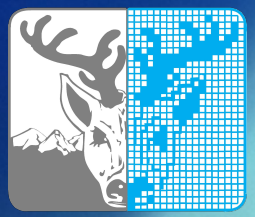

UMEÅ UNIVERSITET

Software accounting using SGAS, Slurm & CGroups

Magnus Jonsson
<magnus@hpc2n.umu.se>
http://www.hpc2n.umu.se/



Background

- What software are our users running?
- What version of the software are they using?
- How much of the software are they using?
- Are the users running the software they say they are using?
- Are we putting optimization effort on the right software?

Background

- This must be doable?

Background

- This must be doable?
- Or?

Background

- This must be doable?
- Or?
- Proof of concept

Techniques used/prerequisite

- /proc/
- Slurm
- CGroups
- perl
- json
- SGAS

CGroups

- Linux
- Contain
  - CPU
  - Memory

Slurm

- Uses CGroups to contain jobs
  - Memory
  - CPU
  - ...
- One CGroup for each job per node

/proc/

- CGroups leaves a trace in /proc
- /proc/<pid>/cpuset
  - /slurm/uid_1234/job_1234567/step_1
- /proc/<pid>/exe
  - Symbolic link to the binary running
- /proc/<pid>/task/stat
  - Same as /proc/<pid>/stat but for every thread.
  - User/System/”walltime”/vsize/RSS

Collector

- Perl script that runs on every node
  - Daemon
  - Cron
- Takes < 0.1s to run
  - Including loading perl and modules

Problems?

- User compiled software
  - Do we care?
- When is a job completed?
  - Assume
  - Ask Slurm
- What to save?
- Sample
  - What are we missing?
- Translate running exec to software

When is a job completed?

- Assume?
- Ask slurm?

What to save?

- To much to save it all... or?
- Aggregated user/system time for each executable
- Some other metadata
- Save the largest 90% by cputime

Translate executable to program

- Orders list
- Regular expression to map path to “software” + “version”
  - Software/Version can be fixed or collected as part of path
  - User provided

How to save?

- Currently saves files into our // file system
  - Lots of files...
- Plugin for SGAS to save info in the same database.
  - Information about jobs already available in the same database
  - Easy to do with the new plugin infrastructure

Sample problem

- Today
  - Cron every 5min
- Next generation
  - Cron every 5min
  - Daemon every minute
- Short running jobs/processes
- Kernel support?
- LD_PRELOAD?

