Jobs Resource Utilization as a Metric for Clusters Comparison and Optimization

Joseph Emeras Cristian Ruiz Jean-Marc Vincent Olivier Richard

Slurm User Group Meeting
9 - 10 October, 2012

- Tracing of jobs consumptions for about a year
- Processor, disk, memory, network, but unfortunately not energy
- 2 production clusters
- Analyze the processor resource consumption vs. the resource allocation
- Results were not exactly what we expected...


Outline

Outline

Concepts

- Mesocenter, several universities and labs share computational resources.
- Lightweight grid software CiGri.
- Submission of multi-parametric jobs, bag of task, big amount of jobs.
- Based on the submission of **Low priority jobs**, preemptible called **Best-effort** jobs.
  - Improve the cluster utilization, by using unused space (free resources).
- **Production clusters.**
- Various domains : Chemistry, Astronomy, Physics, etc.
- Underlying Batch Scheduler is OAR (it could have been Slurm...)

CIGRI problematics

Questions

- Best-Effort jobs seems a good idea, platform resources are highly allocated.
- But is it really a good thing to over-allocate ? (bottlenecks ?)
- Other optimizations possible ?
- We need data from jobs consumptions.
- OAR, as SLURM provides an accounting plugin, gives **means**.
- We suspected that consumption means are not enough, we have very long jobs (days).
- Indeed, 30% of the jobs had a std dev over 20%, for CPU consumption.
- We need finer granularity.

System Utilization as a metric

- Metric commonly used to evaluate computational infrastructures utilization.
- Definition : ratio of the computing resources allocated to the jobs over the available resources in the system.
- Downsides : **misses information** about the computer sub-systems (disk , memory, processor).
- **Jobs resource utilization**.

