XT3 Operational Enhancements

Chad Vizino <vizino@psc.edu>
Pittsburgh Supercomputing Center

CUG 2006

Overview

- Job scheduling environment
  - See CUG 2005 Proceedings for detail
- Job Specific Console Logging
- Event Handling
- Graphical Monitor

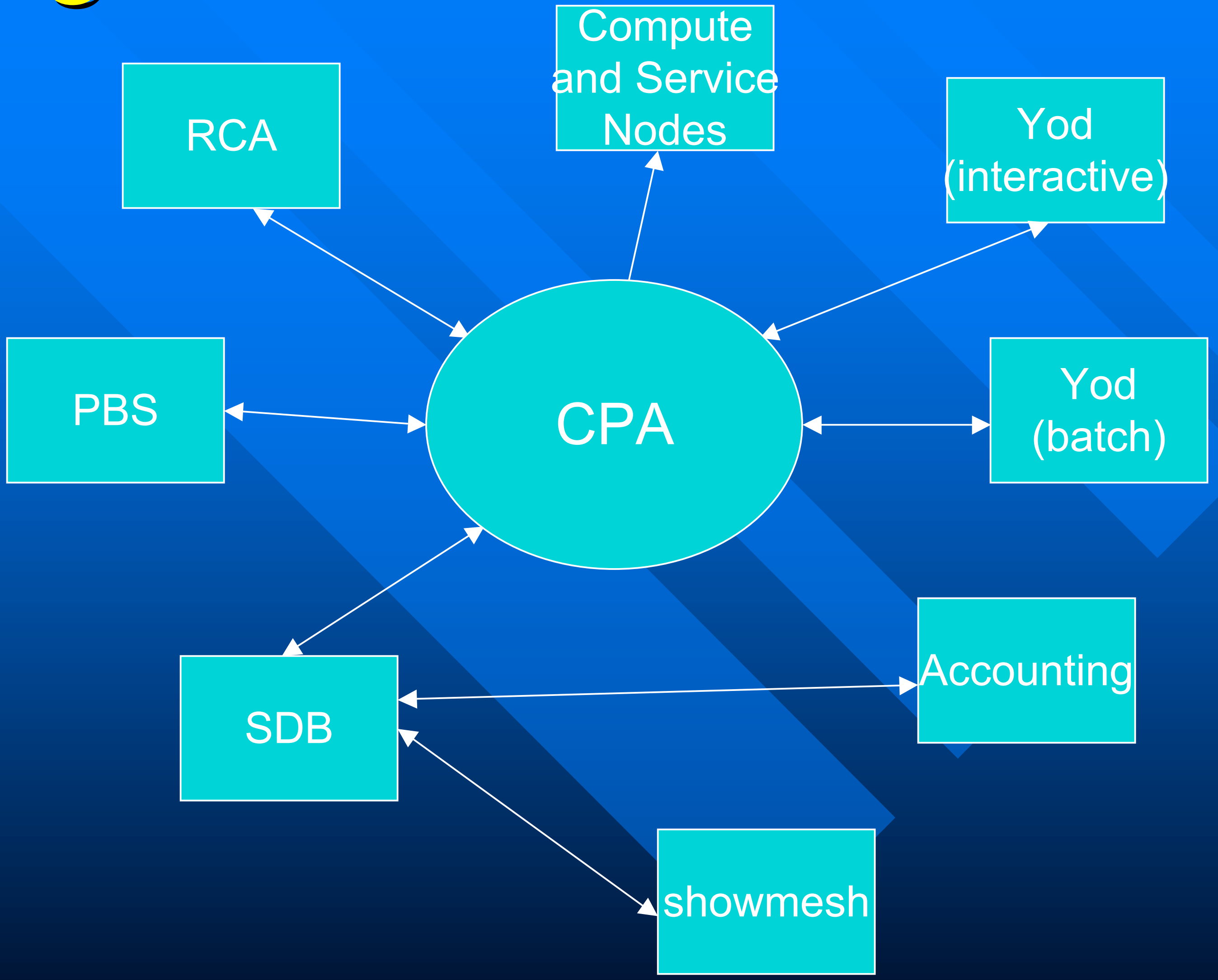
Job life cycle with Tools

- User submits job
- Job runs on pre-checked nodes selected by scheduler
- Job can be monitored visually for placement
- User gets
  - Job output
  - Console output
- Admins get node failure alerts if any

