Fun with Grid Engine XML

Hello!

- I'm Chris
  - 'dag@sonsorol.org' (public)
  - 'chris@bioteam.net' (corporate)
- I work for the BioTeam
  - http://bioteam.net
  - Independent consultant shop
  - Scientists self-taught at IT
  - Bridging the science-HPC gap
- Long OSS involvement
  - http://bioperl.org
  - http://gridengine.info
  - http://xml-qstat.org

Bias Disclosure

- I’m the industry jerk
  - Cynical
  - Tight focus on practical, deployable solutions
- In my world …
  - We are not funded by sovereign nations
  - We do not have petabyte-scale filesystems & multi-gig optical WAN links
  - Don’t have 7 figure IT budgets

Background

- This is a light talk
  - Rehash of 2007 SGE Workshop Talk
- Basic intro to “doing stuff with XML”
- Talking about work published here:
  - http://xml-qstat.org
  - Simple web based SGE status dashboard
  - Not rocket science:
    - Transforms SGE qstat XML into useful XHTML

Smarter people than me …

- Petr Jung (Sun Microsystems)
    - Wrote code that allows …
        - Native SGE queries from Apache Cocoon:
        - `java/org/xmlqstat/generator/CommandGenerator`
- Mark “Mr. FLEXlm” Olesen
    - I stole sge-xml-cacher.pl code from his qlicserver
    - Mark Is integrating qlicserver with xmlqstat
        - Resulting in:
            - Much cleaner code for xml-qstat
            - Lots of new functionality (qhost data, resource data, etc.)

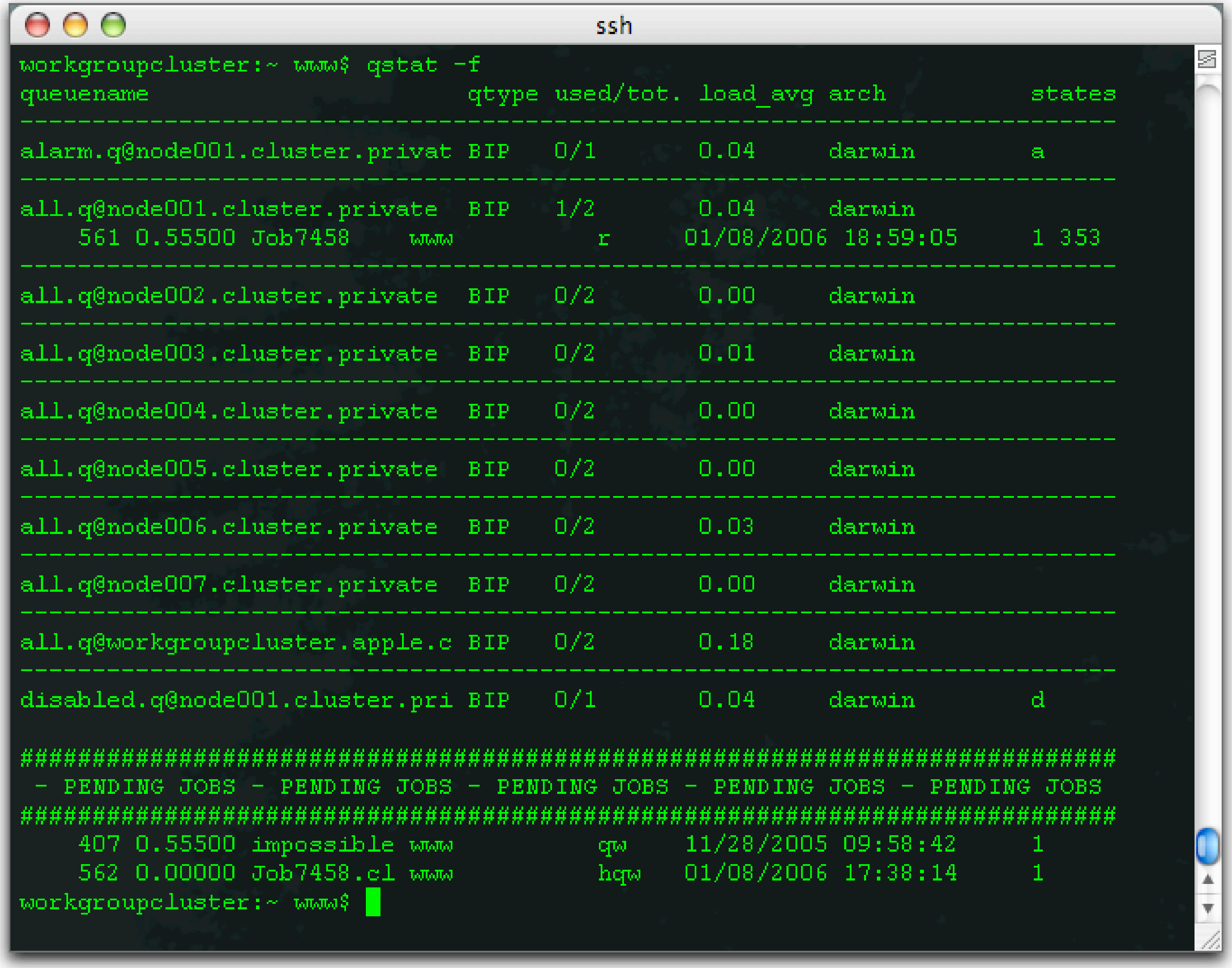
Basic need:

- Web based SGE status monitoring

Basic need:

- Web based SGE status monitoring
- The old way (pre SGE 6.0)
  1. Write a perl CGI
  2. Have it call “qstat -f”
  3. Parse the output and add HTML tags
  4. Hope things don’t break all that often

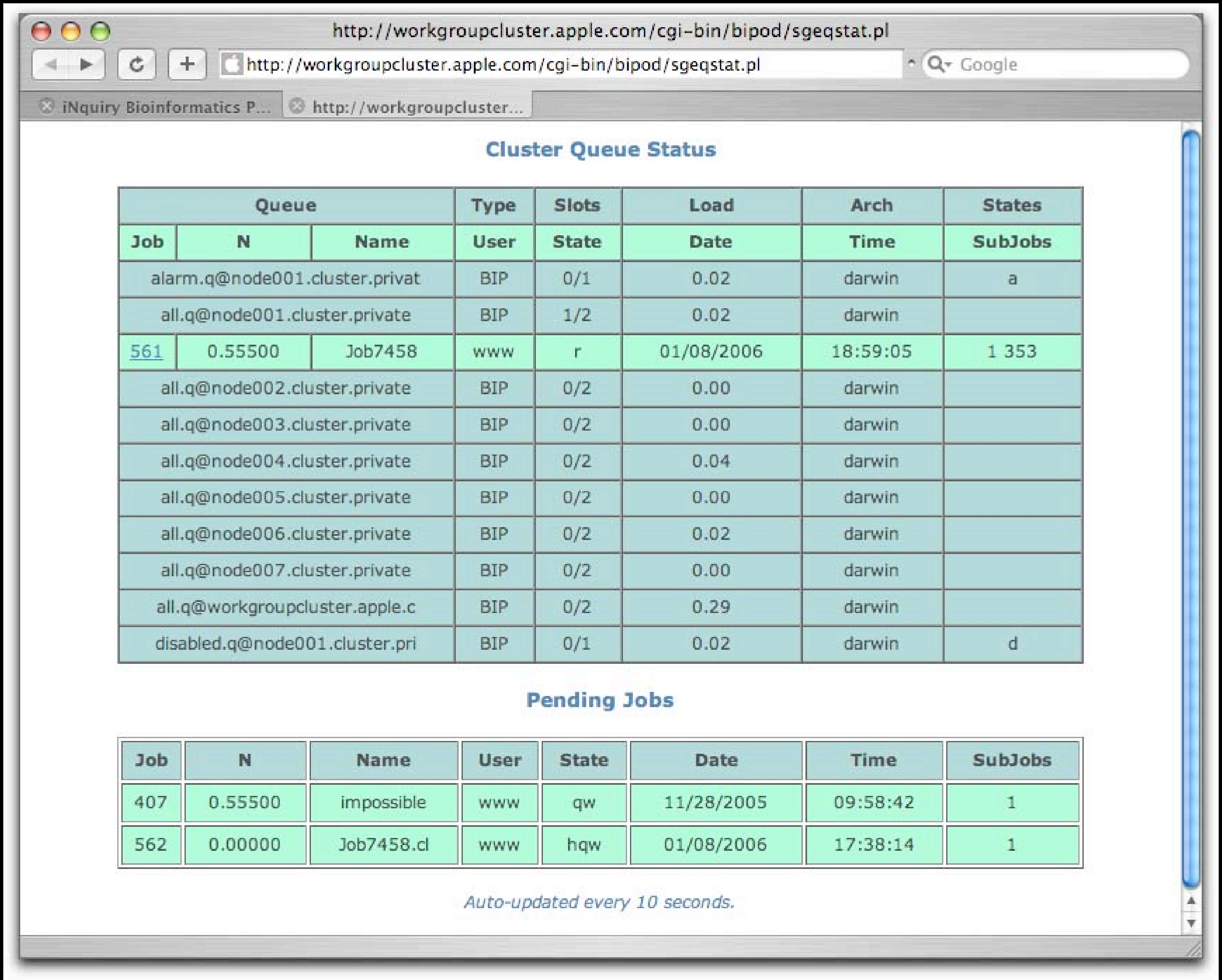
Old way ...



```
workgroupcluster:~ www$ qstat -f
queuename                      qtype used/tot. load_avg arch          states
----------------------------------------------------------------------------
alarm.q@node001.cluster.privat BIP   0/1       0.04     darwin        a
----------------------------------------------------------------------------
all.q@node001.cluster.private  BIP   1/2       0.04     darwin
    561 0.55500 Job7458    www          r     01/08/2006 18:59:05     1 353
----------------------------------------------------------------------------
all.q@node002.cluster.private  BIP   0/2       0.00     darwin
----------------------------------------------------------------------------
all.q@node003.cluster.private  BIP   0/2       0.01     darwin
----------------------------------------------------------------------------
all.q@node004.cluster.private  BIP   0/2       0.00     darwin
----------------------------------------------------------------------------
all.q@node005.cluster.private  BIP   0/2       0.00     darwin
----------------------------------------------------------------------------
all.q@node006.cluster.private  BIP   0/2       0.03     darwin
----------------------------------------------------------------------------
all.q@node007.cluster.private  BIP   0/2       0.00     darwin