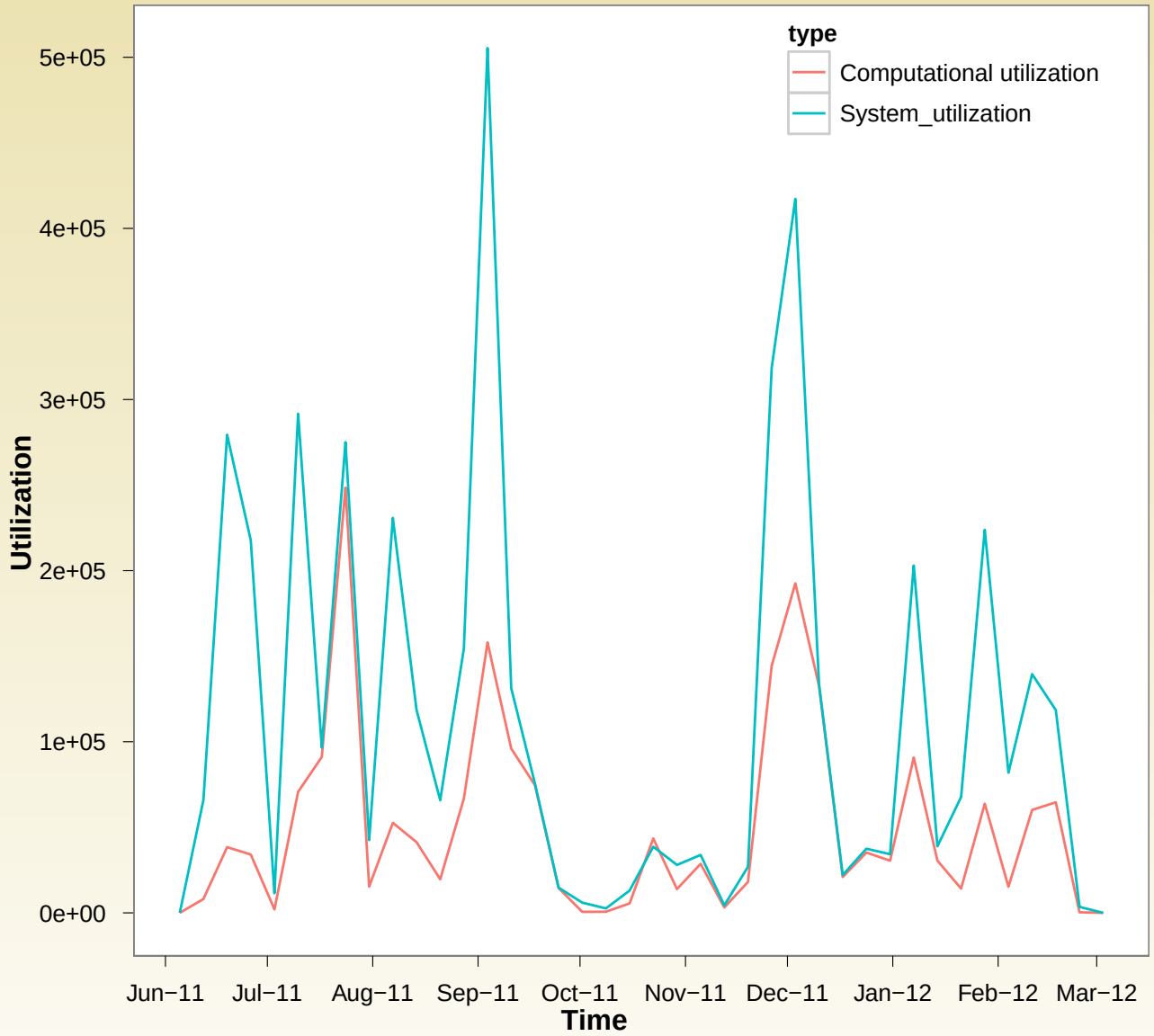



Figure: System Utilization vs Computational Resource Utilization (for normal jobs from one of the CIMENT Clusters)

Outline

Jobs traces

- The idea is to have a trace of resources consumption per job.
- Different from other monitoring approaches such as Ganglia[2], Nagios[1], etc.
- Job centric monitor.
- Information about (CPU, memory, IO, Network) consumption.

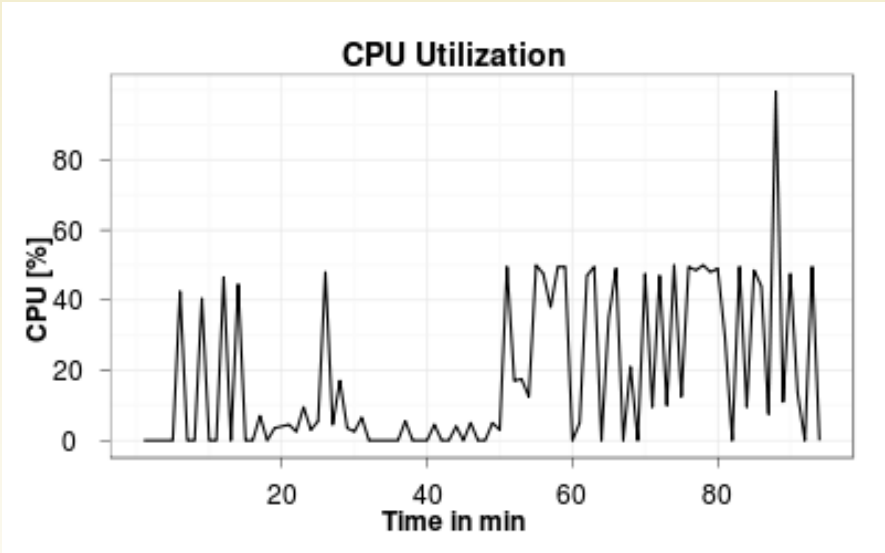


Figure: Job CPU consumption over time

Choices for Tracing Jobs Consumptions

Characteristics

- Independent : No synchronization among the nodes.
- Use of mechanisms supported by most of the batch schedulers.
- Lightweight (Sampling approach) 0.35% speed down.

Choices for Tracing Jobs Consumptions

Characteristics

- Independent : No synchronization among the nodes.
- Use of mechanisms supported by most of the batch schedulers.
- Lightweight (Sampling approach) 0.35% speed down.

Lightweight approach

- 1 min frequency sampling, to avoid storage overhead and to not interfere with jobs execution.
- Capture not expensive using */proc* directory.
- Data processing done off-line.

