High Performance Roundtable

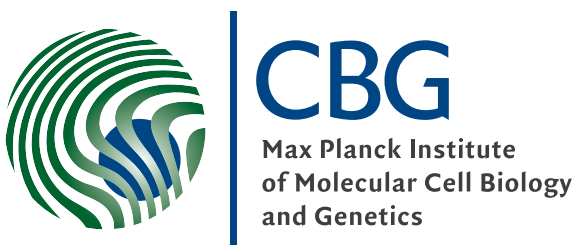
- Introducing Furiosa -

Òscar González[1] Peter Steinbach[2]

[1]Computer Department

[2]Scientific Computing Facility

March 17th, 2016



Outline

Outline

Benchmarks : Status

- required to double check what was promised by vendor
- expected to be concluded until easter
- benchmark performance parts of it or entire cluster

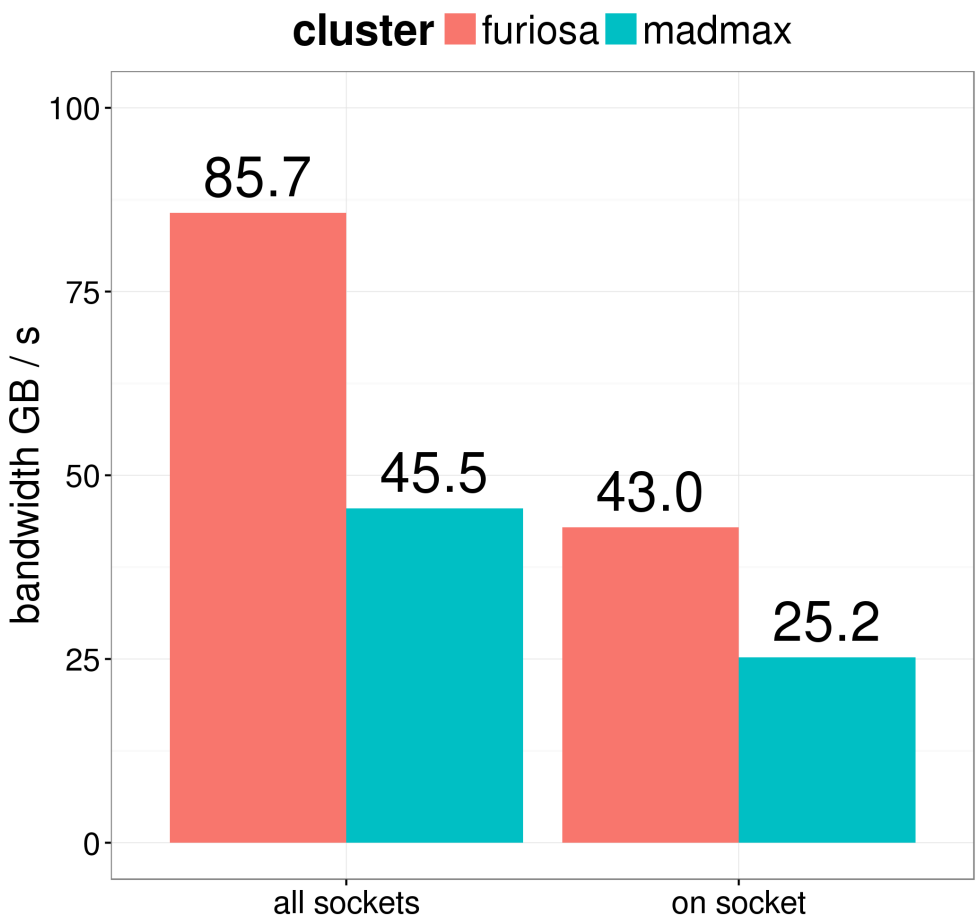
FINISHED	memory
PRELIMINARY	CPU+network
PRELIMINARY	file system

