystem_avail
node_filesystem_device_error
node_filesystem_files
node_filesystem_files_free
node_filesystem_free
node_filesystem_readonly
node_filesystem_size
node_forks
node_hwmon_chip_names
node_hwmon_sensor_label
node_hwmon_temp_celsius
node_hwmon_temp_crit_alarm_celsius
node_hwmon_temp_crit_celsius
node_hwmon_temp_max_celsius
node_intr
node_load1
node_load15
node_load5
node_memory_Active
node_memory_Active_anon
node_memory_Active_file
node_memory_AnonHugePages
node_memory_AnonPages
node_memory_Bounce
node_memory_Buffers
node_memory_Cached
node_memory_CmaFree
node_memory_CmaTotal
node_memory_CommitLimit
node_memory_Committed_AS
node_memory_DirectMap1G
node_memory_DirectMap2M
node_memory_DirectMap4k
node_memory_Dirty
node_memory_HardwareCorrupted
node_memory_HugePages_Free
node_memory_Hugepagesize
node_memory_HugePages_Rsvd
node_memory_HugePages_Surp
node_memory_HugePages_Total
node_memory_Inactive
node_memory_Inactive_anon
node_memory_Inactive_file
node_memory_KernelStack
node_memory_Mapped
node_memory_MemAvailable
node_memory_MemFree
node_memory_MemTotal
node_memory_Mlocked
node_memory_NFS_Unstable
node_memory_PageTables
node_memory_Shmem
node_memory_ShmemHugePages
node_memory_ShmemPmdMapped
node_memory_Slab
node_memory_SReclaimable
node_memory_SUnreclaim
node_memory_SwapCached
node_memory_SwapFree
node_memory_SwapTotal
node_memory_Unevictable
node_memory_VmallocChunk
node_memory_VmallocTotal
node_memory_VmallocUsed
node_memory_Writeback
node_memory_WritebackTmp
node_netstat_Icmp6_InCsumErrors
node_netstat_Icmp6_InDestUnreachs
node_netstat_Icmp6_InEchoReplies
node_netstat_Icmp6_InEchos
node_netstat_Icmp6_InErrors
node_netstat_Icmp6_InGroupMembQueries
node_netstat_Icmp6_InGroupMembReductions
node_netstat_Icmp6_InGroupMembResponses
node_netstat_Icmp6_InMLDv2Reports
node_netstat_Icmp6_InMsgs
node_netstat_Icmp6_InNeighborAdvertisements
node_netstat_Icmp6_InNeighborSolicits
node_netstat_Icmp6_InParmProblems
node_netstat_Icmp6_InPktTooBigs
node_netstat_Icmp6_InRedirects
node_netstat_Icmp6_InRouterAdvertisements
node_netstat_Icmp6_InRouterSolicits
node_netstat_Icmp6_InTimeExcds
node_netstat_Icmp6_InType136
node_netstat_Icmp6_OutDestUnreachs
node_netstat_Icmp6_OutEchoReplies
node_netstat_Icmp6_OutEchos
node_netstat_Icmp6_OutErrors
node_netstat_Icmp6_OutGroupMembQueries
node_netstat_Icmp6_OutGroupMembReductions
node_netstat_Icmp6_OutGroupMembResponses
node_netstat_Icmp6_OutMLDv2Reports
node_netstat_Icmp6_OutMsgs
node_netstat_Icmp6_OutNeighborAdvertisements
node_netstat_Icmp6_OutNeighborSolicits
node_netstat_Icmp6_OutParmProblems
node_netstat_Icmp6_OutPktTooBigs
node_netstat_Icmp6_OutRedirects
node_netstat_Icmp6_OutRouterAdvertisements
node_netstat_Icmp6_OutRouterSolicits
node_netstat_Icmp6_OutTimeExcds
node_netstat_Icmp6_OutType133
node_netstat_Icmp6_OutType135
node_netstat_Icmp6_OutType143
node_netstat_Icmp_InAddrMaskReps
node_netstat_Icmp_InAddrMasks
node_netstat_Icmp_InCsumErrors
node_netstat_Icmp_InDestUnreachs
node_netstat_Icmp_InEchoReps
node_netstat_Icmp_InEchos
node_netstat_Icmp_InErrors
node_netstat_Icmp_InMsgs
node_netstat_Icmp_InParmProbs
node_netstat_Icmp_InRedirects
node_netstat_Icmp_InSrcQuenchs
node_netstat_Icmp_InTimeExcds
node_netstat_Icmp_InTimestampReps
node_netstat_Icmp_InTimestamps
node_netstat_IcmpMsg_InType3
node_netstat_IcmpMsg_InType8
node_netstat_IcmpMsg_OutType0
node_netstat_IcmpMsg_OutType3
node_netstat_Icmp_OutAddrMaskReps
node_netstat_Icmp_OutAddrMasks
node_netstat_Icmp_OutDestUnreachs
node_netstat_Icmp_OutEchoReps
node_netstat_Icmp_OutEchos
node_netstat_Icmp_OutErrors
node_netstat_Icmp_OutMsgs
node_netstat_Icmp_OutParmProbs
node_netstat_Icmp_OutRedirects
node_netstat_Icmp_OutSrcQuenchs
node_netstat_Icmp_OutTimeExcds
node_netstat_Icmp_OutTimestampReps
node_netstat_Icmp_OutTimestamps
node_netstat_Ip6_FragCreates
node_netstat_Ip6_FragFails
node_netstat_Ip6_FragOKs
node_netstat_Ip6_InAddrErrors
node_netstat_Ip6_InBcastOctets
node_netstat_Ip6_InCEPkts
node_netstat_Ip6_InDelivers
node_netstat_Ip6_InDiscards
node_netstat_Ip6_InECT0Pkts
node_netstat_Ip6_InECT1Pkts
node_netstat_Ip6_InHdrErrors
node_netstat_Ip6_InMcastOctets
node_netstat_Ip6_InMcastPkts
node_netstat_Ip6_InNoECTPkts
node_netstat_Ip6_InNoRoutes
node_netstat_Ip6_InOctets
node_netstat_Ip6_InReceives
node_netstat_Ip6_InTooBigErrors
node_netstat_Ip6_InTruncatedPkts
node_netstat_Ip6_InUnknownProtos
node_netstat_Ip6_OutBcastOctets
node_netstat_Ip6_OutDiscards
node_netstat_Ip6_OutForwDatagrams
node_netstat_Ip6_OutMcastOctets
node_netstat_Ip6_OutMcastPkts
node_netstat_Ip6_OutNoRoutes
node_netstat_Ip6_OutOctets
node_netstat_Ip6_OutRequests
node_netstat_Ip6_ReasmFails
node_netstat_Ip6_ReasmOKs
node_netstat_Ip6_ReasmReqds
node_netstat_Ip6_ReasmTimeout
node_netstat_Ip_DefaultTTL
node_netstat_IpExt_InBcastOctets
node_netstat_IpExt_InBcastPkts
node_netstat_IpExt_InCEPkts
node_netstat_IpExt_InCsumErrors
node_netstat_IpExt_InECT0Pkts
node_netstat_IpExt_InECT1Pkts
node_netstat_IpExt_InMcastOctets
node_netstat_IpExt_InMcastPkts
node_netstat_IpExt_InNoECTPkts
node_netstat_IpExt_InNoRoutes
node_netstat_IpExt_InOctets
node_netstat_IpExt_InTruncatedPkts
node_netstat_IpExt_OutBcastOctets
node_netstat_IpExt_OutBcastPkts
node_netstat_IpExt_OutMcastOctets
node_netstat_IpExt_OutMcastPkts
node_netstat_IpExt_OutOctets
node_netstat_Ip_Forwarding
node_netstat_Ip_ForwDatagrams
node_netstat_Ip_FragCreates
node_netstat_Ip_FragFails
node_netstat_Ip_FragOKs
node_netstat_Ip_InAddrErrors
node_netstat_Ip_InDelivers
node_netstat_Ip_InDiscards
node_netstat_Ip_InHdrErrors
node_netstat_Ip_InReceives
node_netstat_Ip_InUnknownProtos
node_netstat_Ip_OutDiscards
node_netstat_Ip_OutNoRoutes
node_netstat_Ip_OutRequests
node_netstat_Ip_ReasmFails
node_netstat_Ip_ReasmOKs
node_netstat_Ip_ReasmReqds
node_netstat_Ip_ReasmTimeout
node_netstat_Tcp_ActiveOpens
node_netstat_Tcp_AttemptFails
node_netstat_Tcp_CurrEstab
node_netstat_Tcp_EstabResets
node_netstat_TcpExt_ArpFilter
node_netstat_TcpExt_BusyPollRxPackets
node_netstat_TcpExt_DelayedACKLocked
node_netstat_TcpExt_DelayedACKLost
node_netstat_TcpExt_DelayedACKs
node_netstat_TcpExt_EmbryonicRsts
node_netstat_TcpExt_IPReversePathFilter
node_netstat_TcpExt_ListenDrops
node_netstat_TcpExt_ListenOverflows
node_netstat_TcpExt_LockDroppedIcmps
node_netstat_TcpExt_OfoPruned
node_netstat_TcpExt_OutOfWindowIcmps
node_netstat_TcpExt_PAWSActive
node_netstat_TcpExt_PAWSEstab
node_netstat_TcpExt_PFMemallocDrop
node_netstat_TcpExt_PruneCalled
node_netstat_TcpExt_RcvPruned
node_netstat_TcpExt_SyncookiesFailed
node_netstat_TcpExt_SyncookiesRecv
node_netstat_TcpExt_SyncookiesSent
node_netstat_TcpExt_TCPAbortFailed
node_netstat_TcpExt_TCPAbortOnClose
node_netstat_TcpExt_TCPAbortOnData
node_netstat_TcpExt_TCPAbortOnLinger
node_netstat_TcpExt_TCPAbortOnMemory
node_netstat_TcpExt_TCPAbortOnTimeout
node_netstat_TcpExt_TCPACKSkippedChallenge
node_netstat_TcpExt_TCPACKSkippedFinWait2
node_netstat_TcpExt_TCPACKSkippedPAWS
node_netstat_TcpExt_TCPACKSkippedSeq
node_netstat_TcpExt_TCPACKSkippedSynRecv
node_netstat_TcpExt_TCPACKSkippedTimeWait
node_netstat_TcpExt_TCPAutoCorking
node_netstat_TcpExt_TCPBacklogDrop
node_netstat_TcpExt_TCPChallengeACK
node_netstat_TcpExt_TCPDeferAcceptDrop
node_netstat_TcpExt_TCPDSACKIgnoredNoUnd
node_netstat_TcpExt_TCPDSACKIgnoredOld
node_netstat_TcpExt_TCPDSACKOfoRecv
node_netstat_TcpExt_TCPDSACKOfoSent
node_netstat_TcpExt_TCPDSACKOldSent
node_netstat_TcpExt_TCPDSACKRecv
node_netstat_TcpExt_TCPDSACKUndo
node_netstat_TcpExt_TCPFACKReorder
node_netstat_TcpExt_TCPFastOpenActive
node_netstat_TcpExt_TCPFastOpenActiveFail
node_netstat_TcpExt_TCPFastOpenBlackhole
node_netstat_TcpExt_TCPFastOpenCookieReqd
node_netstat_TcpExt_TCPFastOpenListenOverfl
node_netstat_TcpExt_TCPFastOpenPassive
node_netstat_TcpExt_TCPFastOpenPassiveFail
node_netstat_TcpExt_TCPFastRetrans
node_netstat_TcpExt_TCPFromZeroWindowAdv
node_netstat_TcpExt_TCPFullUndo
node_netstat_TcpExt_TCPHPAcks
node_netstat_TcpExt_TCPHPHits
node_netstat_TcpExt_TCPHystartDelayCwnd
node_netstat_TcpExt_TCPHystartDelayDetect
node_netstat_TcpExt_TCPHystartTrainCwnd
node_netstat_TcpExt_TCPHystartTrainDetect
node_netstat_TcpExt_TCPKeepAlive
node_netstat_TcpExt_TCPLossFailures
node_netstat_TcpExt_TCPLossProbeRecovery
node_netstat_TcpExt_TCPLossProbes
node_netstat_TcpExt_TCPLossUndo
node_netstat_TcpExt_TCPLostRetransmit
node_netstat_TcpExt_TCPMD5Failure
node_netstat_TcpExt_TCPMD5NotFound
node_netstat_TcpExt_TCPMD5Unexpected
node_netstat_TcpExt_TCPMemoryPressures
node_netstat_TcpExt_TCPMemoryPressuresChr
node_netstat_TcpExt_TCPMinTTLDrop
node_netstat_TcpExt_TCPMTUPFail
node_netstat_TcpExt_TCPMTUPSuccess
node_netstat_TcpExt_TCPOFODrop
node_netstat_TcpExt_TCPOFOMerge
node_netstat_TcpExt_TCPOFOQueue
node_netstat_TcpExt_TCPOrigDataSent
node_netstat_TcpExt_TCPPartialUndo
node_netstat_TcpExt_TCPPureAcks
node_netstat_TcpExt_TCPRcvCoalesce