Clusters characteristics

	Foehn	**Gofree**
CPU Model	Xeon X5550	Xeon L5640
Nodes	16	28
CPU cores	128	336
Total memory	480 GB	2016 GB
Memory/node	24/48 GB	72 GB
Total storage	7 TB	30 TB
Network	IB DDR	IB QDR
Total Gflop/s	1367.04	3177.6
Buy date	2010-03-01	2011-01-01

Trace summary

	Foehn	Gofree
Capture start	2011-06-01	2011-05-24
Capture months	9	9
Number of jobs	41230	9052
Log Size	2.5 Gb	3.0 Gb
Besteffort jobs	38558	5451
Normal jobs	2672	3601

Outline


Data Processing

Pre-analysis

- Off-line data processing
- Traces Correlation
  - Batch Scheduler and Resource Manager logs (SWF Format)

Standard Workload Format

SWF Format	
Headers comments	Fields
Version	Job Number
Computer	Submit Time
Installation	Wait Time
Acknowledge	Run Time
Information	Number of Allocated Processors
Conversion	Average CPU Time Used
MaxRecords	Used Memory
Preemption	Requested Number of Processors
UnixStartTime	Requested Time
TimeZone	Requested Memory
TimeZoneString	Status
StartTime	User ID
Endtime	Group ID
MaxNodes	Executable Number
MaxProcs	Queue Number
MaxRuntime	Partition Number
MaxMemory	Preceding Job Number
AllowOveruse	Think Time form Proceeding Job
MaxQueues	
Queues	
Queue	
MaxPartitions	
Partitions	
Partition	
Note	


Data Processing

Pre-analysis

- Off-line data processing
- Traces Correlation
  - Batch Scheduler and Resource Manager logs (SWF Format)
  - Jobs resource consumption logs

Jobs Resource Consumption Logs

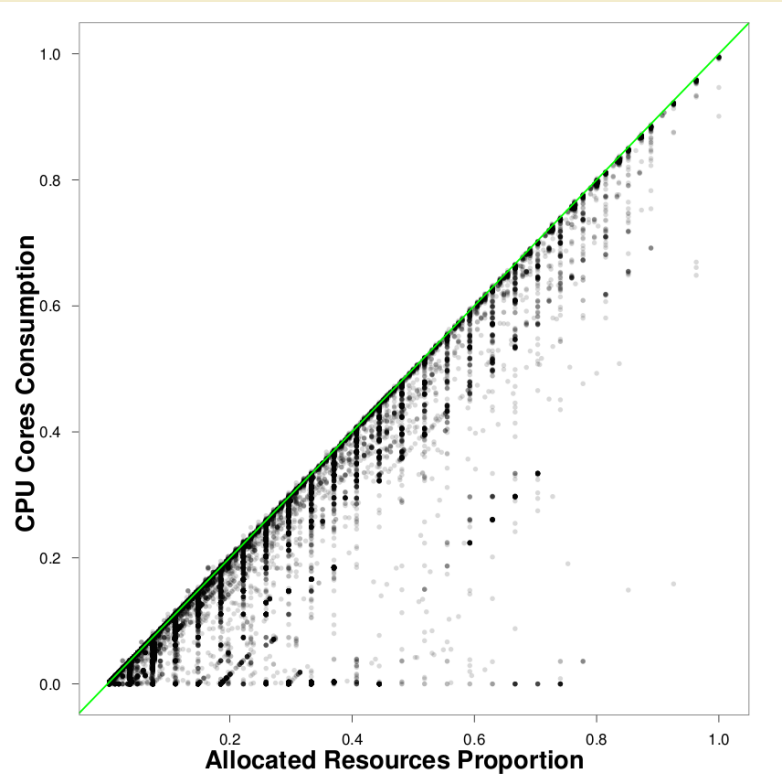
Trace Fields	
Name	Description
Time	Unix Time Stamp in seconds
JOB ID	Job id assigned by the batch scheduler
PID	PID of process that belongs to the job
Node ID	Provenance of the capture
Measure	List of measures of the resource consumptions

Measure (simplified version)	
Name	Description
command	Name of the binary executed
memory_faults	Number of memory faults and their type
virtual_memory	Virtual memory size
pages	Pages used by process and their type
IO_r_w	Num of bytes read/written from/to the storage layer
core_usage	Core utilization percentage
net_r_w	Network Read and Written Bytes
...	...