Benchmarks : Memory

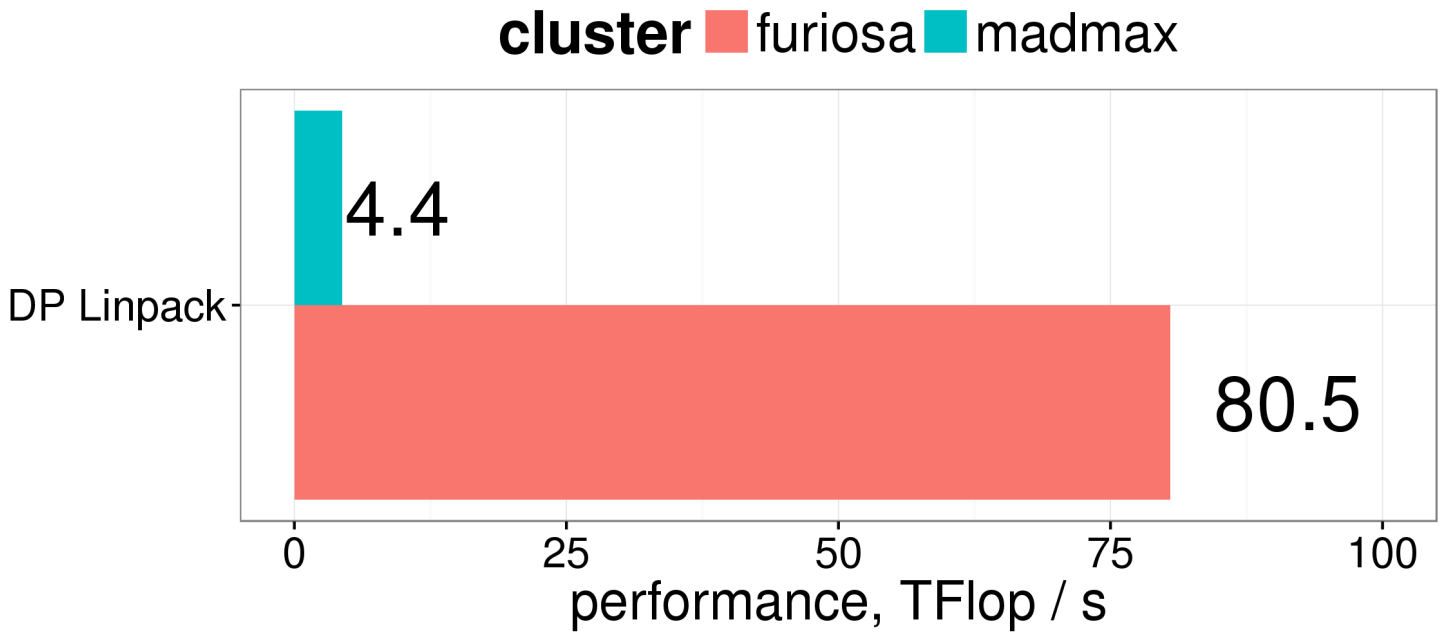
- stream benchmark 5.10 (simple operations on large arrays in memory)
- `Triad`

```
for(size_t i = 0;i<len;++i)
  a[i] = b[i] + c*d[i];
```

