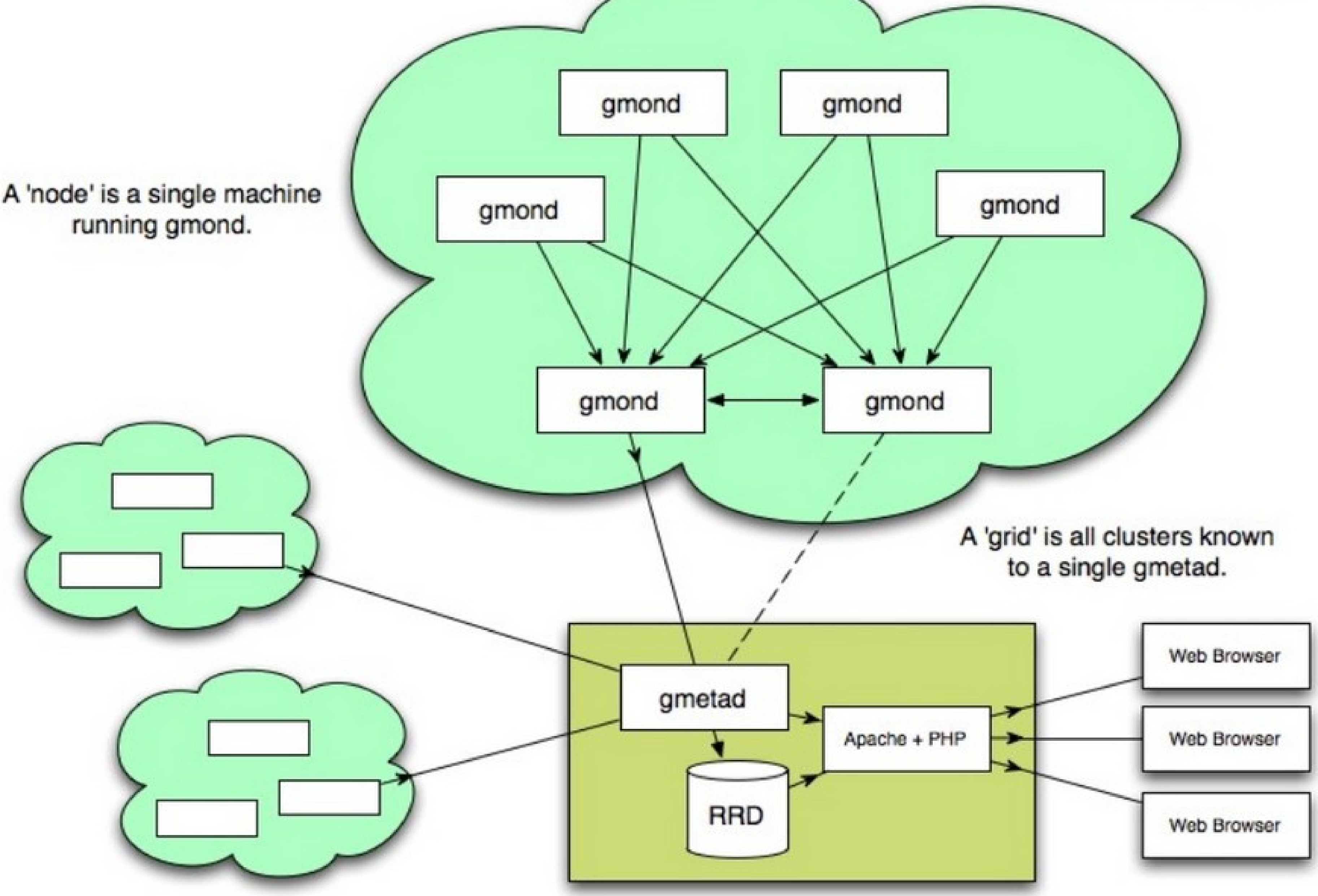
CENTER FOR
HIGH THROUGHPUT
COMPUTING

HTCondor

Monitoring HTCondor with Ganglia

Ganglia Overview

- Scalable distributed monitoring for HPC clusters
- Two daemons
  - gmond – every host; collects and send metrics
  - gmetad – single host; persists metrics from local gmond in RRD
- Web Frontend
  - Presents graphs from persistent data



A 'cluster' is a collection of gmond instances which report to a common set of head nodes.

A 'node' is a single machine running gmond.

A 'grid' is all clusters known to a single gmetad.