----------------------------------------------------------------------------
all.q@workgroupcluster.apple.c BIP   0/2       0.18     darwin
----------------------------------------------------------------------------
disabled.q@node001.cluster.pri BIP   0/1       0.04     darwin        d

############################################################################
 - PENDING JOBS - PENDING JOBS - PENDING JOBS - PENDING JOBS - PENDING JOBS
############################################################################
    407 0.55500 impossible www          qw    11/28/2005 09:58:42     1
    562 0.00000 Job7458.cl www          hqw   01/08/2006 17:38:14     1
workgroupcluster:~ www$
```

Old way …


http://workgroupcluster.apple.com/cgi-bin/bipod/sgeqstat.pl
http://workgroupcluster.apple.com/cgi-bin/bipod/sgeqstat.pl
Google
iNquiry Bioinformatics P...
http://workgroupcluster...

Cluster Queue Status

Queue			Type	Slots	Load	Arch	States
Job	**N**	**Name**	**User**	**State**	**Date**	**Time**	**SubJobs**
alarm.q@node001.cluster.privat			BIP	0/1	0.02	darwin	a
all.q@node001.cluster.private			BIP	1/2	0.02	darwin	
561	0.55500	Job7458	www	r	01/08/2006	18:59:05	1 353
all.q@node002.cluster.private			BIP	0/2	0.00	darwin	
all.q@node003.cluster.private			BIP	0/2	0.00	darwin	
all.q@node004.cluster.private			BIP	0/2	0.04	darwin	
all.q@node005.cluster.private			BIP	0/2	0.00	darwin	
all.q@node006.cluster.private			BIP	0/2	0.02	darwin	
all.q@node007.cluster.private			BIP	0/2	0.00	darwin	
all.q@workgroupcluster.apple.c			BIP	0/2	0.29	darwin	
disabled.q@node001.cluster.pri			BIP	0/1	0.02	darwin	d