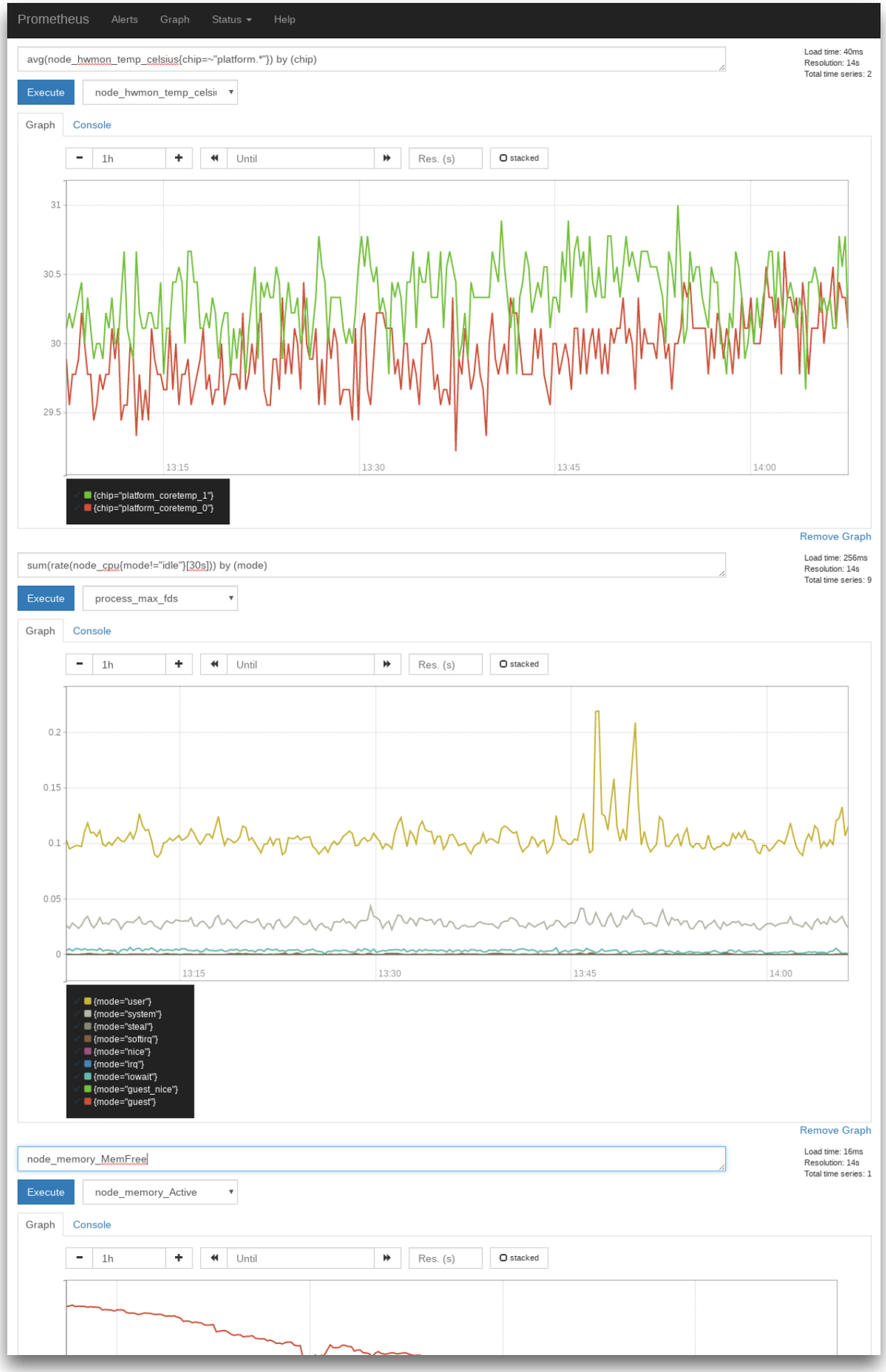
node_netstat_TcpExt_TCPRcvCollapsed
node_netstat_TcpExt_TCPRenoFailures
node_netstat_TcpExt_TCPRenoRecovery
node_netstat_TcpExt_TCPRenoRecoveryFail
node_netstat_TcpExt_TCPRenoReorder
node_netstat_TcpExt_TCPReqQFullDoCookies
node_netstat_TcpExt_TCPReqQFullDrop
node_netstat_TcpExt_TCPRetransFail
node_netstat_TcpExt_TCPSACKDiscard
node_netstat_TcpExt_TCPSackFailures
node_netstat_TcpExt_TCPSackMerged
node_netstat_TcpExt_TCPSackRecovery
node_netstat_TcpExt_TCPSackRecoveryFail
node_netstat_TcpExt_TCPSACKReneging
node_netstat_TcpExt_TCPSACKReorder
node_netstat_TcpExt_TCPSackShifted
node_netstat_TcpExt_TCPSackShiftFallback
node_netstat_TcpExt_TCPSlowStartRetrans
node_netstat_TcpExt_TCPSpuriousRTOs
node_netstat_TcpExt_TCPSpuriousRtxHostQueues
node_netstat_TcpExt_TCPSYNChallenge
node_netstat_TcpExt_TCPSynRetrans
node_netstat_TcpExt_TCPTimeouts
node_netstat_TcpExt_TCPTimeWaitOverflow
node_netstat_TcpExt_TCPToZeroWindowAdv
node_netstat_TcpExt_TCPTSReorder
node_netstat_TcpExt_TCPWantZeroWindowAdv
node_netstat_TcpExt_TCPWinProbe
node_netstat_TcpExt_TW
node_netstat_TcpExt_TWKilled
node_netstat_TcpExt_TWRecycled
node_netstat_Tcp_InCsumErrors
node_netstat_Tcp_InErrs
node_netstat_Tcp_InSegs
node_netstat_Tcp_MaxConn
node_netstat_Tcp_OutRsts
node_netstat_Tcp_OutSegs
node_netstat_Tcp_PassiveOpens
node_netstat_Tcp_RetransSegs
node_netstat_Tcp_RtoAlgorithm
node_netstat_Tcp_RtoMax
node_netstat_Tcp_RtoMin
node_netstat_Udp6_IgnoredMulti
node_netstat_Udp6_InCsumErrors
node_netstat_Udp6_InDatagrams
node_netstat_Udp6_InErrors
node_netstat_Udp6_NoPorts
node_netstat_Udp6_OutDatagrams
node_netstat_Udp6_RcvbufErrors
node_netstat_Udp6_SndbufErrors
node_netstat_Udp_IgnoredMulti
node_netstat_Udp_InCsumErrors
node_netstat_Udp_InDatagrams
node_netstat_Udp_InErrors
node_netstat_UdpLite6_InCsumErrors
node_netstat_UdpLite6_InDatagrams
node_netstat_UdpLite6_InErrors
node_netstat_UdpLite6_NoPorts
node_netstat_UdpLite6_OutDatagrams
node_netstat_UdpLite6_RcvbufErrors
node_netstat_UdpLite6_SndbufErrors
node_netstat_UdpLite_IgnoredMulti
node_netstat_UdpLite_InCsumErrors
node_netstat_UdpLite_InDatagrams
node_netstat_UdpLite_InErrors
node_netstat_UdpLite_NoPorts
node_netstat_UdpLite_OutDatagrams
node_netstat_UdpLite_RcvbufErrors
node_netstat_UdpLite_SndbufErrors
node_netstat_Udp_NoPorts
node_netstat_Udp_OutDatagrams
node_netstat_Udp_RcvbufErrors
node_netstat_Udp_SndbufErrors
node_network_receive_bytes
node_network_receive_compressed
node_network_receive_drop
node_network_receive_errs
node_network_receive_fifo
node_network_receive_frame
node_network_receive_multicast
node_network_receive_packets
node_network_transmit_bytes
node_network_transmit_compressed
node_network_transmit_drop
node_network_transmit_errs
node_network_transmit_fifo
node_network_transmit_frame
node_network_transmit_multicast
node_network_transmit_packets
node_nf_conntrack_entries
node_nf_conntrack_entries_limit
node_procs_blocked
node_procs_running
node_scrape_collector_duration_seconds
node_scrape_collector_success
node_sockstat_FRAG_inuse
node_sockstat_FRAG_memory
node_sockstat_RAW_inuse
node_sockstat_sockets_used
node_sockstat_TCP_alloc
node_sockstat_TCP_inuse
node_sockstat_TCP_mem
node_sockstat_TCP_mem_bytes
node_sockstat_TCP_orphan
node_sockstat_TCP_tw
node_sockstat_UDP_inuse
node_sockstat_UDPLITE_inuse
node_sockstat_UDP_mem
node_sockstat_UDP_mem_bytes
node_time
node_timex_estimated_error_seconds
node_timex_frequency_adjustment
node_timex_loop_time_constant
node_timex_maxerror_seconds
node_timex_offset_seconds
node_timex_pps_calibration_count
node_timex_pps_error_count
node_timex_pps_frequency
node_timex_pps_jitter_count
node_timex_pps_jitter_seconds
node_timex_pps_shift_seconds
node_timex_pps_stability
node_timex_pps_stability_exceeded_count
node_timex_status
node_timex_sync_status