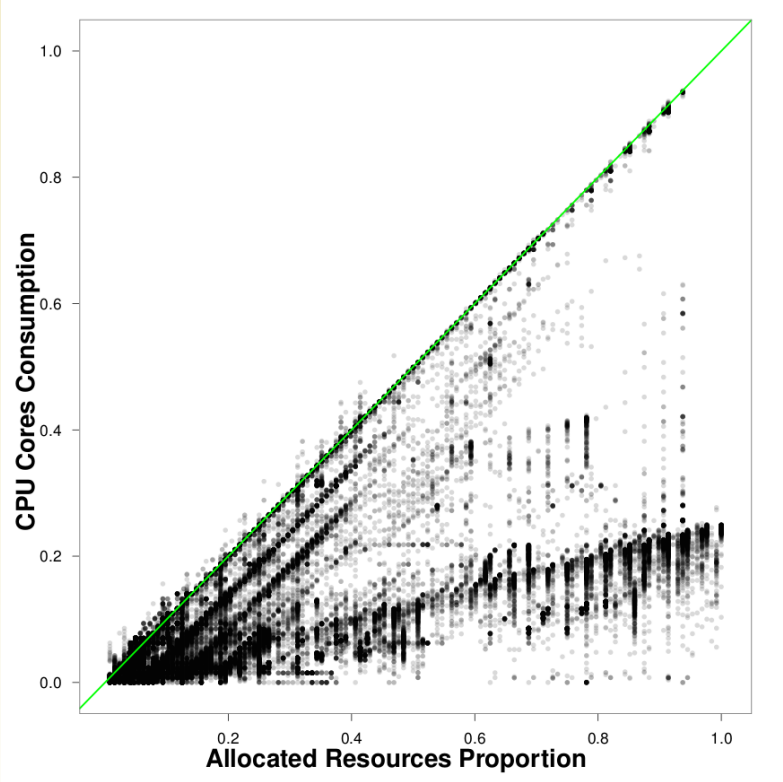
Resource Utilization Criterion

Criterion presentation

- Match Resource Consumption with Resource Allocation
- X axis : allocated resources over available
- Y axis : core consumption in function of X
- Green diagonal : Theoretical optimum, **distances from the line = computing power loss**
- Density graphs : darker point means denser distribution **(alpha=0.1)**