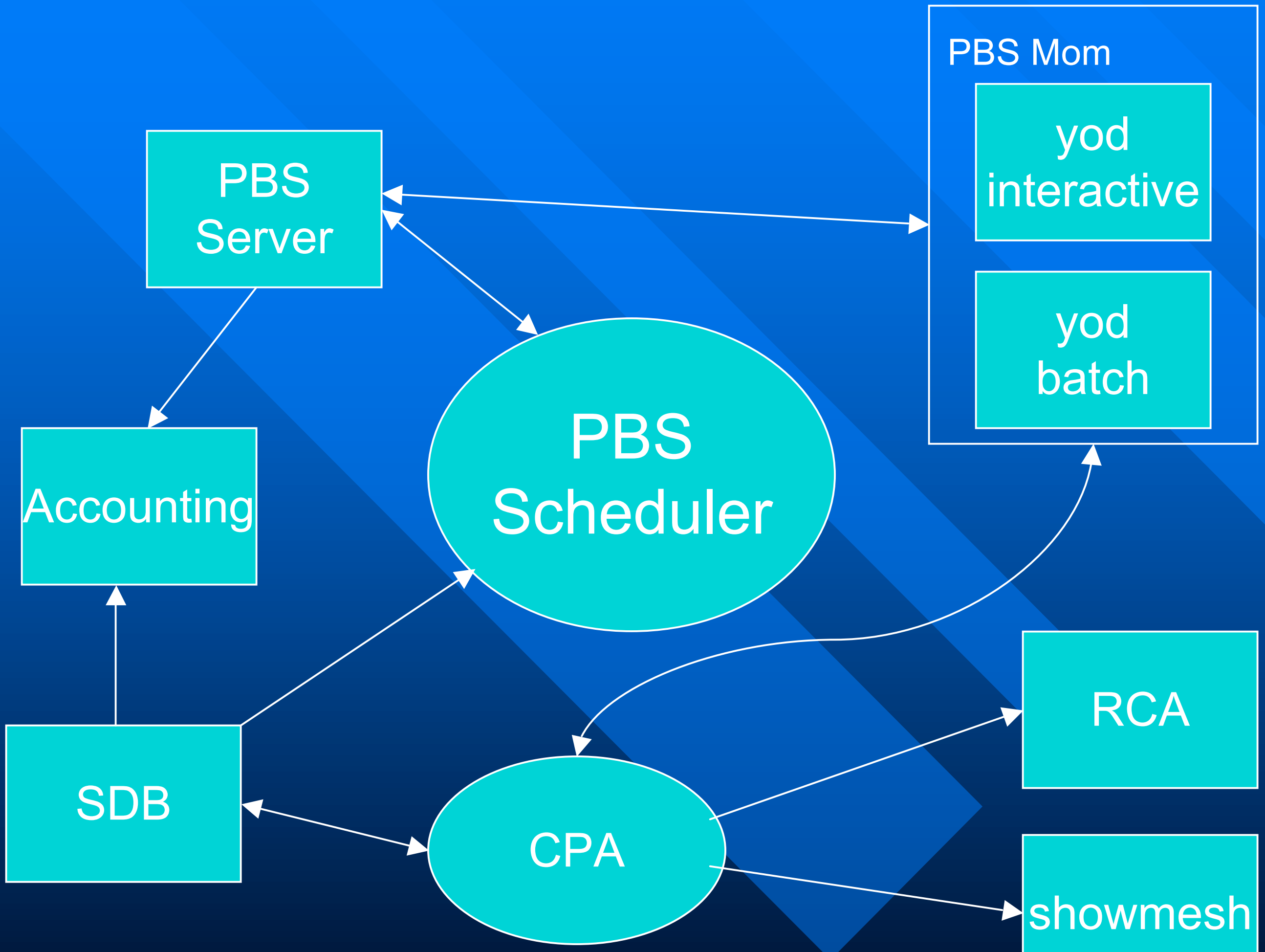
PSC PBS Changes

- Base changes to support nid addressing
- Nid-addressable scheduler
- Prologue changes
  - To support node pre-job scanning
    - Ping_list
  - To support job console log management
- Epilogue changes
  - To support job console log management

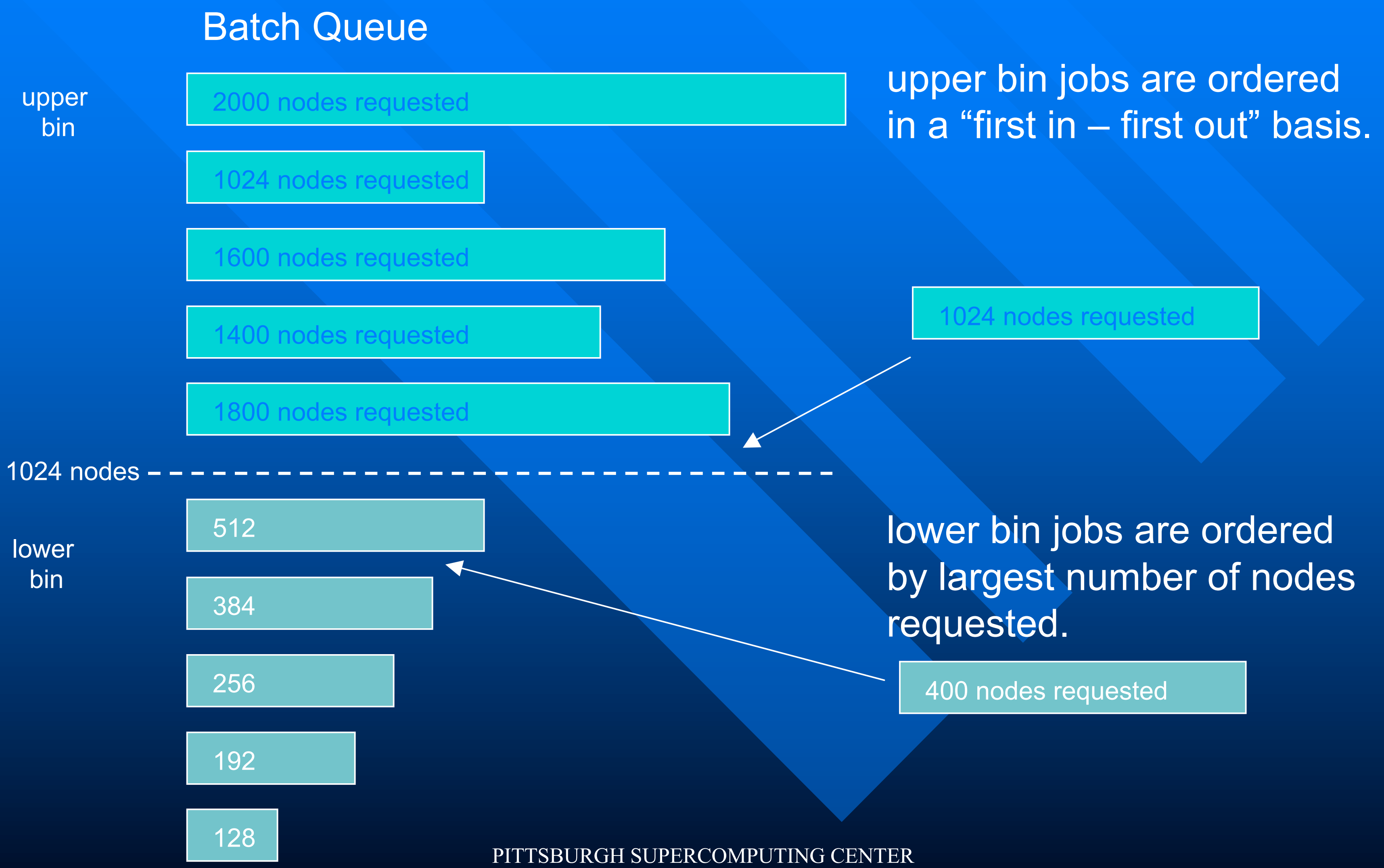
Original Allocation Architecture



Customized Architecture



PSC Scheduling



Batch Queue

upper bin

2000 nodes requested

1024 nodes requested

1600 nodes requested

1400 nodes requested

1800 nodes requested

upper bin jobs are ordered in a “first in – first out” basis.

1024 nodes requested

1024 nodes

lower bin

512

384

256

192

128

lower bin jobs are ordered by largest number of nodes requested.