node_timex_tai_offset
node_timex_tick_seconds
node_vmstat_allocstall_dma
node_vmstat_allocstall_dma32
node_vmstat_allocstall_movable
node_vmstat_allocstall_normal
node_vmstat_balloon_deflate
node_vmstat_balloon_inflate
node_vmstat_balloon_migrate
node_vmstat_compact_daemon_free_scanned
node_vmstat_compact_daemon_migrate_scanned
node_vmstat_compact_daemon_wake
node_vmstat_compact_fail
node_vmstat_compact_free_scanned
node_vmstat_compact_isolated
node_vmstat_compact_migrate_scanned
node_vmstat_compact_stall
node_vmstat_compact_success
node_vmstat_drop_pagecache
node_vmstat_drop_slab
node_vmstat_htlb_buddy_alloc_fail
node_vmstat_htlb_buddy_alloc_success
node_vmstat_kswapd_high_wmark_hit_quickly
node_vmstat_kswapd_inodesteal
node_vmstat_kswapd_low_wmark_hit_quickly
node_vmstat_nr_active_anon
node_vmstat_nr_active_file
node_vmstat_nr_anon_pages
node_vmstat_nr_anon_transparent_hugepages
node_vmstat_nr_bounce
node_vmstat_nr_dirtied
node_vmstat_nr_dirty
node_vmstat_nr_dirty_background_threshold
node_vmstat_nr_dirty_threshold
node_vmstat_nr_file_pages
node_vmstat_nr_free_cma
node_vmstat_nr_free_pages
node_vmstat_nr_inactive_anon
node_vmstat_nr_inactive_file
node_vmstat_nr_isolated_anon
node_vmstat_nr_isolated_file
node_vmstat_nr_kernel_stack
node_vmstat_nr_mapped
node_vmstat_nr_mlock
node_vmstat_nr_page_table_pages
node_vmstat_nr_shmem
node_vmstat_nr_shmem_hugepages
node_vmstat_nr_shmem_pmdmapped
node_vmstat_nr_slab_reclaimable
node_vmstat_nr_slab_unreclaimable
node_vmstat_nr_unevictable
node_vmstat_nr_unstable
node_vmstat_nr_vmscan_immediate_reclaim
node_vmstat_nr_vmscan_write
node_vmstat_nr_writeback
node_vmstat_nr_writeback_temp
node_vmstat_nr_written
node_vmstat_nr_zone_active_anon
node_vmstat_nr_zone_active_file
node_vmstat_nr_zone_inactive_anon
node_vmstat_nr_zone_inactive_file
node_vmstat_nr_zone_unevictable
node_vmstat_nr_zone_write_pending
node_vmstat_nr_zspages
node_vmstat_numa_foreign
node_vmstat_numa_hint_faults
node_vmstat_numa_hint_faults_local
node_vmstat_numa_hit
node_vmstat_numa_huge_pte_updates
node_vmstat_numa_interleave
node_vmstat_numa_local
node_vmstat_numa_miss
node_vmstat_numa_other
node_vmstat_numa_pages_migrated
node_vmstat_numa_pte_updates
node_vmstat_oom_kill
node_vmstat_pageoutrun
node_vmstat_pgactivate
node_vmstat_pgalloc_dma
node_vmstat_pgalloc_dma32
node_vmstat_pgalloc_movable
node_vmstat_pgalloc_normal
node_vmstat_pgdeactivate
node_vmstat_pgfault
node_vmstat_pgfree
node_vmstat_pginodesteal
node_vmstat_pglazyfree
node_vmstat_pglazyfreed
node_vmstat_pgmajfault
node_vmstat_pgmigrate_fail
node_vmstat_pgmigrate_success
node_vmstat_pgpgin
node_vmstat_pgpgout
node_vmstat_pgrefill
node_vmstat_pgrotated
node_vmstat_pgscan_direct
node_vmstat_pgscan_direct_throttle
node_vmstat_pgscan_kswapd
node_vmstat_pgskip_dma
node_vmstat_pgskip_dma32
node_vmstat_pgskip_movable
node_vmstat_pgskip_normal
node_vmstat_pgsteal_direct
node_vmstat_pgsteal_kswapd
node_vmstat_pswpin
node_vmstat_pswpout
node_vmstat_slabs_scanned
node_vmstat_swap_ra
node_vmstat_swap_ra_hit
node_vmstat_thp_collapse_alloc
node_vmstat_thp_collapse_alloc_failed
node_vmstat_thp_deferred_split_page
node_vmstat_thp_fault_alloc
node_vmstat_thp_fault_fallback
node_vmstat_thp_file_alloc
node_vmstat_thp_file_mapped
node_vmstat_thp_split_page
node_vmstat_thp_split_page_failed
node_vmstat_thp_split_pmd
node_vmstat_thp_split_pud
node_vmstat_thp_swpout
node_vmstat_thp_swpout_fallback
node_vmstat_thp_zero_page_alloc
node_vmstat_thp_zero_page_alloc_failed
node_vmstat_unevictable_pgs_cleared
node_vmstat_unevictable_pgs_culled
node_vmstat_unevictable_pgs_mlocked
node_vmstat_unevictable_pgs_m
node_vmstat_unevictable_pgs_re
node_vmstat_unevictable_pgs_s
node_vmstat_unevictable_pgs_st
node_vmstat_workingset_activat
node_vmstat_workingset_nodere
node_vmstat_workingset_refault
node_vmstat_zone_reclaim_failed