- double precision items
- array size $>$ L3 cache size



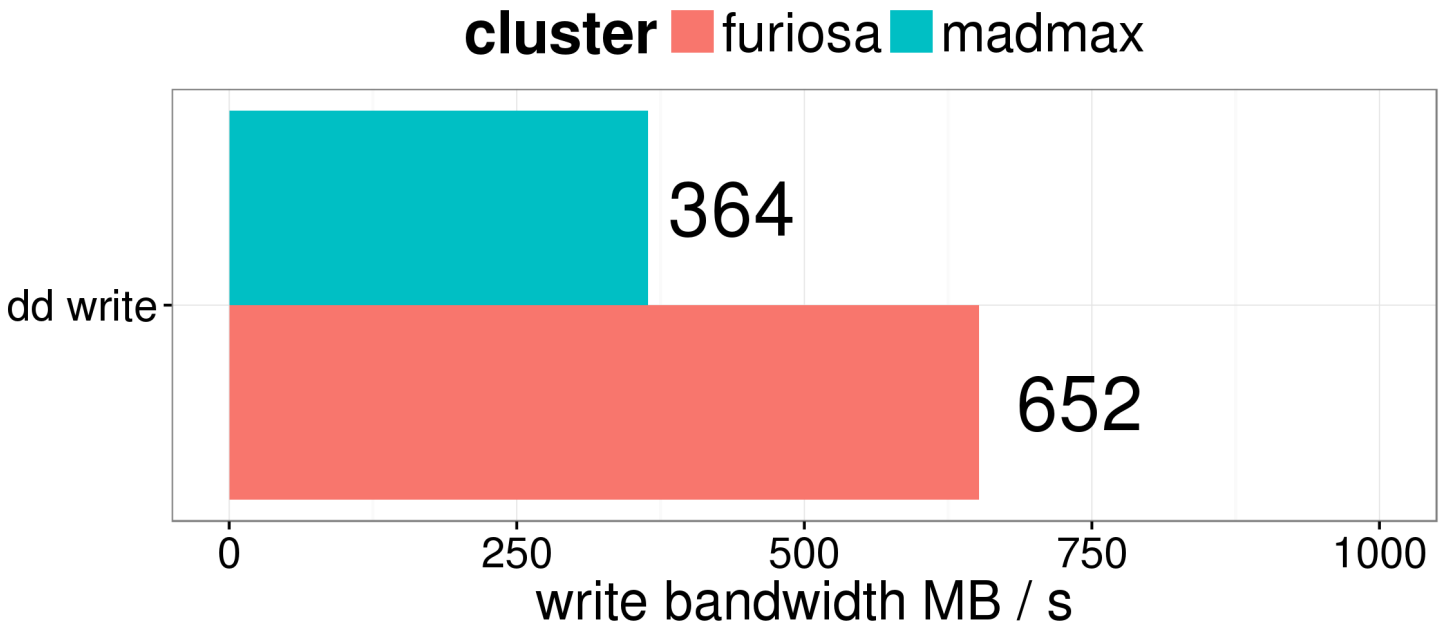
Benchmarks : CPU+network



- High Performance Linpack 2.1
- huge matrix inversion through LU factorization stretching all nodes

- double precision workload
- furiosa: **No. 1 in Top500 in 2004**
- enclosure (16 nodes): 11.5 $\mathrm{TFlop/s}$

Benchmarks : File System



- simple `dd` call
  (sequential write of 48 GB file)
- single node i/o

- 2× bandwidth than madmax
- 10× storage volume than madmax
- expect $> 20\,\mathrm{GB/s}$ cluster-wide

Outline

SLURM : Batch System

- **Free and Open Source Job Scheduler** (GPL)
- 6 out of 10 Top500 clusters use it (June 2015)
- TUD uses it on taurus
- energy aware (built in profiling)
- scalable (islands of nodes)
- flexible (plugin system)
- fault tolerant
- ...

SLURM : Batch System

- **Free and Open Source Job Scheduler** (GPL)
- **6 out of 10 Top500 clusters** use it (June 2015)
- TUD uses it on taurus
- energy aware (built in profiling)
- scalable (islands of nodes)
- flexible (plugin system)
- fault tolerant
- ...