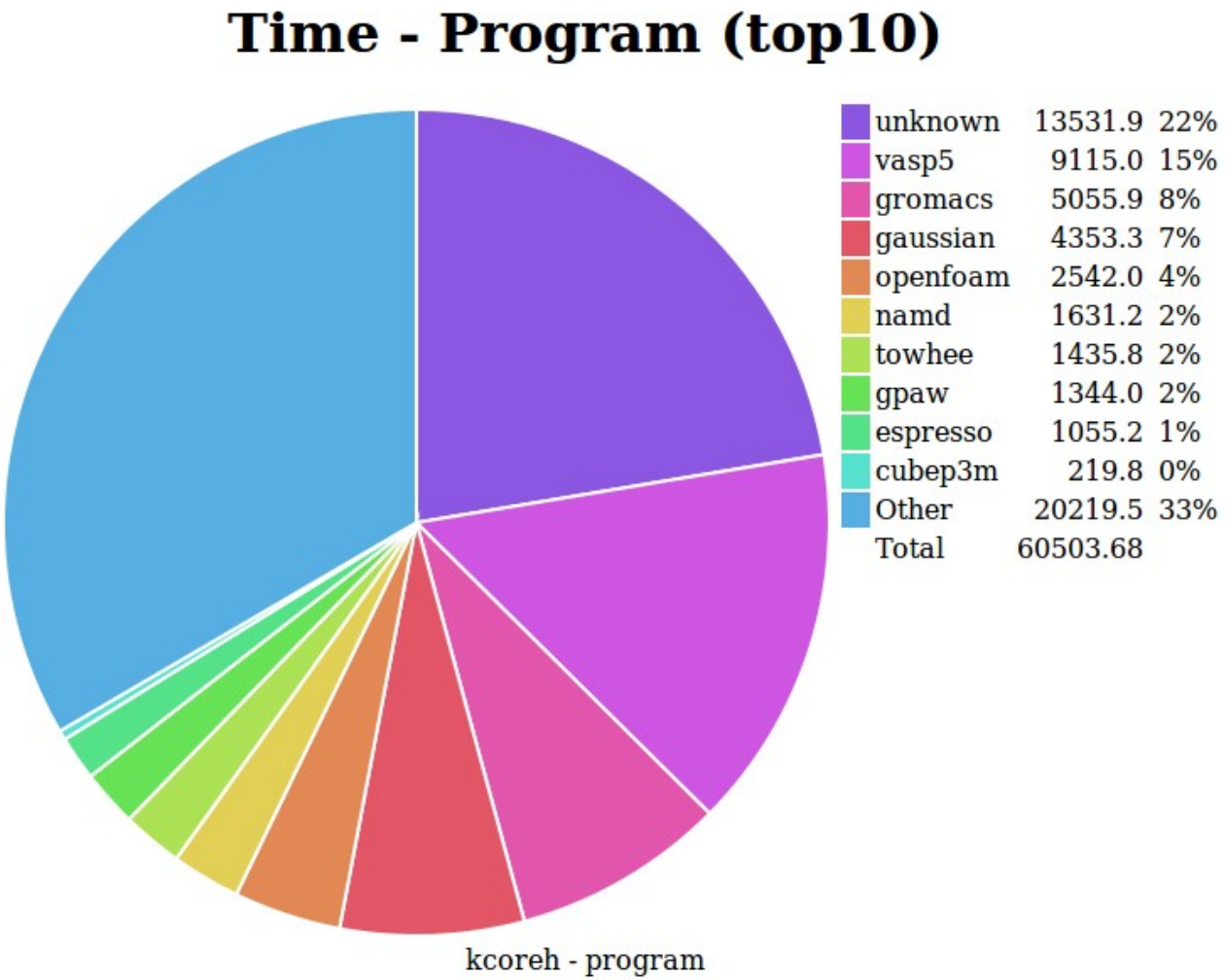
Post Processing

- Aggregate software for each job
  - Read Slurm accounting database
    - Project
    - Number of nodes
    - Job Walltime
    - ...?
  - Read from SGAS