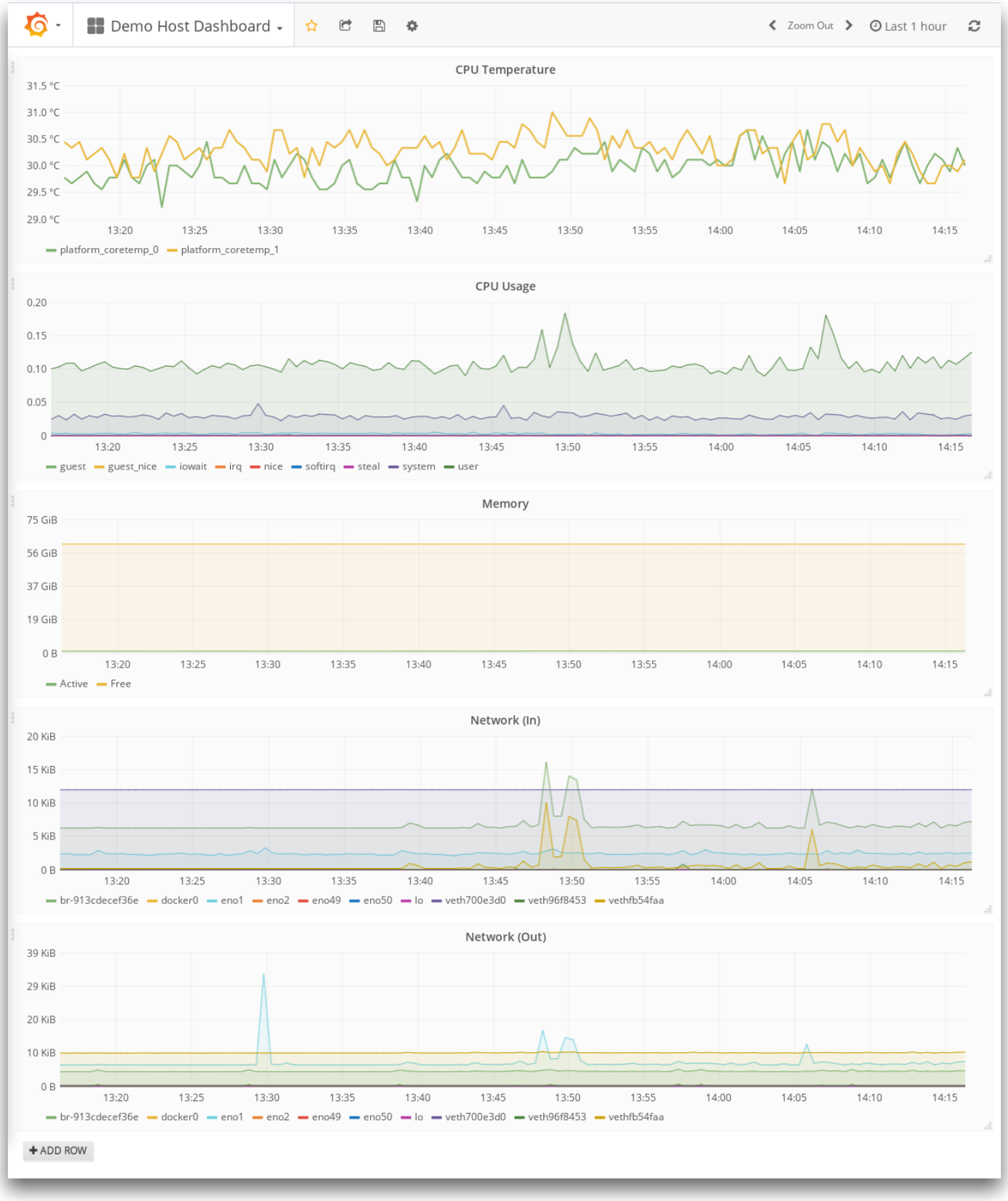
Prometheus Config

- We can add system monitoring by adding a simple rule to our prometheus config file.
- Adds another job, or we could run a second prometheus instance that only gathered system metrics.
- Easy to scale to any number of hosts by splitting up jobs across different Prometheus servers.

```yaml
global:
  scrape_interval: 15s

scrape_configs:
  - job_name: 'prometheus'
    scrape_interval: 5s
    static_configs:
      - targets: ['localhost:9090']

  - job_name: 'cadvisor'
    scrape_interval: 5s
    static_configs:
      - targets: ['cadvisor:8080']

  - job_name: 'host'
    scrape_interval: 5s
    static_configs:
      - targets: ['kube003.beowulf.cluster:9100']
```

```
prometheus.yml                                                   1,1
"prometheus.yml" 18L, 376C
```