Pending Jobs

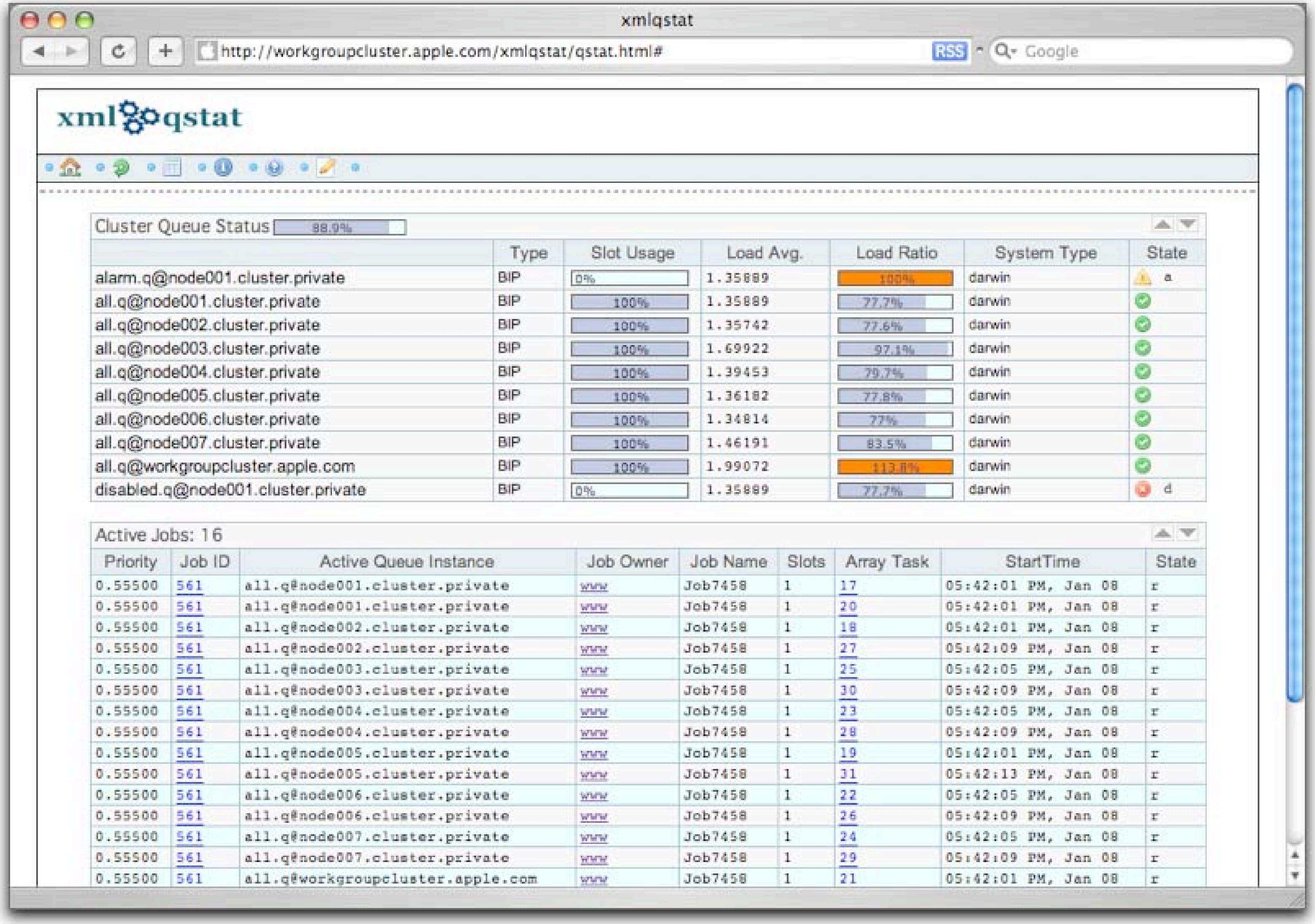
Job	N	Name	User	State	Date	Time	SubJobs
407	0.55500	impossible	www	qw	11/28/2005	09:58:42	1
562	0.00000	Job7458.cl	www	hqw	01/08/2006	17:38:14	1

Auto-updated every 10 seconds.