400 nodes requested

Related PSC XT3 Talk

- Deborah Weisser
- Application placement within the XT3
- 2:30pm, Room B1

XT3 Console Log

- Contains each node’s console output
- Provides trace information for user
- Provides useful information for SPR in event of failure
- Huge
- Not specific to job
- Hard to read and decode visually

Job Specific Console Logging Goals

- Direct each console log entry to file related to user job
- Provide user with supplemental information about job
- Provide admins and Cray with predictable place to look for system related problems encountered during job
- Low overhead on system resources
- Readable output

Job Specific Console Logging Implementation

- xtconsole_watcher on smw
  - Starts at smw boot time, persistent after that
- Daemon watches xtconsole output
- Nid to job id mapping maintained in file on smw for use by event handler daemon
- Hardware name to nid translation
- Group messages by job

Translate physnid=0x to int

Before

```
[2006-02-24 09:35:14][c8-1c1s0n1]******* _cstart2(), yod_pid=1243
rank=65 lognid=65 physnid=0x8a1 pid=3
```

After

```
[2006-02-24 09:35:14][2209][c8-1c1s0n1]******* _cstart2(), yod_pid=1243
rank=65 lognid=65 physnid=2209 pid=3
```

Translate nid = 0x… to int

Before

```
[2006-02-06 12:52:01][c0-0c2s2n2]607f 5
look_for_first_message_timeout(), Rank 1 did not reply.   [nid = 0x4b]
```

After

```
[2006-02-06 12:52:01][  74][c0-0c2s2n2]607f 5
look_for_first_message_timeout(), Rank 1 did not reply.   [nid = 75]
```

Prologue

- Put started tag into console log
  - PBS <jid> started <time> <nid_list>
  - Split line into 70 characters
  - Read by xtconsole_watcher
- Perform pre-scan

PBS started Tag

```
[2006-02-23 12:54:39][   7][c0-0c0s1n3]PBS 19966 started
    1140717279 79..95,152..188,285..324,465..479,512..607,64+

    …

[2006-02-23 12:54:39][   7][c0-0c0s1n3]PBS 19966
    2048..2143,2176..2271,2304..2399,2432..2527,2560..2655,26
    88..2723
```

PBS pre-scan Tags

```
[2006-02-23 12:54:39][  79][c0-0c2s3n3]recv'd
    "pbs_19966_prescan" from portal 9, len 18, nid 7 pid
    [107], flag= 0x0
        ...

[2006-02-23 12:54:40][2615][c10-0c1s5n3]recv'd
    "pbs_19966_prescan" from portal 9, len 18, nid 7 pid
    [107], flag= 0x0

[2006-02-23 12:54:40][   7][c0-0c0s1n3]PBS_PRESCAN
    19966 tg-login2 these nids failed check 285
```

Epilogue

- Put ended tag into console log
  - PBS <jid> ended <time>
  - Read by xtconsole_watcher
- Copy job_<jid>_console.log to login node in ~<user>/job_console_logs/
- Remove job console log file from smw

PBS end Tag

```
[2006-02-23 12:55:00][   7][c0-0c0s1n3]PBS 19966 ended 1140717300
```

Console Message Decoding

Algorithm:

```
Open xtconsole stream {
    Decode line
            Watch for PBS tags (started, ended)
            Update nid to job id map file for event handler
            Map nid to file descriptor
    Open file descriptor for job
    Route console messages to file descriptor
    Close file descriptor
}
```

lsof

```
laserloop:~ # ssh boot qstat -r

phantom.psc.edu:
                                           Time In Req'd  Req'd   Elap
Job ID Username Queue    Jobname    SessID  Queue  Nodes  Time  S Time
------ -------- -------- ---------- ------ ------- ------ ----- - -----
21077  kspiegel batch    job_nose3    9475  020:22     64 06:00 R 01:02
21104  dalperar batch    cpkA         9498  018:34   1536 05:50 R 01:23
21114  mho      batch    job-prod3    9109  018:19    128 06:00 R 01:23
21182  ivanov   batch    pscript     12384  013:37    160 06:00 R 00:59
21189  paramore batch    u4qas1box.   9608  012:42    128 02:30 R 00:22

Total generic compute nodes allocated: 2016
laserloop:~ # ps -ef|grep xtconsole_w
root     21155     1  0 Feb22 ?        00:00:53 /usr/psc/bin/perl /usr/local/bin/xtconsole_watcher -t
laserloop:~ # lsof -p 21155
COMMAND     PID USER   FD   TYPE DEVICE     SIZE     NODE NAME
        ...
xtconsole 21155 root    0r   CHR    1,3             67706 /dev/null
xtconsole 21155 root    1w   REG    3,5 5842485  2851228 /usr/local/bin/xtconsole_watcher.log
xtconsole 21155 root    2w   REG    3,5 5842485  2851228 /usr/local/bin/xtconsole_watcher.log
xtconsole 21155 root    3r  FIFO    0,6            709418 pipe
xtconsole 21155 root    4w   REG    3,5  434428  9126852 /tmp/job_console_logs/job_21104_console.log
xtconsole 21155 root    5w   REG    3,5   29053  9126853 /tmp/job_console_logs/job_21114_console.log
xtconsole 21155 root    6w   REG    3,5   18411  9126841 /tmp/job_console_logs/job_21077_console.log
xtconsole 21155 root    7w   REG    3,5   46161  9126850 /tmp/job_console_logs/job_21182_console.log
xtconsole 21155 root    8w   REG    3,5   27211  9126845 /tmp/job_console_logs/job_21189_console.log
```

Final job console log

```
[2006-02-14 13:39:15][   7][c0-0c0s1n3]PBS 18936 started 1139942355 74..75
[2006-02-14 13:39:15][  74][c0-0c2s2n2]recv'd "pbs_18936_prescan" from portal
     9, len 18, nid 7 pid [107], flag= 0x0
[2006-02-14 13:39:15][  75][c0-0c2s2n3]recv'd "pbs_18936_prescan" from portal
     9, len 18, nid 7 pid [107], flag= 0x0
[2006-02-14 13:40:43][  74][c0-0c2s2n2]******* _cstart2(), yod_pid=20514
     rank=0 lognid=0 physnid=74 pid=3
[2006-02-14 13:40:43][  75][c0-0c2s2n3]******* _cstart2(), yod_pid=20514
     rank=1 lognid=1 physnid=75 pid=2
[2006-02-14 13:40:43][  75][c0-0c2s2n3]received final app termination, pid=2
[2006-02-14 13:40:48][  74][c0-0c2s2n2]received final app termination, pid=3
[2006-02-14 14:10:03][   7][c0-0c0s1n3]PBS 18936 ended 1139944203
```

Daemon load

- CPU consumption is about 53 seconds in about 10 days (<<1%)
- Memory consumption is 5MB out of 2GB (<1%)—without memory leak!
- File descriptor use is 4+N where N is the number of running jobs (capped at 75 for safety)
- Plus extra xtconsole
  - 54 seconds in about 10 days
  - 1MB memory

CRMS Events

- Event generated for compute node failure
- Useful for determining when failure occurred
- Not tied to job information
- Duplicate events can be generated
- Not captured in way we can track failure

Event Handling Goals

- Notify admins of component failures
- Throttle event messages to admins
- Low overhead
- Update events in local assets DB

Event Handling Implementation

- xtconsumer_watcher on smw
  - Starts at smw boot time, persistent after that
- Daemon watches xtconsumer output for specific events
  - ec_node_failed
- Nid to job id mapping obtained from file on smw maintained by job console logging daemon
- Event aggregation
- Event e-mails show event object, time, reason and job detail if it can be determined
- Update Assets DB