Prometheus vs Grafana

Application and Legacy Monitoring

```
svr009:~# condor_ls --vo
Owner                Idle     Running     Removed  Completed        Held   Transfer  Suspended
gridpp054             589           1           0          0           0          0          0
ilc059                311           0           0          0           1          0          0
nast038               676           0           0          0           0          0          0
ops061                  0           1           0          0           0          0          0
pheno016             5105          74           0          0           3          0          0
pilatlas024            37          67           0          0          21          0          0
pilcms008             152          75           0          0          11          0          0
pilcms011               8           0           0          0           0          0          0
pillhcb002            195         789           0          0           1          0          0
pillhcb007              3           0           0          0           0          0          0
prdatlas006           567         280           0          0           8          0          0
prdatlas072             0           9           0          0           4          0          0
prdlhcb009              1           0           0          0           0          0          0
sgmcms                  8           0           0          0           0          0          0
svr009:~# _
```

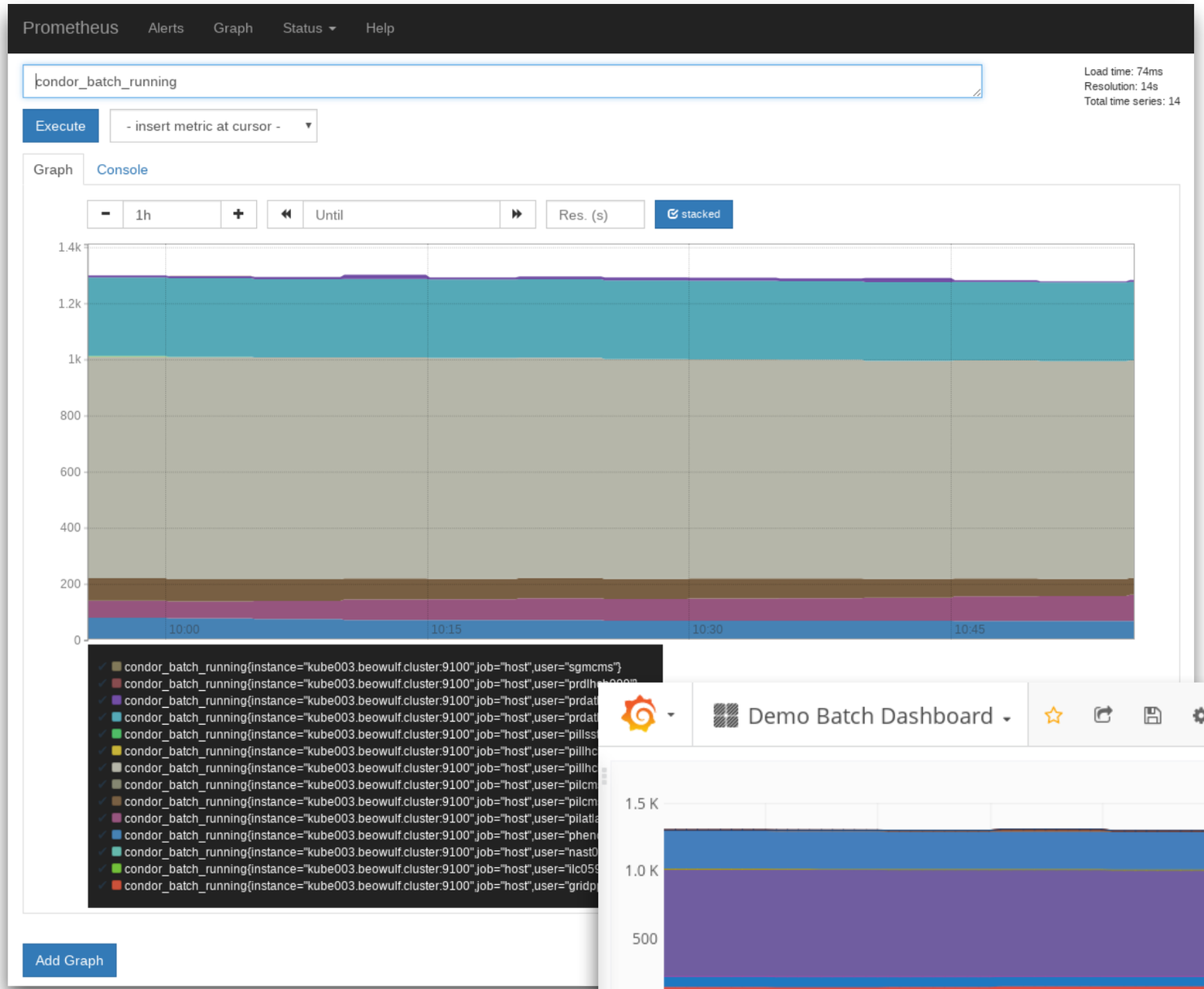
textfile plugin

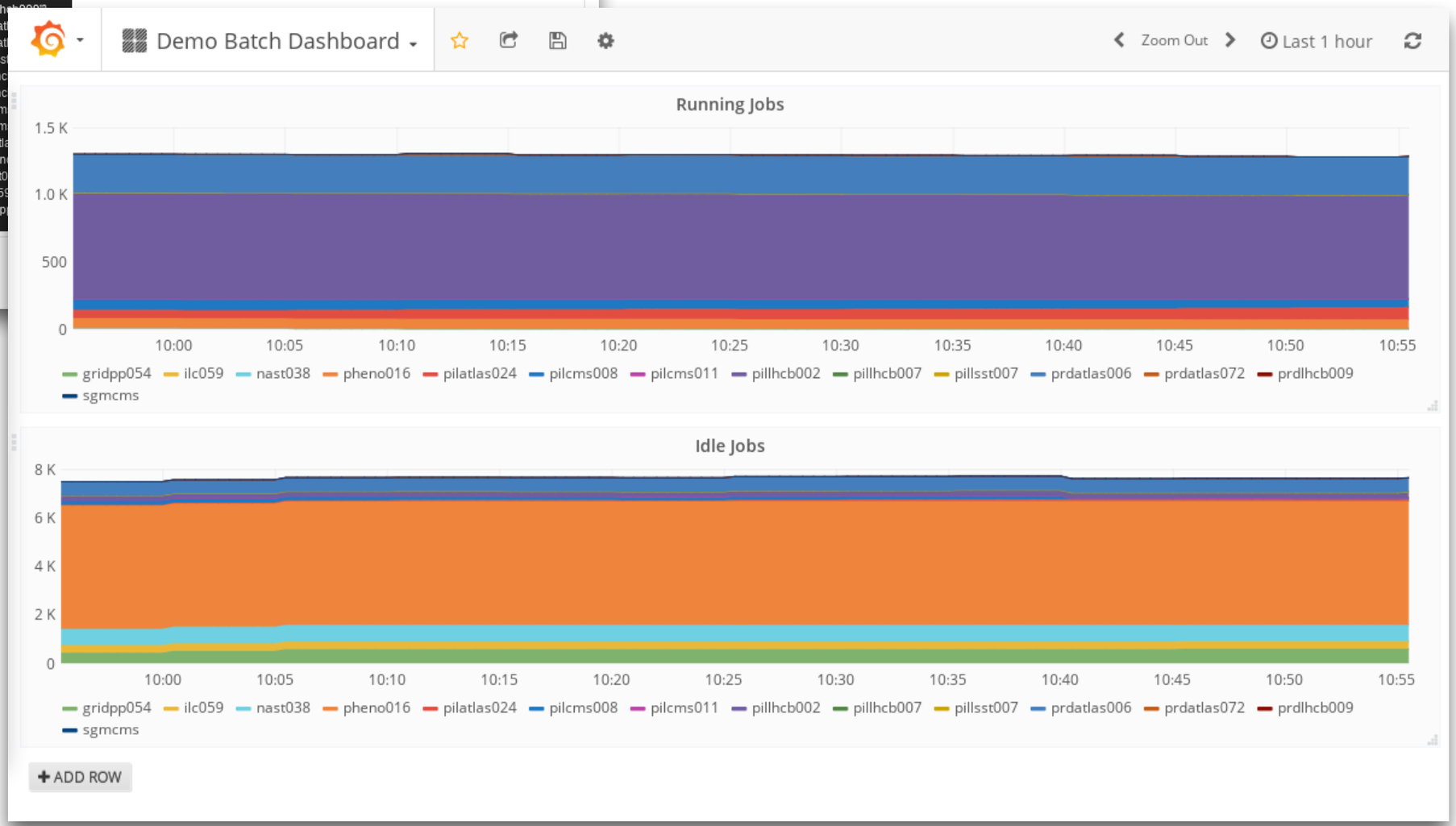
- It's simple to add arbitrary metrics using the textile plugin to node_exporter.
- We can specify a directory and node_exporter will parse any files with a *.prom extension looking for metrics.
- Example parses our condor_ls tool and publishes running jobs.
- Exposition format can be found at: https://prometheus.io/docs/instrumenting/exposition_formats/
- Metric types:
  - Counter
  - Gauge
  - Histogram
  - Summary
  - Untyped

```
#!/bin/bash

DATA=$(ssh svr009 "condor_ls --vo" | grep -v Owner)
PROMFILE="/var/lib/node_exporter/textfile_collector/htcondor.prom"

echo "# Condor Batch Statistics" > $PROMFILE

IFS=$'\n'
for i in $DATA; do


  USER=$(echo $i | awk '{print $1}')
  IDLE=$(echo $i | awk '{print $2}')
  RUNNING=$(echo $i | awk '{print $3}')

  echo "condor_batch_running{user=\"$USER\"} $RUNNING" >> $PROMFILE
  echo "condor_batch_idle{user=\"$USER\"} $IDLE" >> $PROMFILE
done
~
```

```
[root@kube003 ~]# cat /var/lib/node_exporter/textfile_collector/htcondor
# Condor Batch Statistics
condor_batch_running{user="gridpp054"} 0
condor_batch_idle{user="gridpp054"} 589
condor_batch_running{user="ilc059"} 0
condor_batch_idle{user="ilc059"} 311
condor_batch_running{user="nast038"} 0
condor_batch_idle{user="nast038"} 676
condor_batch_running{user="pheno016"} 73
condor_batch_idle{user="pheno016"} 5105
condor_batch_running{user="pilatlas024"} 73
condor_batch_idle{user="pilatlas024"} 40
condor_batch_running{user="pilcms008"} 75
condor_batch_idle{user="pilcms008"} 147
condor_batch_running{user="pilcms011"} 0
condor_batch_idle{user="pilcms011"} 8
condor_batch_running{user="pillhcb002"} 788
condor_batch_idle{user="pillhcb002"} 195
condor_batch_running{user="pillhcb007"} 0
condor_batch_idle{user="pillhcb007"} 4
condor_batch_running{user="prdatlas006"} 280
condor_batch_idle{user="prdatlas006"} 572
condor_batch_running{user="prdatlas072"} 15
condor_batch_idle{user="prdatlas072"} 0
condor_batch_running{user="prdlhcb009"} 0
condor_batch_idle{user="prdlhcb009"} 2
condor_batch_running{user="sgmcms"} 0
```

Prometheus vs Grafana





Monitoring VAC

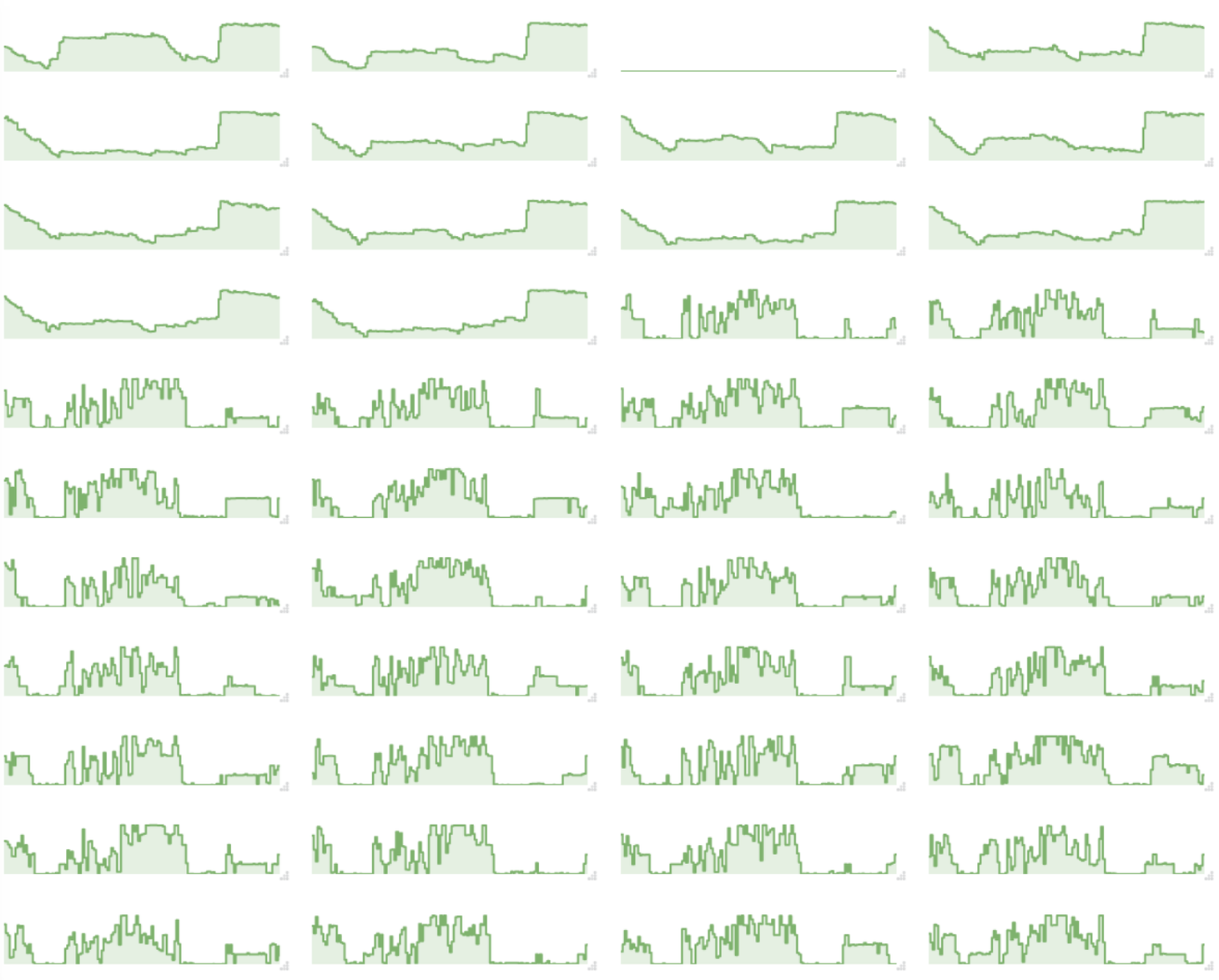
- Wanted to monitor what VAC nodes where actually doing.
- vacmon.gridpp.ac.uk
- Fine for occupancy etc, but couldn't see low level metrics.
- Run node_exporter on all machines, scrape via Prometheus.
- Initially attempt to replicate vacmon at local level.