SLURM : Submission 1

LSF on madmax

```
$ bsub -W 00:20 -o my.log -n 12 -R "span[hosts=1]" ./a.out
```

SLURM : Submission 1

LSF on madmax

```
$ bsub -W 00:20 -o my.log -n 12 -R "span[hosts=1]" ./a.out
```

SLURM on furiosa

```
$ srun --time=00:20:00 -o my.log -N 1 -n 1 -c 24 ./a.out &
```

- srun does wait for job to complete
- how many nodes `-N 1`
- how many tasks per node `-n 1`
- how many cores per task `-c 24`
- provide timing information to get better placement

SLURM : Submission 2

lsf.script

```
#!/bin/bash
#BSUB -W 00:20
#BSUB -o my.log
#BSUB -n 12
#BSUB -R "span[hosts=1]"

./a.out
```

LSF on madmax

```
$ bsub < lsf.script
```

SLURM : Submission 2

lsf.script

```
#!/bin/bash
#BSUB -W 00:20
#BSUB -o my.log
#BSUB -n 12
#BSUB -R "span[hosts=1]"

./a.out
```

slurm.script

```
#!/bin/bash
#SBATCH --time=00:20:00
#SBATCH -N 1
#SBATCH -n 1
#SBATCH -c 24
#SBATCH -o my.log
./a.out
```

LSF on madmax

```
$ bsub < lsf.script
```

SLURM on furiosa

```
$ sbatch slurm.script
```

SLURM : Monitoring

from the command-line (very customizable)

```
$ squeue -u ${USER}
JOBID PARTITION NAME USER ST TIME NODES NODELIST(REASON)
180 batch slurm.sc steinbac R 0:11 1 c07n04
181 batch slurm.sc steinbac R 0:11 1 c07n07
182 batch slurm.sc steinbac R 0:11 1 c07n08
...
$ squeue -t RUNNING -o "%.6i %p"
JOBID PRIORITY
180 0.00000081583858
181 0.00000081583858
182 0.00000081583858
...
```

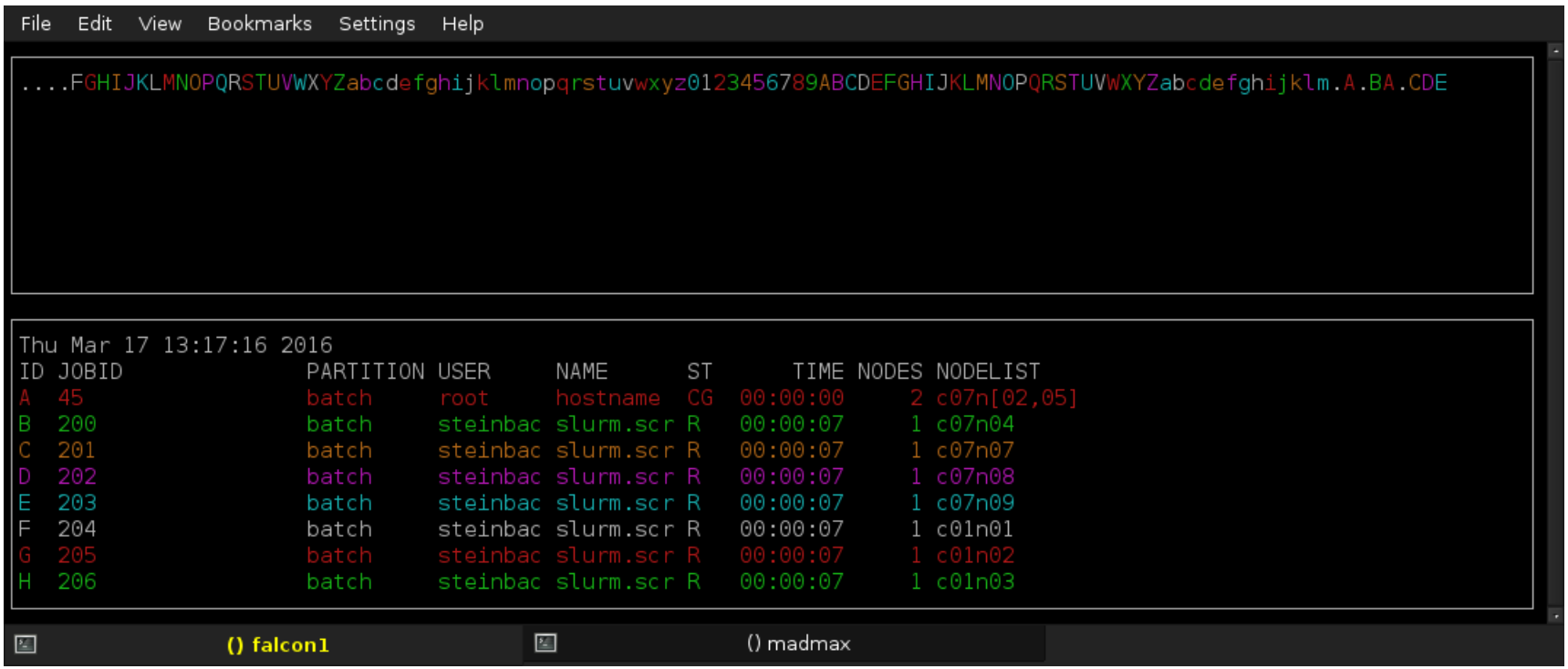
- as every SLURM command, squeue very versatile
- -o supports 54 different columns types