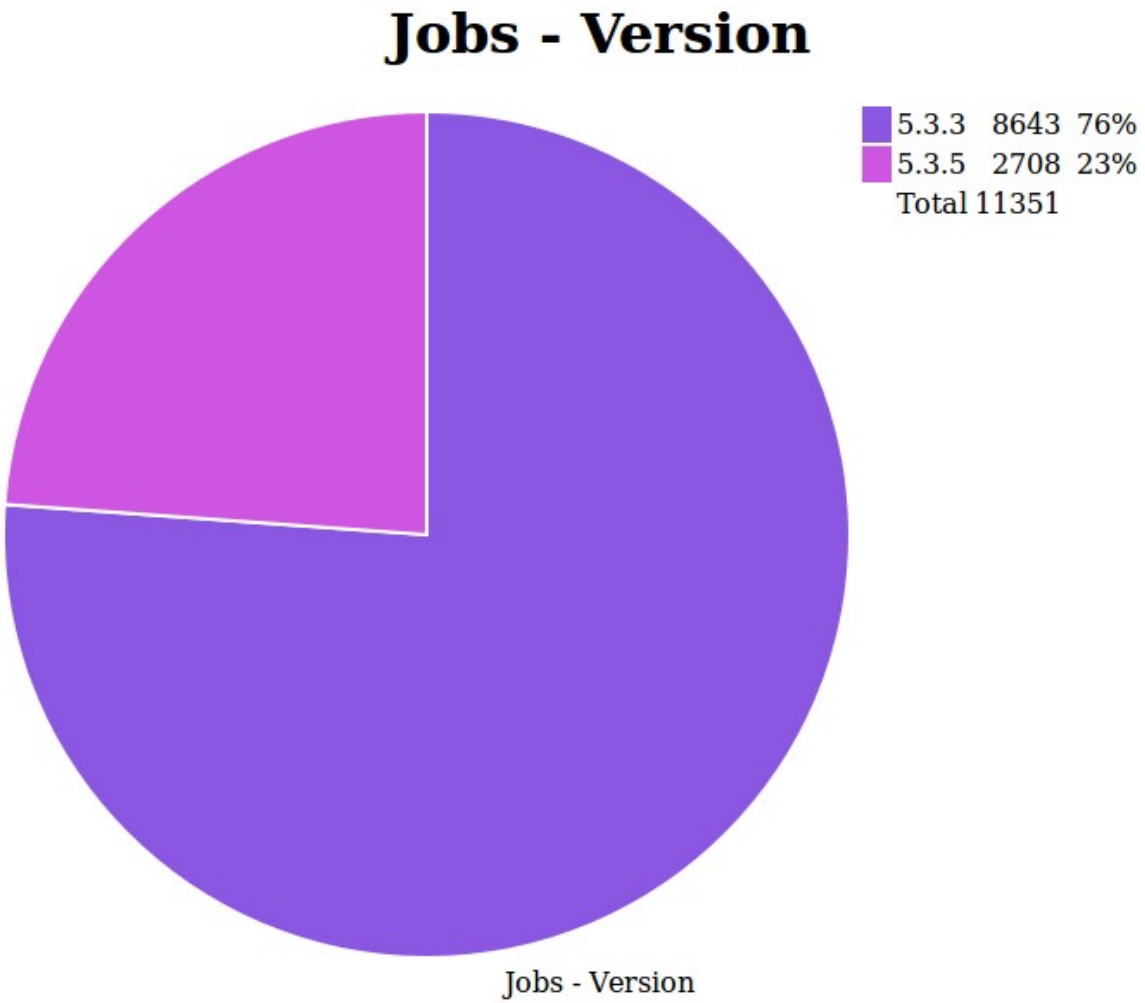
Make pretty graphs

- We are using a JavaScript tool.
  - Developed by me for an private project
  - All calculations on client side
  - Can be made into a single html file
  - Usable limit around 200-300k jobs