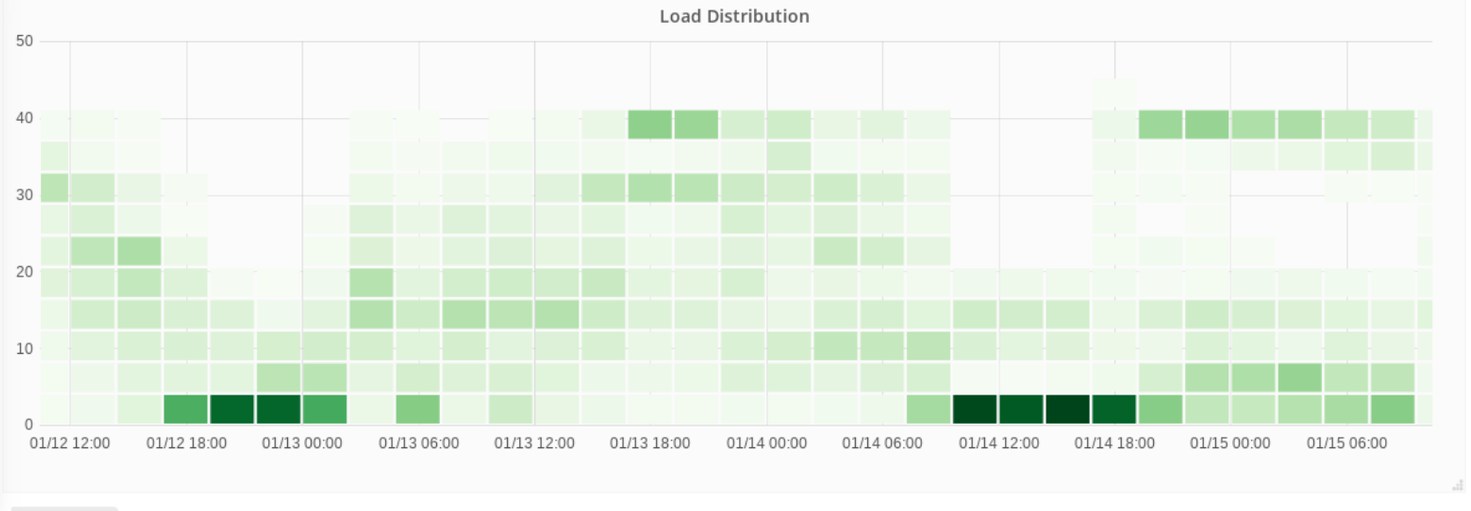
```
global:
  scrape_interval: 15s

scrape_configs:
  - job_name: 'prometheus'
    scrape_interval: 5s
    static_configs:
      - targets: ['localhost:9090']

  - job_name: 'vac'
    scrape_interval: 5s
    static_configs:
      - targets:
          - vac001.beowulf.cluster:9100
          - vac002.beowulf.cluster:9100
          - vac003.beowulf.cluster:9100
          - vac004.beowulf.cluster:9100
          - vac005.beowulf.cluster:9100
          - vac006.beowulf.cluster:9100
          - vac007.beowulf.cluster:9100
          - vac008.beowulf.cluster:9100
          - vac009.beowulf.cluster:9100
          - vac010.beowulf.cluster:9100
          - vac011.beowulf.cluster:9100
          - vac012.beowulf.cluster:9100
          - vac013.beowulf.cluster:9100
          - vac014.beowulf.cluster:9100
          - vac015.beowulf.cluster:9100
          - vac016.beowulf.cluster:9100
          - vac017.beowulf.cluster:9100
          - vac018.beowulf.cluster:9100
          - vac019.beowulf.cluster:9100
          - vac020.beowulf.cluster:9100
          - vac021.beowulf.cluster:9100
          - vac022.beowulf.cluster:9100
          - vac023.beowulf.cluster:9100
          - vac024.beowulf.cluster:9100
          - vac025.beowulf.cluster:9100
          - vac026.beowulf.cluster:9100
          - vac027.beowulf.cluster:9100
          - vac028.beowulf.cluster:9100
          - vac029.beowulf.cluster:9100
          - vac030.beowulf.cluster:9100
prometheus.yml                                                1,1
"prometheus.yml" 57L, 1954C
```