Or you can have something more fancy . . .

SLURM : Monitoring 2

ncurses based command-line monitoring

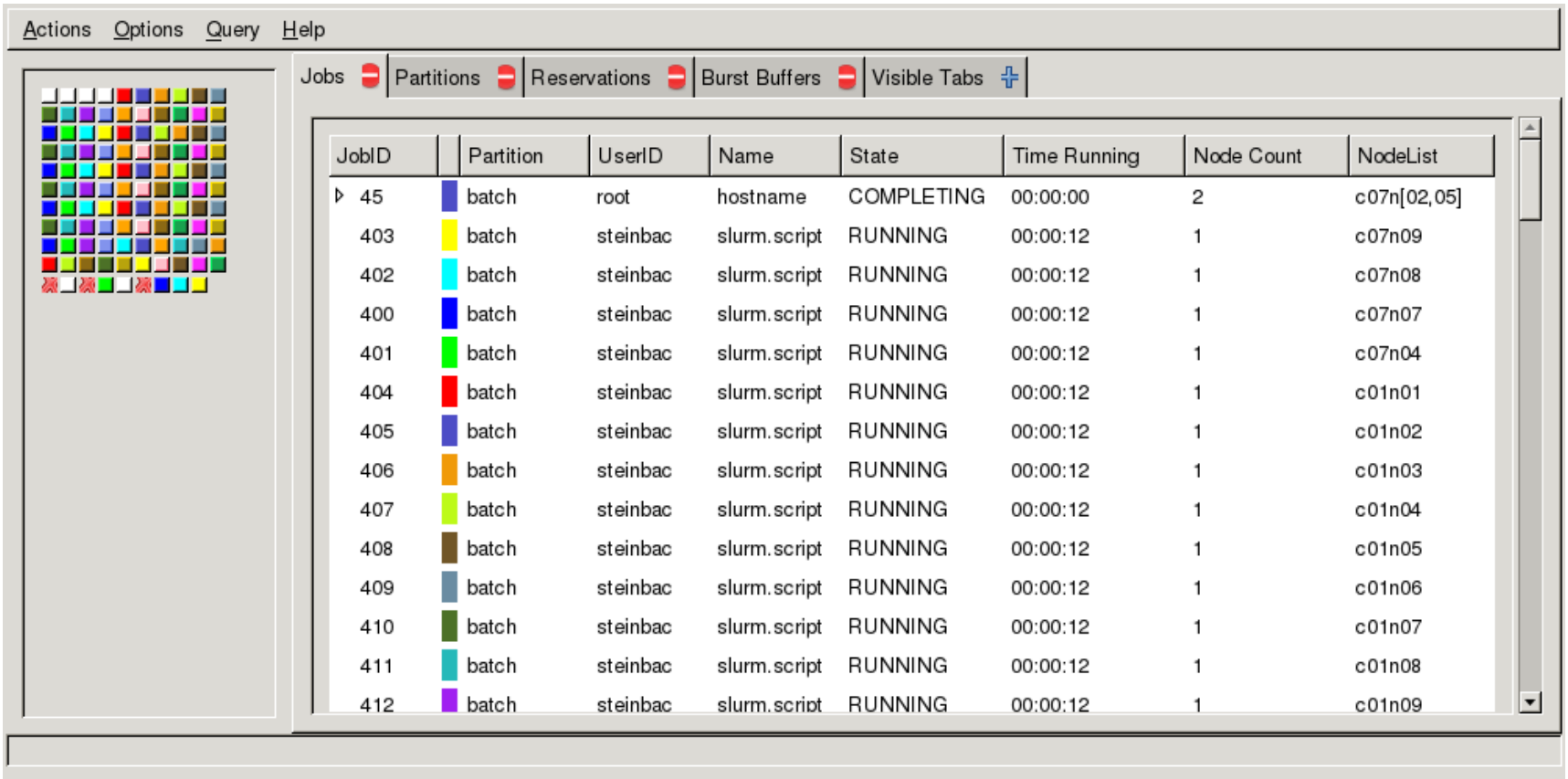
```
$ smap
```



Or something even more fancy . . .

GTK2 based GUI (Xorg session required)

```
$ sview
```



JobID	Partition	UserID	Name	State	Time Running	Node Count	NodeList
45	batch	root	hostname	COMPLETING	00:00:00	2	c07n[02,05]
403	batch	steinbac	slurm.script	RUNNING	00:00:12	1	c07n09
402	batch	steinbac	slurm.script	RUNNING	00:00:12	1	c07n08
400	batch	steinbac	slurm.script	RUNNING	00:00:12	1	c07n07
401	batch	steinbac	slurm.script	RUNNING	00:00:12	1	c07n04
404	batch	steinbac	slurm.script	RUNNING	00:00:12	1	c01n01
405	batch	steinbac	slurm.script	RUNNING	00:00:12	1	c01n02
406	batch	steinbac	slurm.script	RUNNING	00:00:12	1	c01n03
407	batch	steinbac	slurm.script	RUNNING	00:00:12	1	c01n04
408	batch	steinbac	slurm.script	RUNNING	00:00:12	1	c01n05
409	batch	steinbac	slurm.script	RUNNING	00:00:12	1	c01n06
410	batch	steinbac	slurm.script	RUNNING	00:00:12	1	c01n07
411	batch	steinbac	slurm.script	RUNNING	00:00:12	1	c01n08
412	batch	steinbac	slurm.script	RUNNING	00:00:12	1	c01n09

SLURM : Control

In case something goes wrong

```
$ scancel <job-id>
```

In case everything goes wrong

```
$ scancel -u $USER
```

SLURM : Open Questions

- SLURM does not require anything like queues
- some clusters can live without them
  (but require walltime limit, memory limits, ...)

- **We suggest:**
  To clone the queues we have in `madmax` and set them up in furiosa!

SLURM : Open Questions

- SLURM does not require anything like queues
- some clusters can live without them
  (but require walltime limit, memory limits, ...)

- **We suggest:**
  To clone the queues we have in `madmax` and set them up in furiosa!

What do you think?

Outline

Cluster Expansion

- budget available for cluster expansion in 2017
- question: What to use it for?

Cluster Expansion

- budget available for cluster expansion in 2017
- question: What to use it for?

More CPUs

More Disk

GPUs*

What do our users think?

Outline

HPC Consolidation

- keep madmax operational
- reinstall OS to match furiosa's and upgrade to use SLURM
- ...

What do our users think?