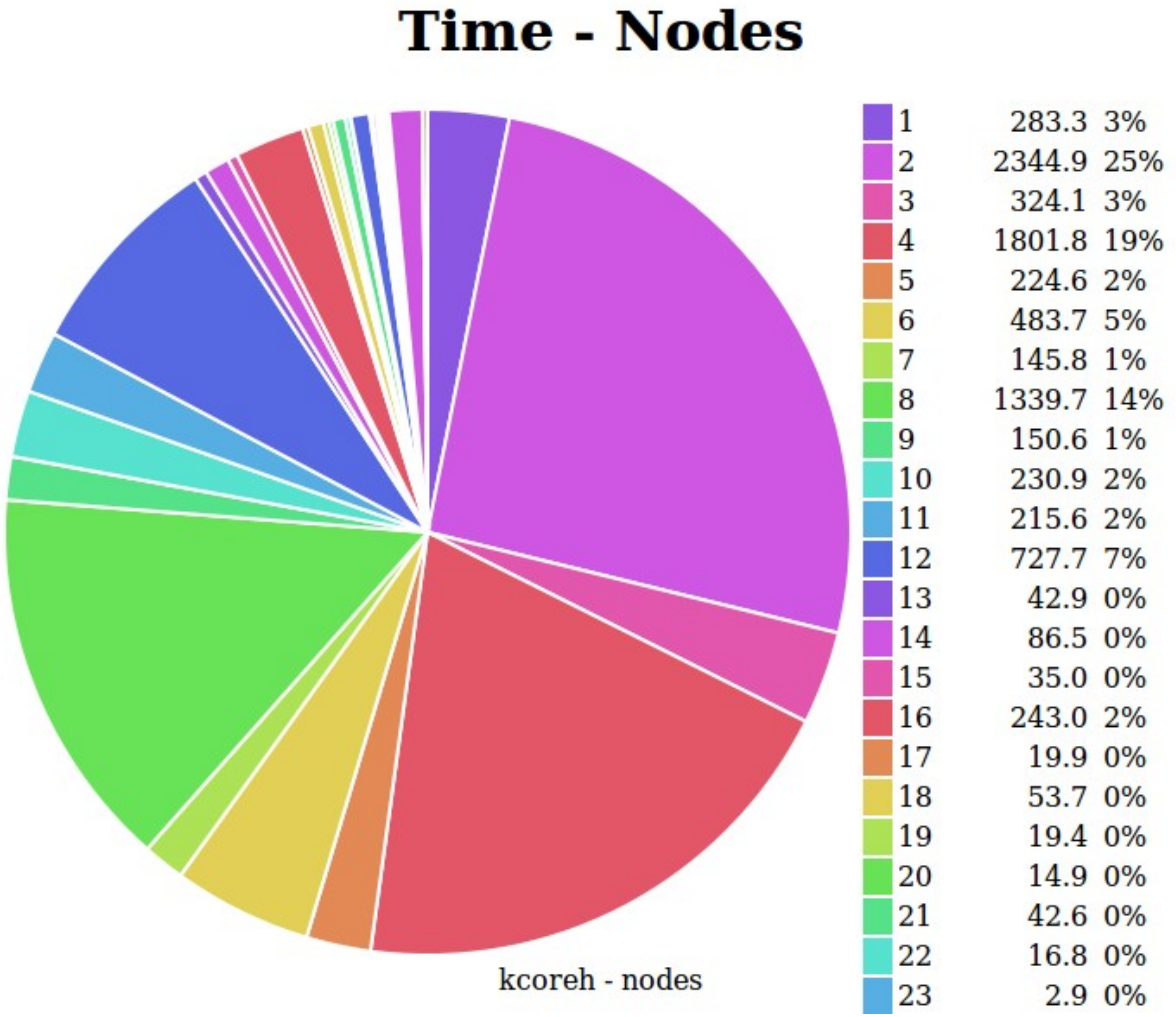
Show pretty graphs...