Decoding an event

```
2006-03-02 14:00:05|2006-03-02 14:00:05|0x40008063 -
    ec_node_failed|src=:1:s0|::c1-1c2s1n2
```

- Field separator is |
- Field 0 – time received by xtconsumer
- Field 1 – time event occurred
- Field 2 – event description – hex code is decoded by xtconsumer to string
- Field 3 – source of event
- Field 4 – target node that failed

Event Message Handling

Algorithm:

```
Open xtconsumer stream {
    Wait for node failed event or time out
    Decode line
            Record time received
            Ignore duplicates
            Get owning job id from nid to job id map file
            Get yod info from SDB
            Store in buffer for aggregation
    Flush buffer (send mail, update Assets DB) if…
            Oldest message > THRESHOLD and
            Last send > THRESHOLD
}
```

Event Mail Message

```
Mar 02 14:00:05 Node 454 (c1-1c2s1n2), "node failed", owned by PBS job id 21023
    2006-03-02 14:00:05|2006-03-02 14:00:05|0x40008063 -
    ec_node_failed|src=:1:s0|::c1-1c2s1n2

Job id: 21023
Owner: blood
Group: mtsmupp
Job Name: nbar_dops10x2.2
Walltime Used: 00:01:38
Walltime Req'd: 03:00:00
Size: 256
Nid List: 454,456..479,512..547,550..586,588..590,592..607,644..735,768..803,805..815
Cmd:
job_id partition_id yod_id num_of_compute_processors command
21023.phantom.psc.edu 18422 31595 256 yod -small_pages namd2 d.namd
```

Daily Update – Cross check

```
Compute node failures
etimestamp processor_id processor_type
2006-03-02 00:04:16 36 c
2006-03-02 00:06:58 16 c
2006-03-02 07:56:10 455 c
2006-03-02 12:03:25 76 c
2006-03-02 12:03:37 77 c
2006-03-02 12:03:49 78 c
2006-03-02 14:00:05 454 c
Owning jobs (ones that were running when failures occurred)
JobID JobName User Queue Started Ended Cpus TimeReq TimeUsed Exit Nids
INFO: job 20995 used 36
20995 nbar_dops.10x2. blood res 03/02 00:02:22 03/02 00:04:45 256 04:30:00
    00:02:23 1
    36..41,43..77,80..83,93..95,152..154,164..223,256..257,260..267,270..303,305..3
    51,384..437
      command = "yod -small_pages namd2 e.namd"
INFO: job 21009 used 16
21009 nbar_dops10x2.2 blood batch 03/02 00:05:08 03/02 00:07:27 256 03:00:00
    00:02:19 1
    16,37..41,43..77,80..83,93..95,152..154,164..223,256..257,260..267,270..303,305
    ..351,384..437
     command = "yod -small_pages namd2 d.namd" INFO: job 20832 used 455
20832 ch_eq_noNas pantano batch 03/02 07:55:14 03/02 07:56:40 256 06:00:00
    00:01:26 0
    455..479,512..547,550..586,588..590,592..607,644..735,768..803,805..815
     command = "yod /usr/local/packages/namd/NAMD_2.6b1_Cray_XT3/namd2 namd-
    noNas.conf"
INFO: job 21023 used 454
21023 nbar_dops10x2.2 blood batch 03/02 13:58:09 03/02 14:00:40 256 03:00:00
    00:02:31 1
    454,456..479,512..547,550..586,588..590,592..607,644..735,768..803,805..815
    command = "yod -small_pages namd2 d.namd"
```

Daemon load

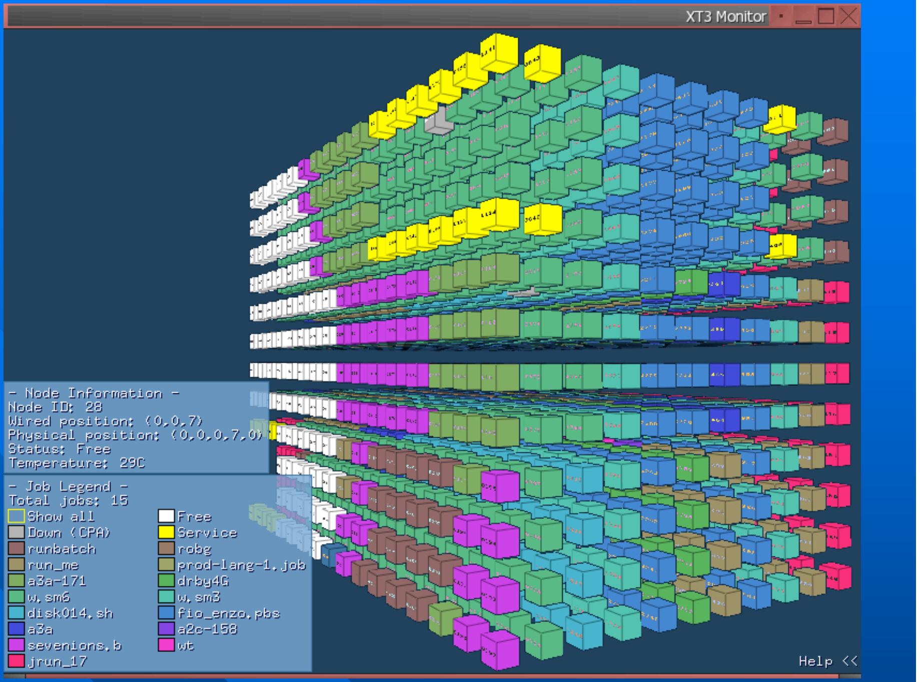
- CPU consumption approx. <1 second in about 8 days (<<1%)
- Memory consumption is 8.5MB out of 2GB (<1%)
- Plus extra xtconsumer process
  - <1 second in about 8 days
  - 1MB memory

Things to think about

- Message floods
  - Power failure
- Job kills
  - Stacked yod jobs
- Diagnosing further failure
  - Other event types
- Mail events on pre-scan, reaper failure