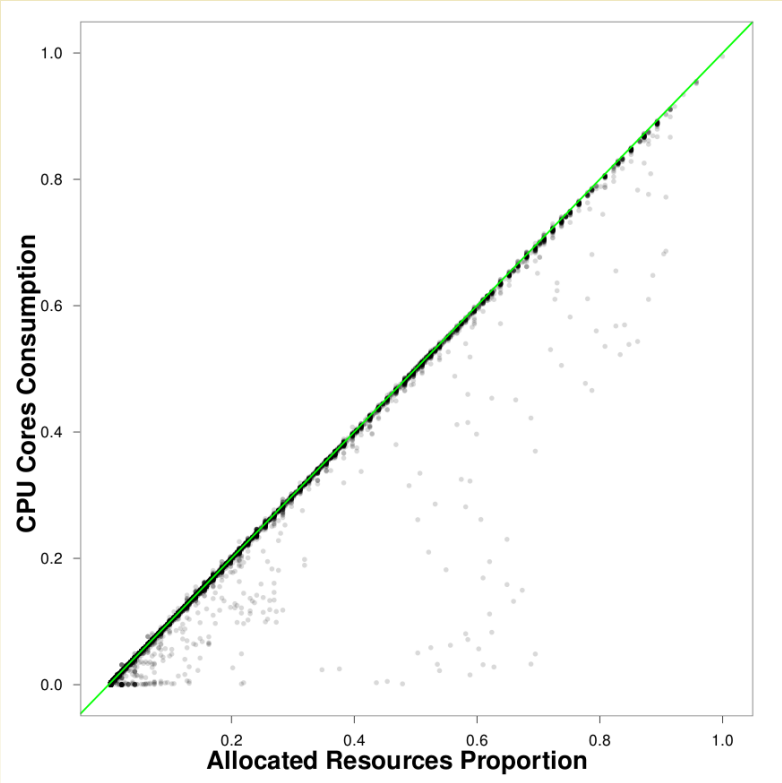


Gofree Cluster



Foehn Cluster

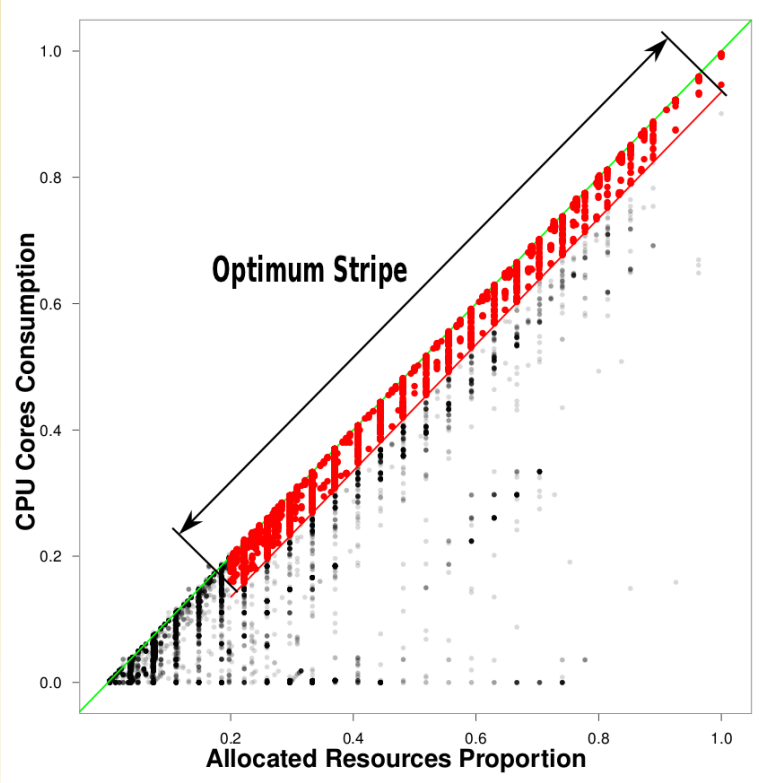
Gofree Cluster



Gofree BestEffort Jobs

Full consumption of reserved cores

BestEffort jobs are core efficient.

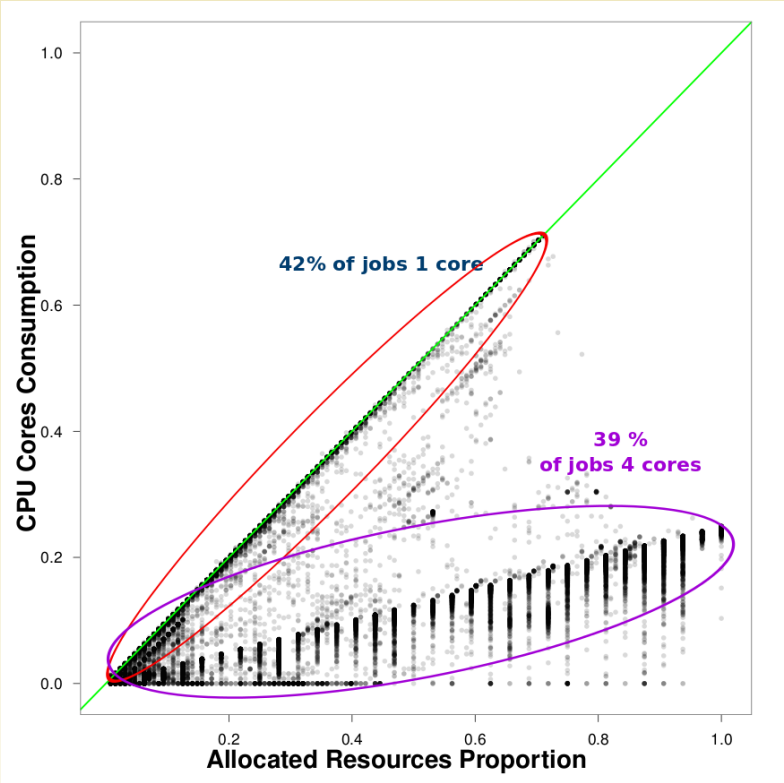


Gofree Normal Jobs

Red stripe : 80% of the values.
Linear regression gave a 0.998 slope.