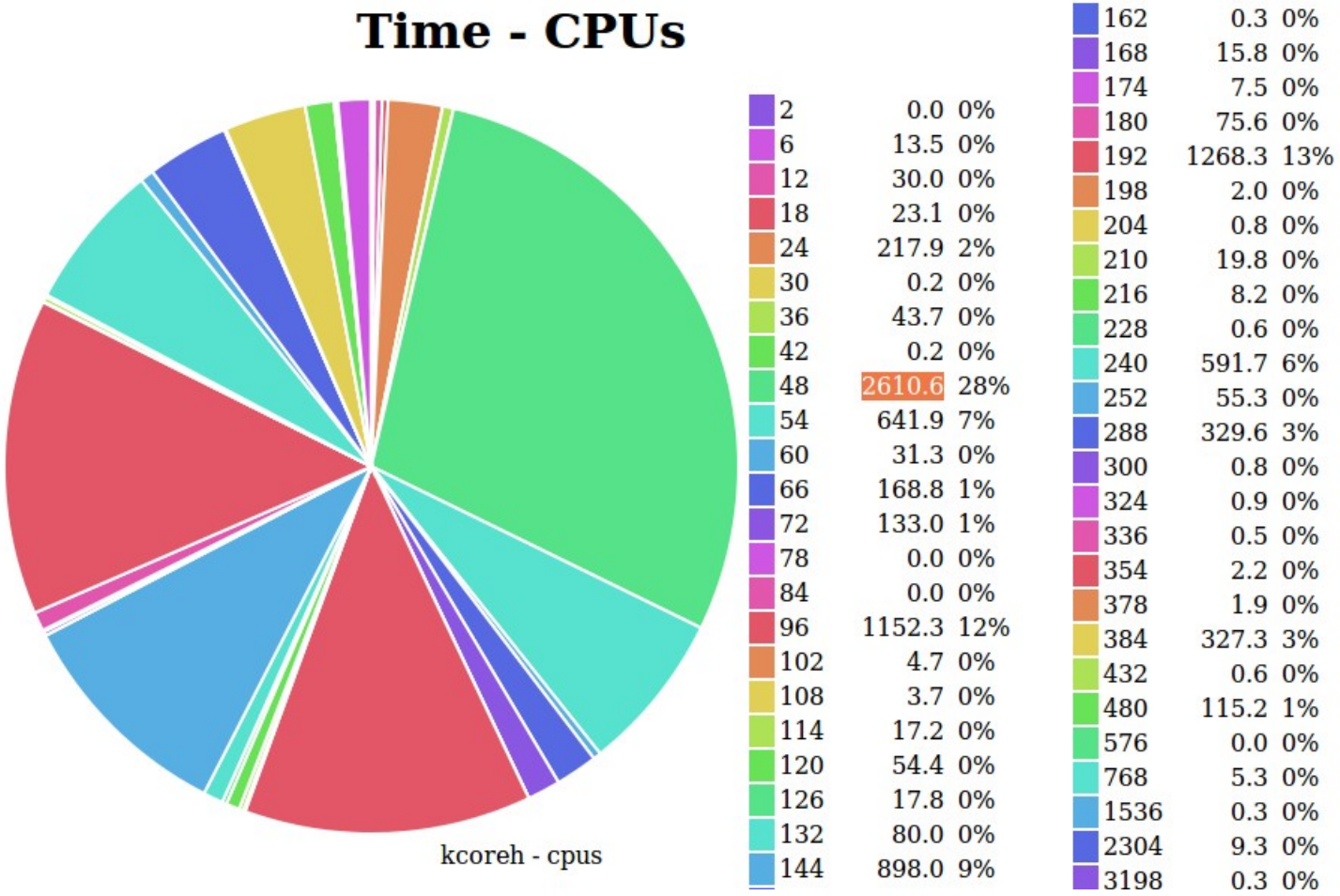
VASP


Jobs - Version
5.3.3 8643 76%
5.3.5 2708 23%
Total 11351
Jobs - Version

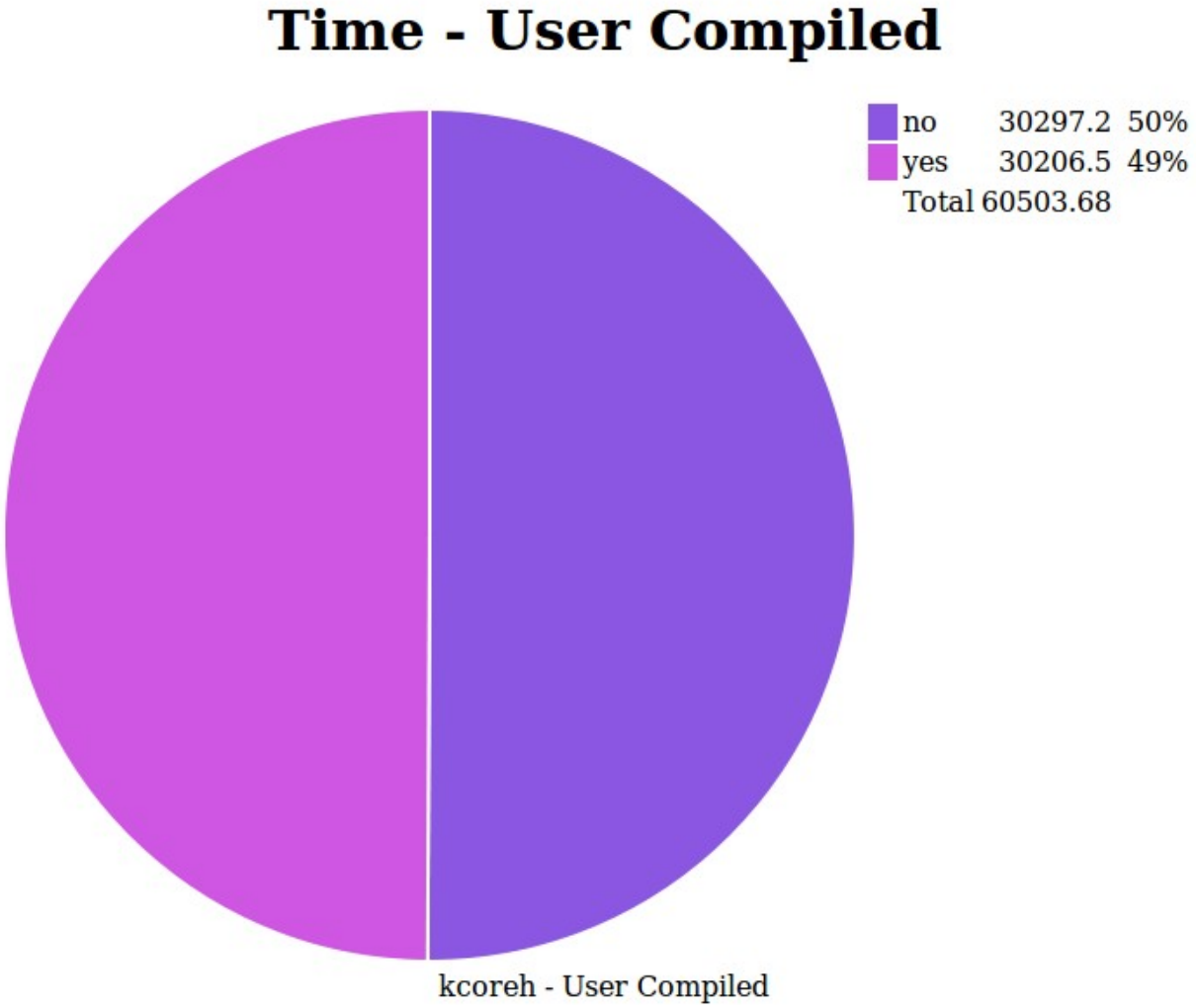
VASP


Time - Nodes
kcoreh - nodes
1 283.3 3%
2 2344.9 25%
3 324.1 3%
4 1801.8 19%
5 224.6 2%
6 483.7 5%
7 145.8 1%
8 1339.7 14%
9 150.6 1%
10 230.9 2%
11 215.6 2%
12 727.7 7%
13 42.9 0%
14 86.5 0%
15 35.0 0%
16 243.0 2%
17 19.9 0%
18 53.7 0%
19 19.4 0%
20 14.9 0%
21 42.6 0%
22 16.8 0%
23 2.9 0%

VASP


Time - CPUs
kcoreh - cpus
2 0.0 0%
6 13.5 0%
12 30.0 0%
18 23.1 0%
24 217.9 2%
30 0.2 0%
36 43.7 0%
42 0.2 0%
48 2610.6 28%
54 641.9 7%
60 31.3 0%
66 168.8 1%
72 133.0 1%
78 0.0 0%
84 0.0 0%
96 1152.3 12%
102 4.7 0%
108 3.7 0%
114 17.2 0%
120 54.4 0%
126 17.8 0%
132 80.0 0%
144 898.0 9%
162 0.3 0%
168 15.8 0%
174 7.5 0%
180 75.6 0%
192 1268.3 13%
198 2.0 0%
204 0.8 0%
210 19.8 0%
216 8.2 0%
228 0.6 0%
240 591.7 6%
252 55.3 0%
288 329.6 3%
300 0.8 0%
324 0.9 0%
336 0.5 0%
354 2.2 0%
378 1.9 0%
384 327.3 3%
432 0.6 0%
480 115.2 1%
576 0.0 0%
768 5.3 0%
1536 0.3 0%
2304 9.3 0%
3198 0.3 0%

User Compiled Software



Problems?

- User compiled software
  - Do we care?
- When is a job completed?
  - Assume
  - Ask Slurm
- What to save?
- Sample
  - What are we missing?

Live Demo :-)

Thanks for me

- Questions?