Graphical Monitor

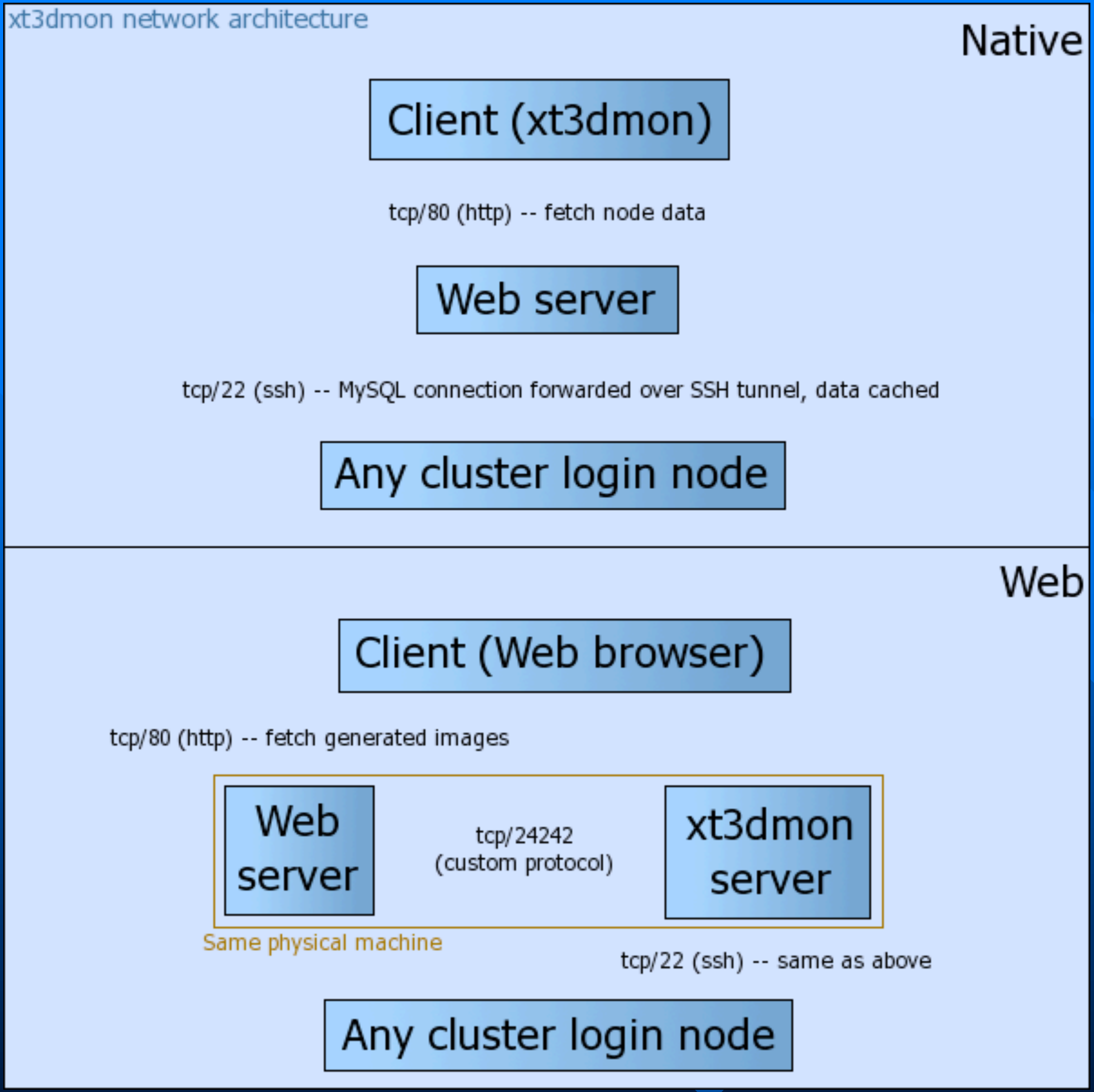
- Visualize torus in 3-d
- Visualize physical layout
- Show machine characteristics
- 13,000 lines of C
- OpenGL and GLUT
- Portable (Windows, Linux, Mac)
- Web enabled front-end for low end systems

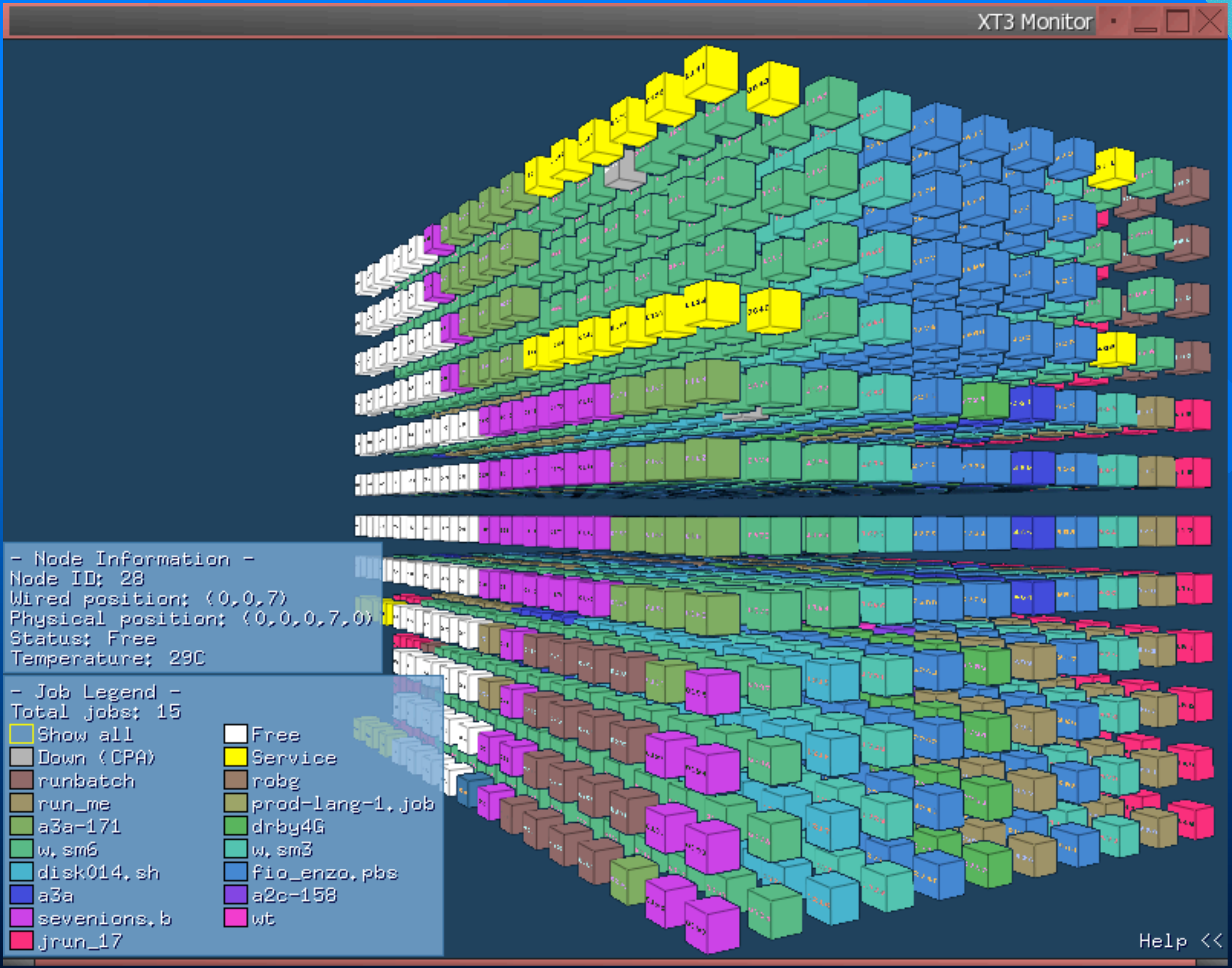


```
- Node Information -
Node ID: 28
Wired position: (0,0,7)
Physical position: (0,0,0,7,0)
Status: Free
Temperature: 29C
```