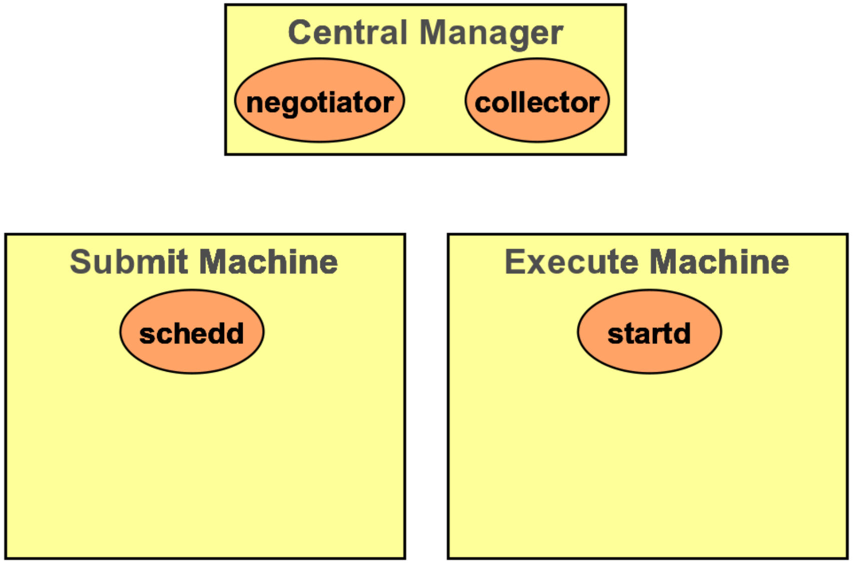
Why Ganglia?

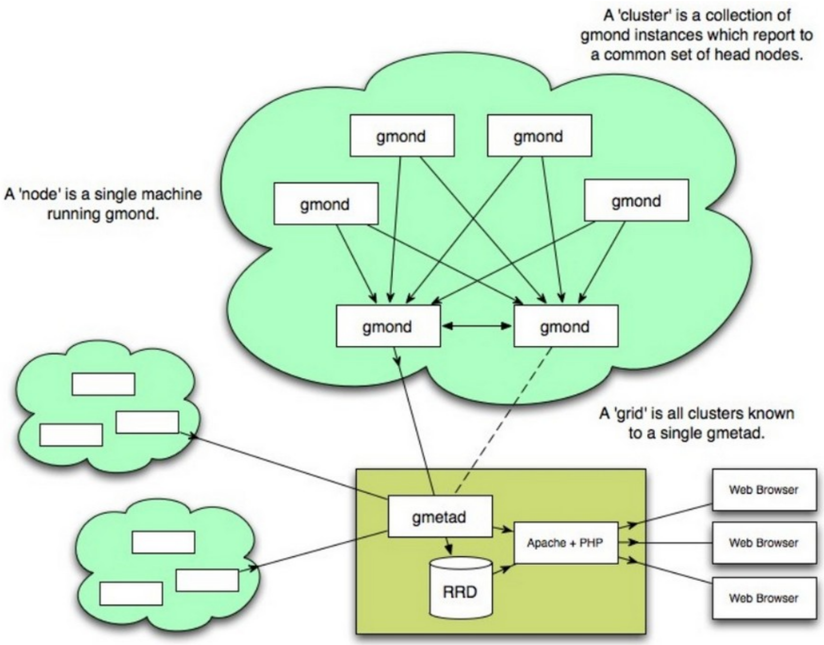
- Widely used monitoring system for cluster and grids
- Easy to add new metrics
- Can create custom graphs

Running condor_gangliad

- condor_gangliad runs on a single host
  - Gathers daemon ClassAds from the Collector
  - Publishes metrics to ganglia with host spoofing
- Can be on any host
- May be co-located with
  - condor_collector
  - gmetad
- Consider network traffic

Put Them Together




A 'cluster' is a collection of gmond instances which report to a common set of head nodes.
A 'node' is a single machine running gmond.
gmond
gmond
gmond
gmond
gmond
gmond
A 'grid' is all clusters known to a single gmetad.
gmetad
Apache + PHP
RRD
Web Browser
Web Browser
Web Browser

Possible Deployments

- Ganglia is already used for monitoring
  - Start condor_gangliad on gmetad host
    - Least configuration
  - Start condor_gangliad on Central Manager
    - Saves network traffic
- Ganglia is not in use for monitoring
  - Setup dedicated host to run ganglia and condor_gangliad
  - Generates graphs for web pages on demand