A new way …

- Since SGE 6.0
  - Grid Engine has supported XML output options for ‘qhost’ and ‘qstat’
- Opportunity:
  - No more parsing of output meant for human eyeballs!
  - Lets try to do something useful with SGE XML

New way …



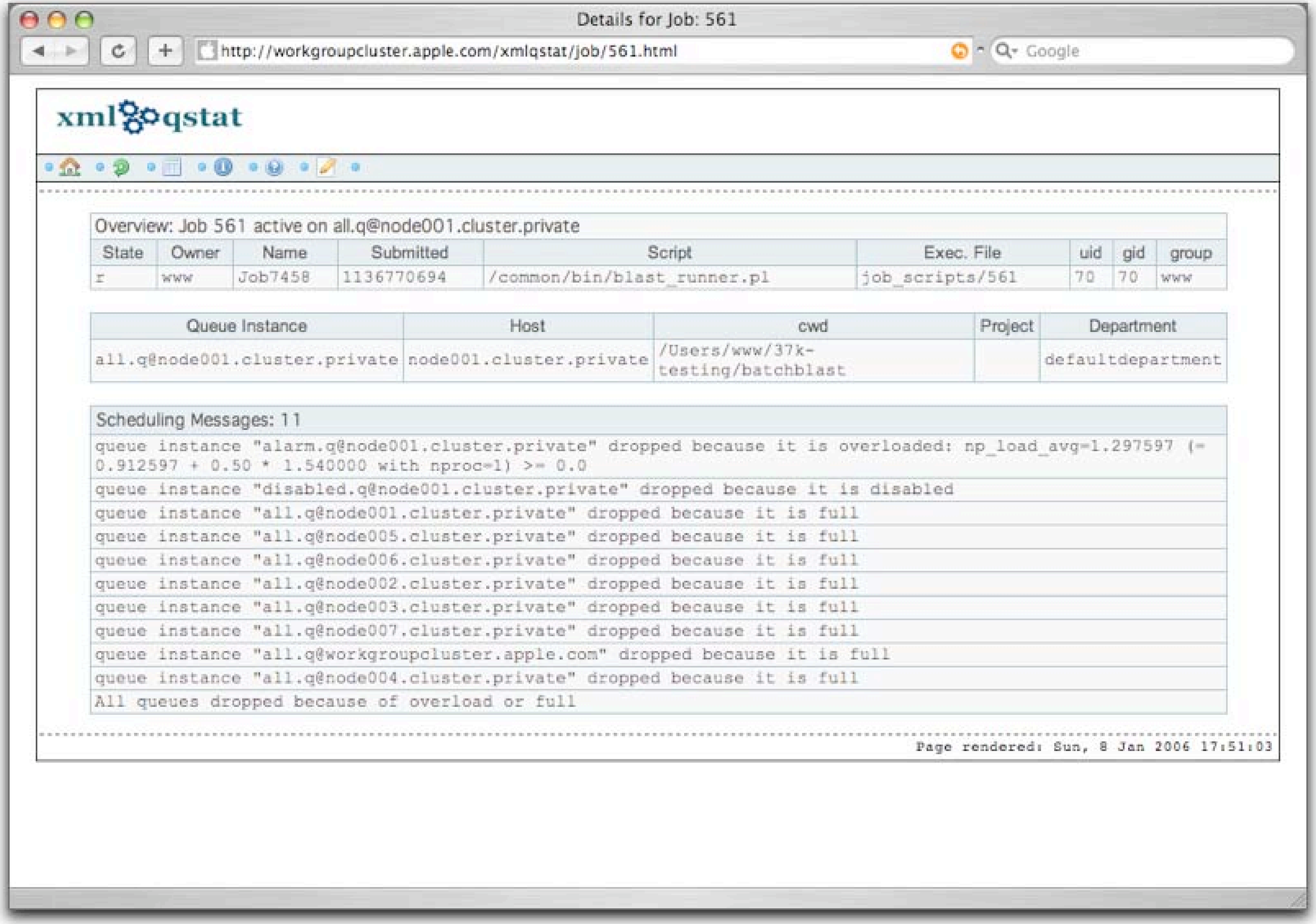
Cluster Queue Status 88.9%

	Type	Slot Usage	Load Avg.	Load Ratio	System Type	State
alarm.q@node001.cluster.private	BIP	0%	1.35889	100%	darwin	⚠ a
all.q@node001.cluster.private	BIP	100%	1.35889	77.7%	darwin	✓
all.q@node002.cluster.private	BIP	100%	1.35742	77.6%	darwin	✓
all.q@node003.cluster.private	BIP	100%	1.69922	97.1%	darwin	✓
all.q@node004.cluster.private	BIP	100%	1.39453	79.7%	darwin	✓
all.q@node005.cluster.private	BIP	100%	1.36182	77.8%	darwin	✓
all.q@node006.cluster.private	BIP	100%	1.34814	77%	darwin	✓
all.q@node007.cluster.private	BIP	100%	1.46191	83.5%	darwin	✓
all.q@workgroupcluster.apple.com	BIP	100%	1.99072	113.8%	darwin	✓
disabled.q@node001.cluster.private	BIP	0%	1.35889	77.7%	darwin	✕ d

Active Jobs: 16

Priority	Job ID	Active Queue Instance	Job Owner	Job Name	Slots	Array Task	StartTime	State
0.55500	561	all.q@node001.cluster.private	www	Job7458	1	17	05:42:01 PM, Jan 08	r