- Job Legend -
Total jobs: 15

□ Show all	⬜ Free
⬜ Down (CPA)	🟨 Service
🟫 runbatch	🟫 robg
🟫 run_me	🟩 prod-lang-1.job
🟩 a3a-171	🟩 drby4G
🟩 w.sm6	🟩 w.sm3
🟦 disk014.sh	🟦 fio_enzo.pbs
🟦 a3a	🟪 a2c-158
🟪 sevenions.b	🟪 wt
🟥 jrun_17	

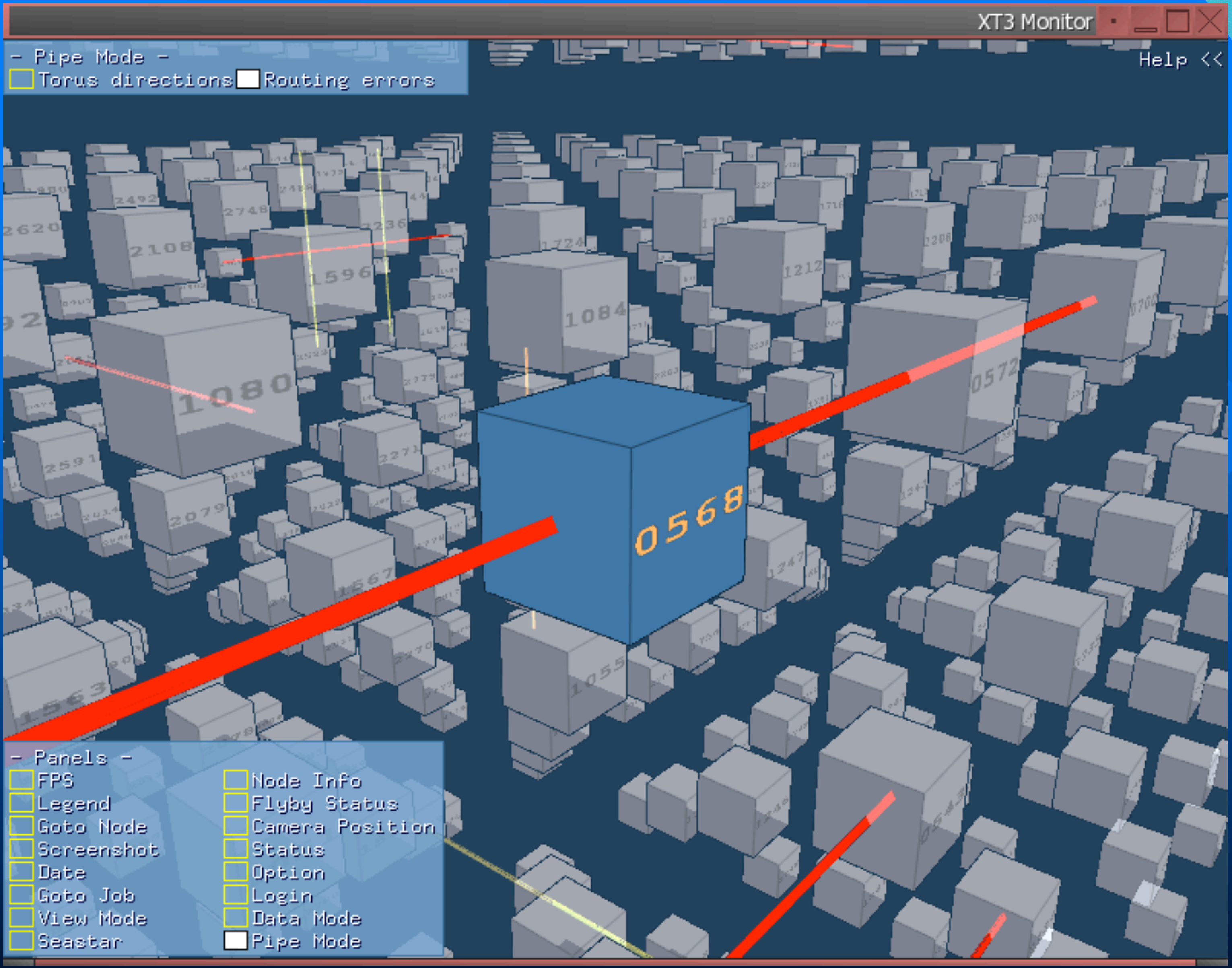

xt3dmon network architecture
Native
Client (xt3dmon)
tcp/80 (http) -- fetch node data
Web server
tcp/22 (ssh) -- MySQL connection forwarded over SSH tunnel, data cached
Any cluster login node
Web
Client (Web browser)
tcp/80 (http) -- fetch generated images
Web server
tcp/24242 (custom protocol)
xt3dmon server
Same physical machine
tcp/22 (ssh) -- same as above
Any cluster login node



- Node Information -
Node ID: 28
Wired position: (0,0,7)
Physical position: (0,0,0,7,0)
Status: Free
Temperature: 29C

- Job Legend -
Total jobs: 15

- □ Show all
- ⬜ Down (CPA)
- 🟫 runbatch
- 🟫 run_me
- 🟩 a3a-171
- 🟩 w.sm6
- 🟦 disk014.sh
- 🟦 a3a
- 🟪 sevenions.b
- 🟥 jrun_17
- ⬜ Free
- 🟨 Service
- 🟫 robg
- 🟫 prod-lang-1.job
- 🟩 drby4G
- 🟩 w.sm3
- 🟦 fio_enzo.pbs
- 🟪 a2c-158
- 🟪 wt