Ganglia Interface

- Uses gmetric method to add metrics to ganglia
  - Uses shared library on system to send updates
    - Fast and efficient
  - Falls back to using gmetric command
    - Much slower
- Uses gstat to determine which hosts are already monitored by ganglia

Configuration Macros

- GANGLIA_GSTAT_COMMAND
  - Defaults to localhost (change master gmond running elsewhere)
  - “gstat --all --mpifile --gmond_ip=localhost –gmond_port=8649”
- GANGLIA_SEND_DATA_FOR_ALL_HOSTS
  - Set to true if want hosts not currently in ganglia
- GANGLIAD_VERBOSITY
  - Defaults to 0, set higher for more monitoring

Running condor_gangliad

- Add to DAEMON_LIST
  - DAEMON_LIST = …, GANGLIAD
- Check GangliadLog for gmetric integration
  - Look for libganglia load message
    - Library has been stable over many releases
    - May have to specify path to library
  - If fall back to gmetric command look closely at timing

Log Snippet

```
04/24/14 08:05:43 Testing gmetric
04/24/14 08:05:43 Loading libganglia /usr/lib64/libganglia-3.1.7.so.0.0.0
04/24/14 08:05:43 Will use libganglia to interact with ganglia.
04/24/14 08:06:03 Starting update...
04/24/14 08:06:03 Ganglia is monitoring 1 hosts
04/24/14 08:06:10 Got 7687 daemon ads
04/24/14 08:06:14 Ganglia metrics sent: 1858
04/24/14 08:06:14 Heartbeats sent: 80
```

Limit Data

- GANGLIAD_PER_EXECUTE_NODE_METRICS
  - Set to false if large pool
- Use Requirement express to limit data fetched
  - GANGLIAD_REQUIREMENTS = Machine == "cm.chtc.wisc.edu" || Machine == "submit-1.chtc.wisc.edu" || Machine == "submit-2.chtc.wisc.edu" || Machine == "submit-3.chtc.wisc.edu"

Metrics to Track

- Described in /etc/condor/ganglia.d/
- Default set provided
- Expressed as ClassAds
  - Name: Unique metric name used by ganglia
  - Value: ClassAd expression, defaults to “Name”

Minimal Example

```
[
  Name   = "JobsSubmitted";
  Desc   = "Number of jobs submitted";
  Units  = "jobs";
  TargetType = "Scheduler";
]
```

Simple Example

```
[
  Name   = strcat(MyType,"DaemonCoreDutyCycle");
  Value  = RecentDaemonCoreDutyCycle;
  Desc   = "Recent fraction of busy time in the daemon event loop";
  Scale  = 100;
  Units  = "%";
  TargetType = "Scheduler,Negotiator,Machine_slot1";
]
```

Aggregate Metrics

- Can aggregate metrics over entire pool
  - Sums: running jobs over pool
  - Min and Max: Space Available
  - Average
- Aggregates appear in “HTCondor Pool” group on central manager

Aggregate Example

```
[
  Name    = "TotalJobAds";
  Desc    = "Number of jobs currently in this schedd's queue";
  Units   = "jobs";
  TargetType = "Scheduler";
]
[
  Aggregate = "SUM";
  Name    = "Jobs in Pool";
  Value   = TotalJobAds;
  Desc    = "Number of jobs currently in schedds reporting to this pool";
  Units   = "jobs";
  TargetType = "Scheduler";
]
```

Scaling Example

```
[
  Name    = strcat(MyType,"MonitorSelfResidentSetSize");
  Value   = MonitorSelfResidentSetSize;
  Verbosity = 1;
  Desc    = "RAM allocated to this daemon";
  Units   = "bytes";
  Scale   = 1024;
  Type    = "float";
  TargetType = "Scheduler,Negotiator,Machine_slot1";
]
```

Other Attributes

- Title = “Graph Title” (defaults to Name)
- Regex = for dynamic metric (users)
- Type = automatic based on type
  - Coerce integers to floats if scaling or large
- Group = “Group on Web Page”

Future Work

- Composite graphs
  - For example, I/O load and throughput
  - Better able to draw conclusions
- Graph slot states
- Determine which metrics are most useful

Live Demo

- http://timt.chtc.wisc.edu/ganglia
- http://cm.batlab.org/ganglia