(red stripe : stripe between the optimum and the mean of the distances from the optimum)

Foehn Cluster

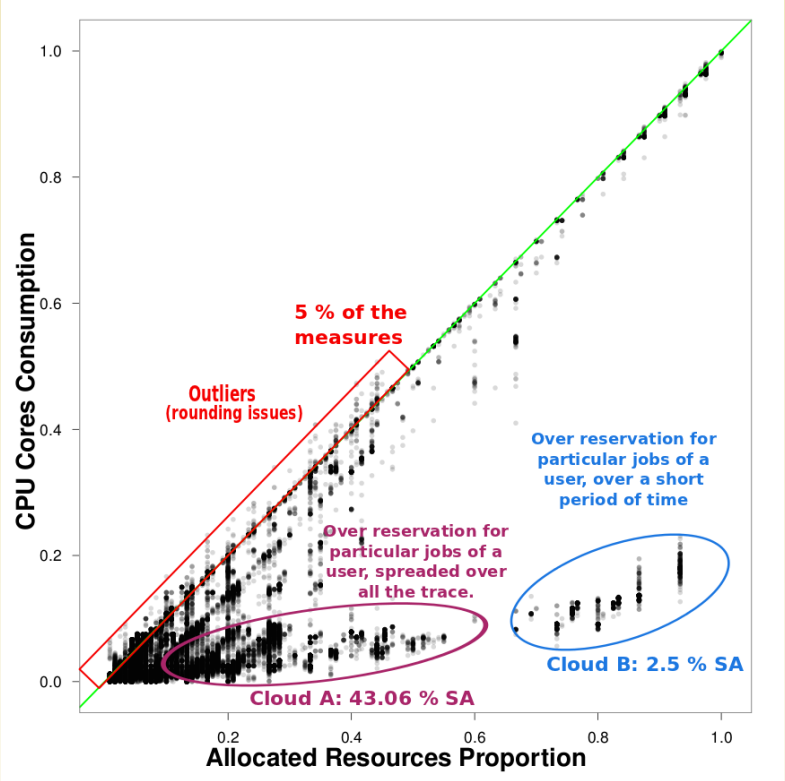


Foehn BestEffort Jobs

2 patterns : one optimal, one 1/4.

One user with particular needs for memory bandwidth.

$$SA = \sum_{j \in jobs} alloc_j \times run\_time_j.$$



Foehn Normal Jobs

2 “low-utilization” clouds.

Outliers (up to 110%) due to */proc* roundings when consumption jitter.

Users Special Needs

Needs not expressible by the user to the batch scheduler ?

IO/Memory Bandwidth Constraint

- accept the loss ?
- modify batch scheduler ?
- bandwidth as a resource ?

Outline


Discussion

System Instrumentation

- Interesting and at little cost **(our implementation : 0.35% speed-down)**
- Need to correlate with other metrics **(memory size and bandwidth usage, IO)**

Results

- Processor consumption on 2 clusters of same grid can be **very different**
- Users behaviors impact
- Shows Batch Scheduler request constraint lacks

Future Works

- Characterize jobs consumption patterns
- Learn from past, classify couple <User,Code>
- On-Line tagging of jobs at submission, useful to anticipate IO intensive jobs

Ideas

- SWF : fields for Max Memory, particular user constraint (license, hardware, locality)
- Slurm : plugin for tracing the jobs by sampling

Data available in a git repository


Thank you for your attention

joseph.emeras@imag.fr


Emir Imamagic and Dobrisa Dobrenic.
Grid infrastructure monitoring system based on nagios.
In *Proceedings of the 2007 workshop on Grid monitoring*, GMW '07, pages 23–28, New York, NY, USA, 2007. ACM.


Matthew L. Massie, Brent N. Chun, and David E. Culler.
The ganglia distributed monitoring system : Design, implementation and experience, 2004.