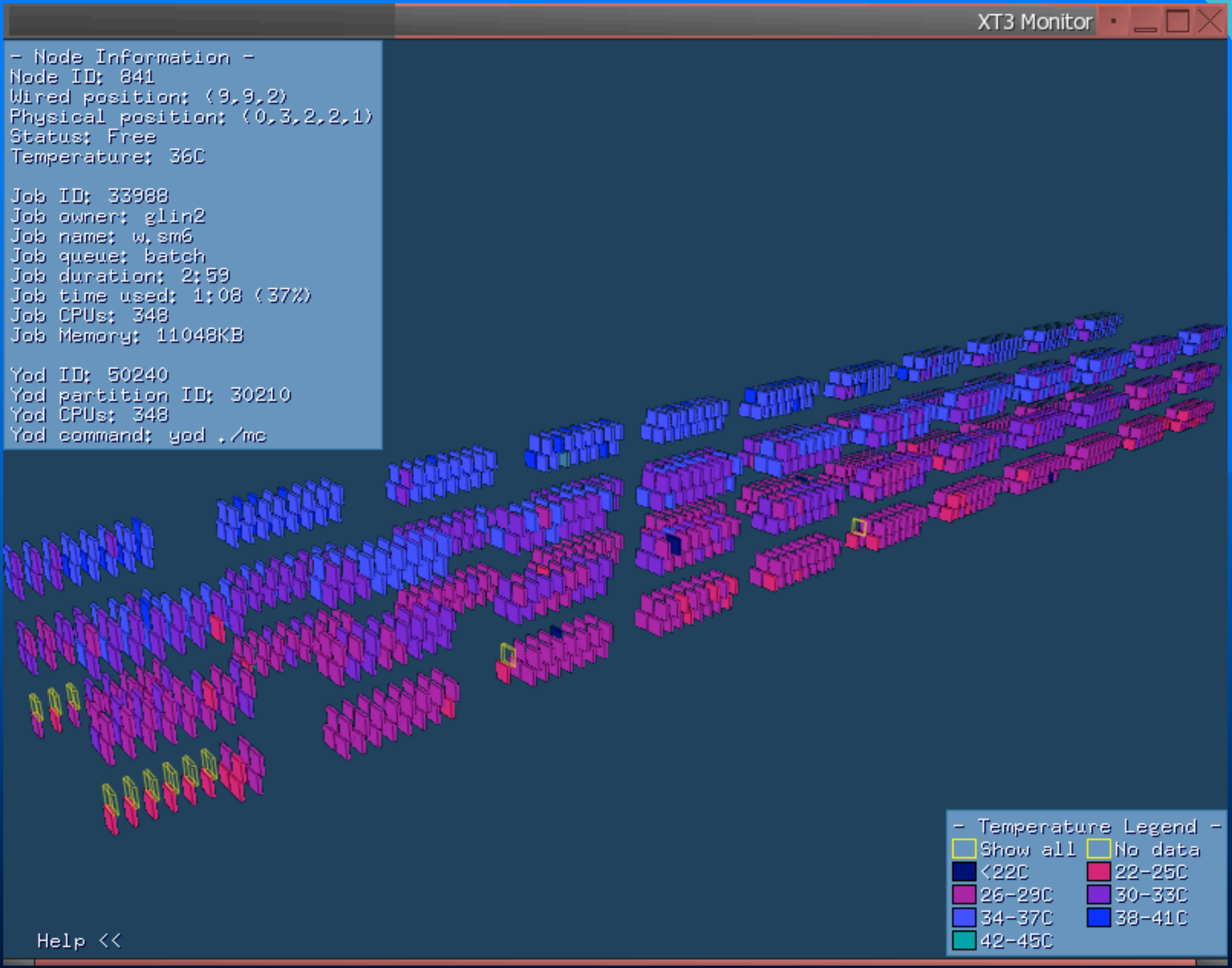
- Pipe Mode -

□ Torus directions ■ Routing errors

- Panels -

□ FPS
□ Legend
□ Goto Node
□ Screenshot
□ Date
□ Goto Job
□ View Mode
□ Seastar

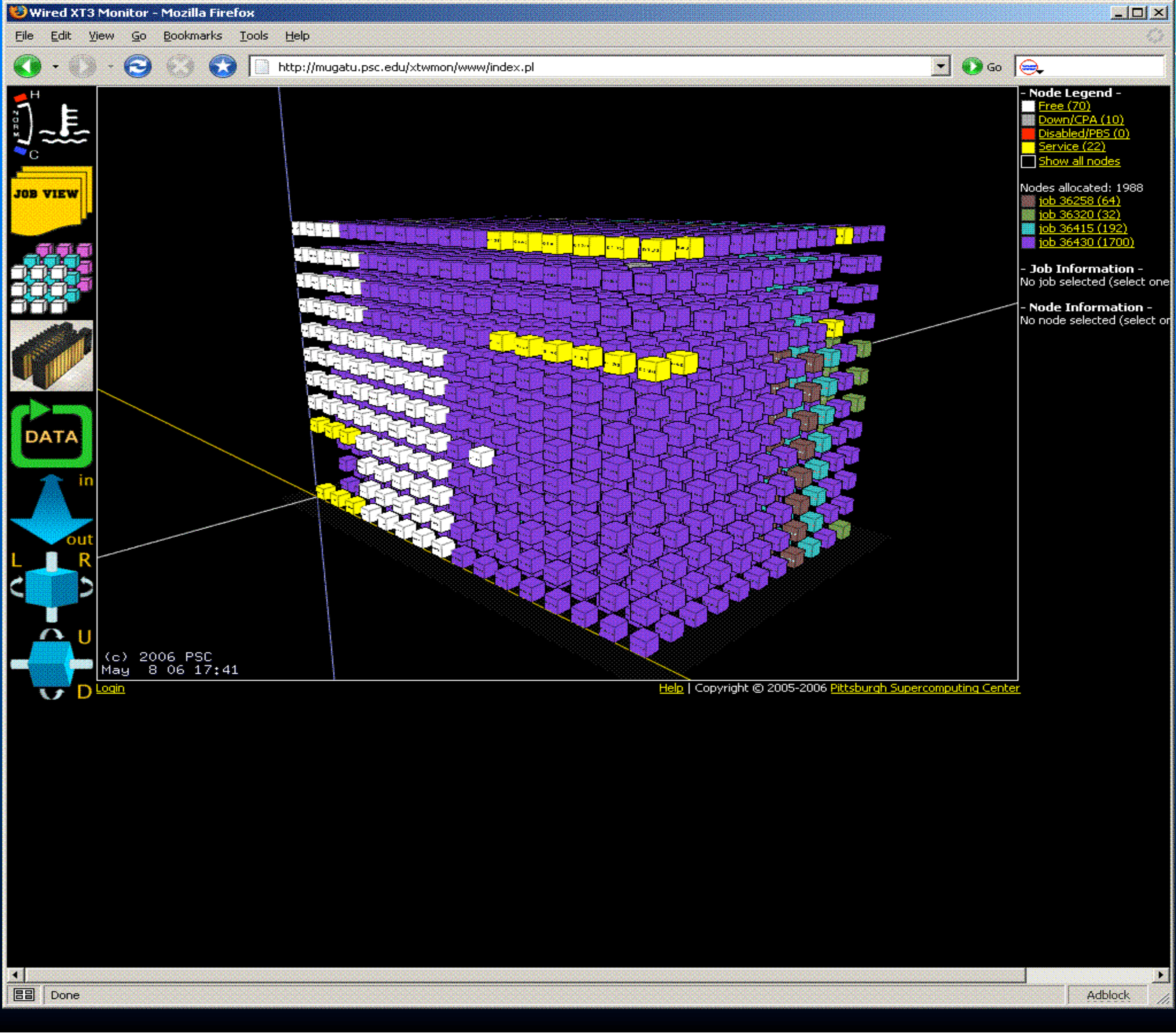
□ Node Info
□ Flyby Status
□ Camera Position
□ Status
□ Option
□ Login
□ Data Mode
■ Pipe Mode



```
- Node Information -
Node ID: 841
Wired position: (9,9,2)
Physical position: (0,3,2,2,1)
Status: Free
Temperature: 36C

Job ID: 33988
Job owner: glin2
Job name: w.sm6
Job queue: batch
Job duration: 2:59
Job time used: 1:08 (37%)
Job CPUs: 348
Job Memory: 11048KB

Yod ID: 50240
Yod partition ID: 30210
Yod CPUs: 348
Yod command: yod ./mc
```