0.55500	561	all.q@node001.cluster.private	www	Job7458	1	20	05:42:01 PM, Jan 08	r
0.55500	561	all.q@node002.cluster.private	www	Job7458	1	18	05:42:01 PM, Jan 08	r
0.55500	561	all.q@node002.cluster.private	www	Job7458	1	27	05:42:09 PM, Jan 08	r
0.55500	561	all.q@node003.cluster.private	www	Job7458	1	25	05:42:05 PM, Jan 08	r
0.55500	561	all.q@node003.cluster.private	www	Job7458	1	30	05:42:09 PM, Jan 08	r
0.55500	561	all.q@node004.cluster.private	www	Job7458	1	23	05:42:05 PM, Jan 08	r
0.55500	561	all.q@node004.cluster.private	www	Job7458	1	28	05:42:09 PM, Jan 08	r
0.55500	561	all.q@node005.cluster.private	www	Job7458	1	19	05:42:01 PM, Jan 08	r
0.55500	561	all.q@node005.cluster.private	www	Job7458	1	31	05:42:13 PM, Jan 08	r
0.55500	561	all.q@node006.cluster.private	www	Job7458	1	22	05:42:05 PM, Jan 08	r
0.55500	561	all.q@node006.cluster.private	www	Job7458	1	26	05:42:09 PM, Jan 08	r
0.55500	561	all.q@node007.cluster.private	www	Job7458	1	24	05:42:05 PM, Jan 08	r
0.55500	561	all.q@node007.cluster.private	www	Job7458	1	29	05:42:09 PM, Jan 08	r
0.55500	561	all.q@workgroupcluster.apple.com	www	Job7458	1	21	05:42:01 PM, Jan 08	r