VAC Cpu Load

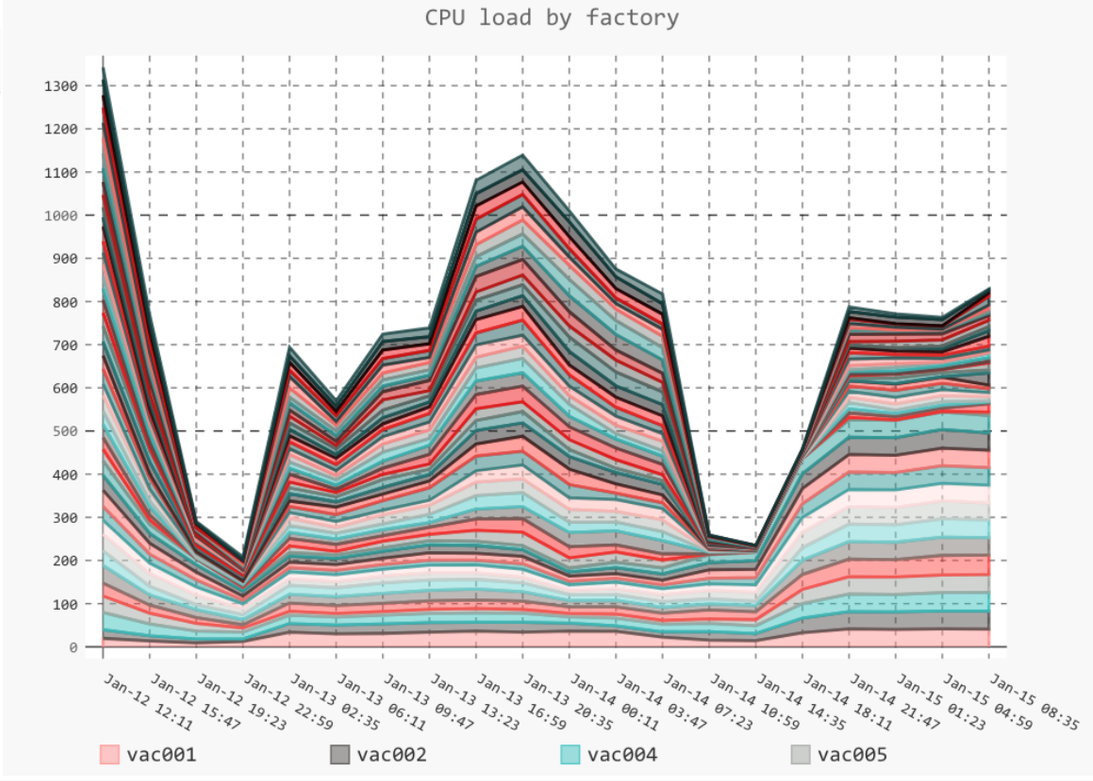
Zoom Out | Last 3 days

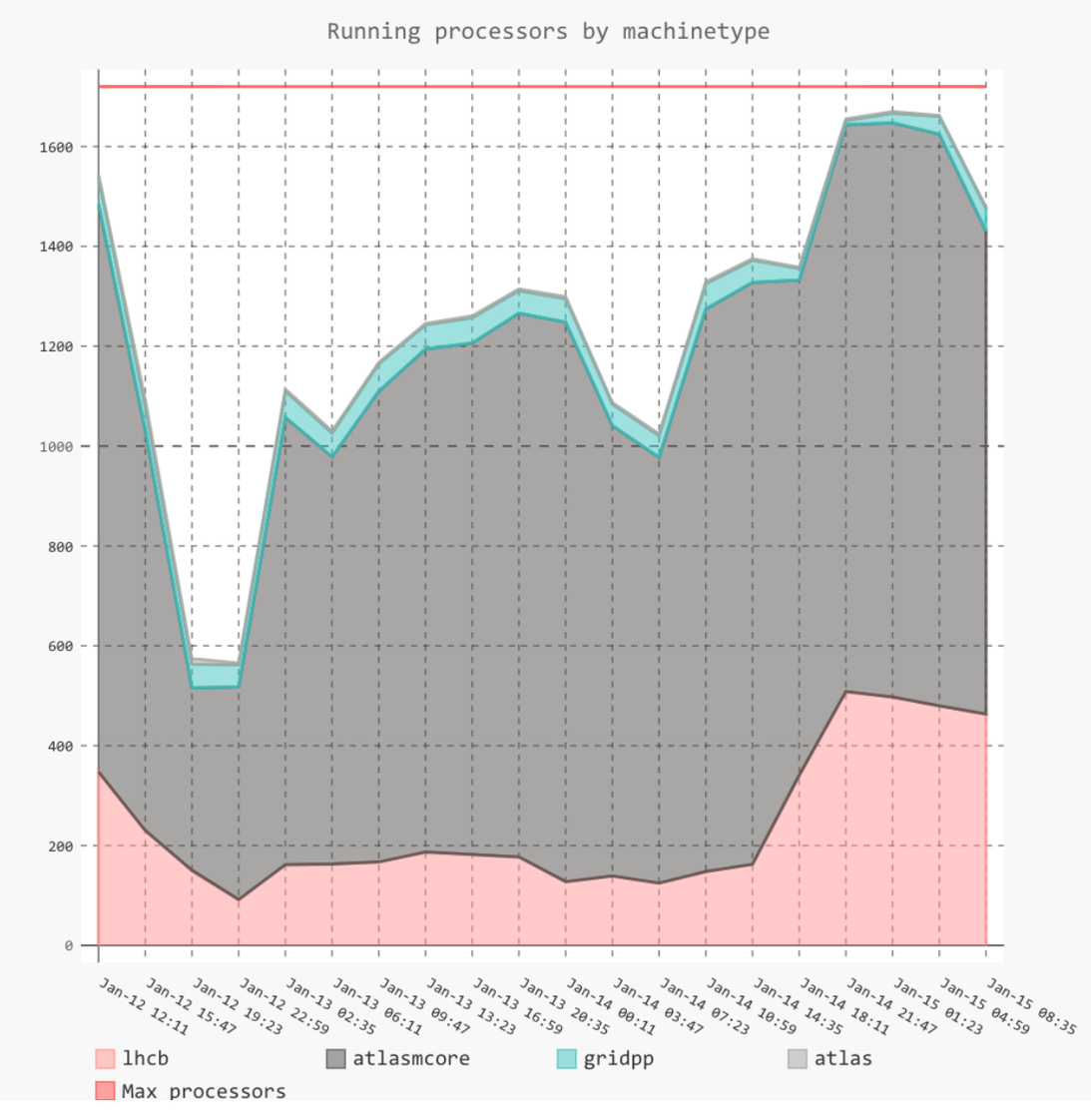
hostnames | All





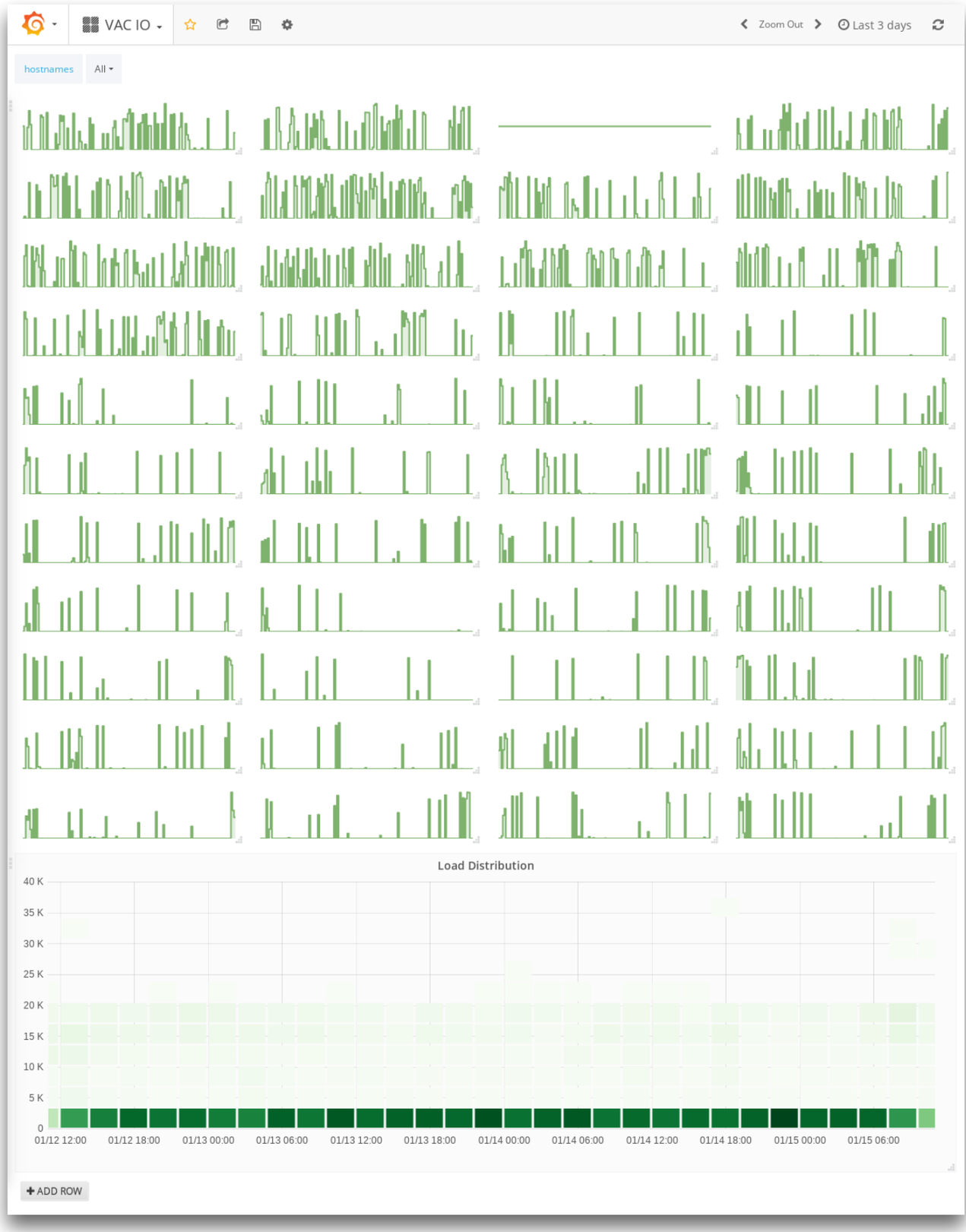
✚ ADD ROW





Network

Disk IO



Prometheus

- A good candidate for site monitoring.
- Things still to do:
  - CentOS6 init.d scripts
- Things I haven’t covered:
  - Recording
  - Alerting
  - Forwarding

mattdixon.co.uk