Wired XT3 Monitor - Mozilla Firefox

http://mugatu.psc.edu/xtwmon/www/index.pl

- Node Legend -

□ Free (70)
▨ Down/CPA (10)
🟥 Disabled/PBS (0)
🟨 Service (22)
□ Show all nodes

Nodes allocated: 1988
▨ job 36258 (64)
🟩 job 36320 (32)
▨ job 36415 (192)
🟪 job 36430 (1700)

- Job Information -
No job selected (select one

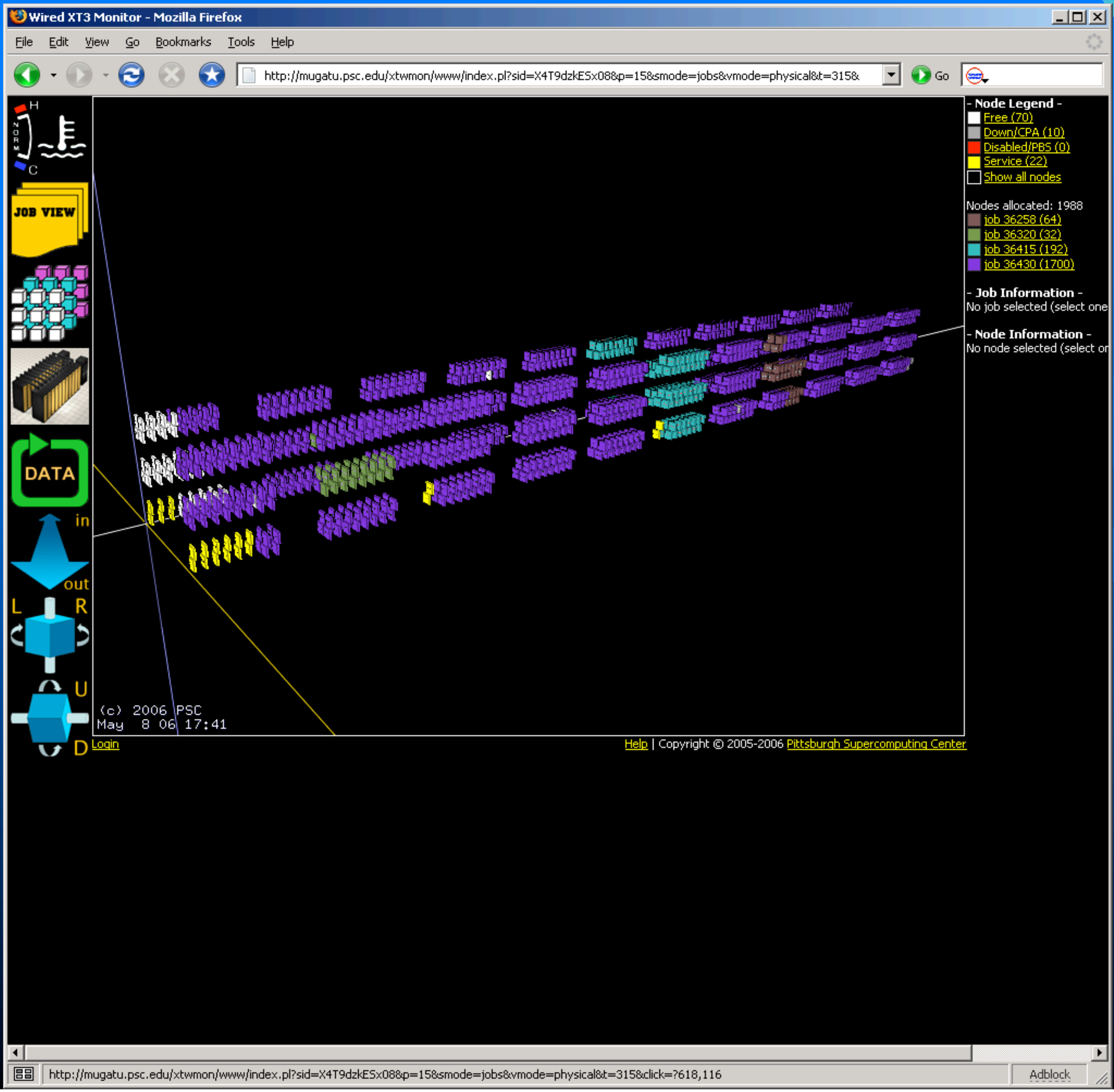
- Node Information -
No node selected (select or

(c) 2006 PSC
May 8 06 17:41

Login

Help | Copyright © 2005-2006 Pittsburgh Supercomputing Center

Done



Wired XT3 Monitor - Mozilla Firefox

http://mugatu.psc.edu/xtwmon/www/index.pl?sid=X4T9dzkE5x08&p=15&smode=jobs&vmode=physical&t=315&

- Node Legend -
□ Free (70)
▨ Down/CPA (10)
🟥 Disabled/PBS (0)
🟨 Service (22)
□ Show all nodes

Nodes allocated: 1988
🟫 job 36258 (64)
🟩 job 36320 (32)
🟦 job 36415 (192)
🟪 job 36430 (1700)

- Job Information -
No job selected (select one

- Node Information -
No node selected (select or

(c) 2006 PSC
May 8 06 17:41

Login

Help | Copyright © 2005-2006 Pittsburgh Supercomputing Center

http://mugatu.psc.edu/xtwmon/www/index.pl?sid=X4T9dzkE5x08&p=15&smode=jobs&vmode=physical&t=315&click=?618,116

Summary

- PSC tools and enhancements provide
  - Users with information useful for the interperatation of their runs
    - » Console output
    - » Visual job placement
  - Admins with effective tools to utilize and gather information from machine
    - » Custom scheduling environment
    - » Console output
    - » Event management