New way …



Overview: Job 561 active on all.q@node001.cluster.private

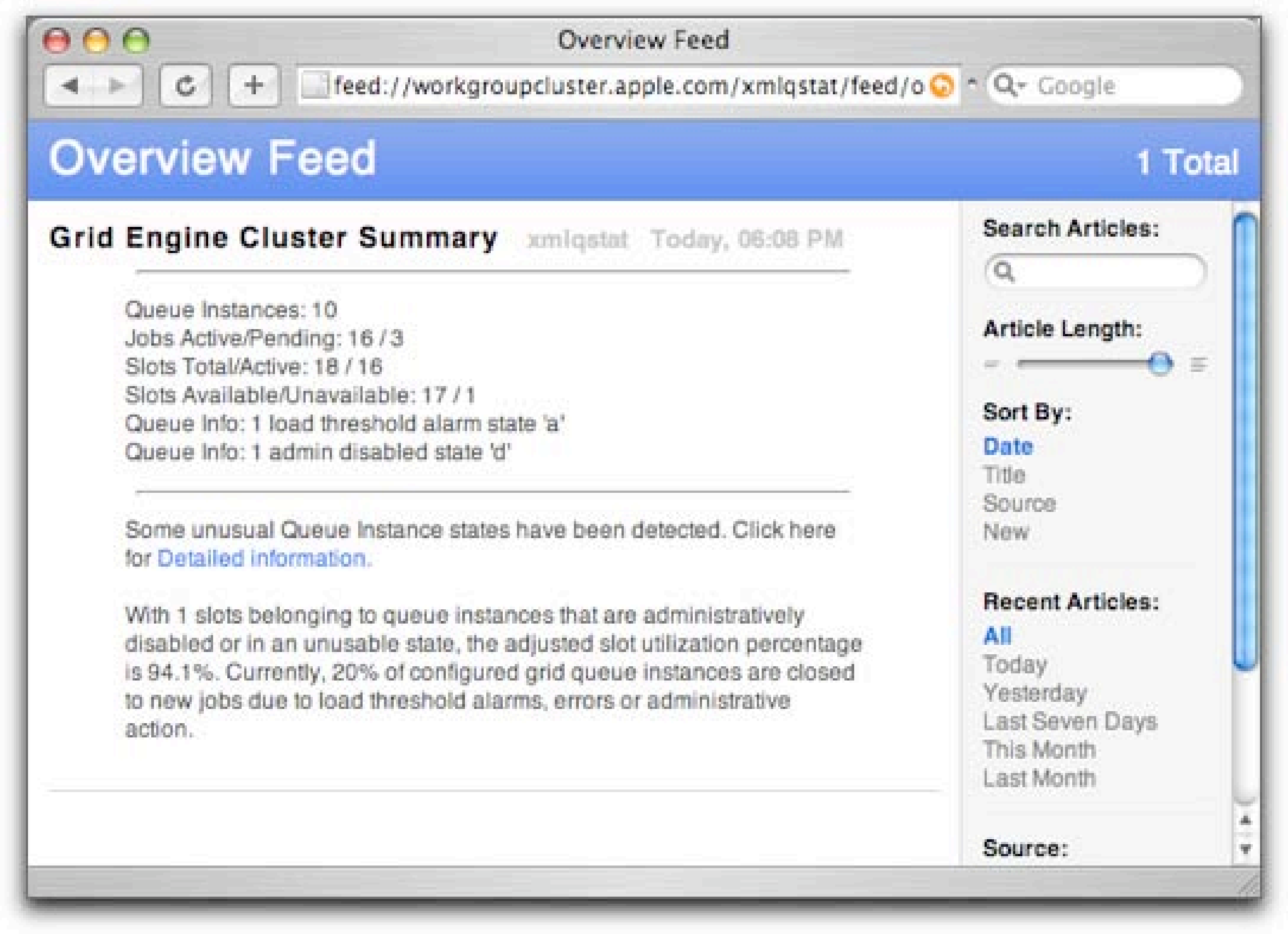
State	Owner	Name	Submitted	Script	Exec. File	uid	gid	group
r	www	Job7458	1136770694	/common/bin/blast_runner.pl	job_scripts/561	70	70	www

Queue Instance	Host	cwd	Project	Department
all.q@node001.cluster.private	node001.cluster.private	/Users/www/37k-testing/batchblast		defaultdepartment

Scheduling Messages: 11
queue instance "alarm.q@node001.cluster.private" dropped because it is overloaded: np_load_avg=1.297597 (= 0.912597 + 0.50 * 1.540000 with nproc=1) >= 0.0
queue instance "disabled.q@node001.cluster.private" dropped because it is disabled
queue instance "all.q@node001.cluster.private" dropped because it is full
queue instance "all.q@node005.cluster.private" dropped because it is full
queue instance "all.q@node006.cluster.private" dropped because it is full
queue instance "all.q@node002.cluster.private" dropped because it is full
queue instance "all.q@node003.cluster.private" dropped because it is full
queue instance "all.q@node007.cluster.private" dropped because it is full
queue instance "all.q@workgroupcluster.apple.com" dropped because it is full
queue instance "all.q@node004.cluster.private" dropped because it is full
All queues dropped because of overload or full

Page rendered: Sun, 8 Jan 2006 17:51:03

New way …



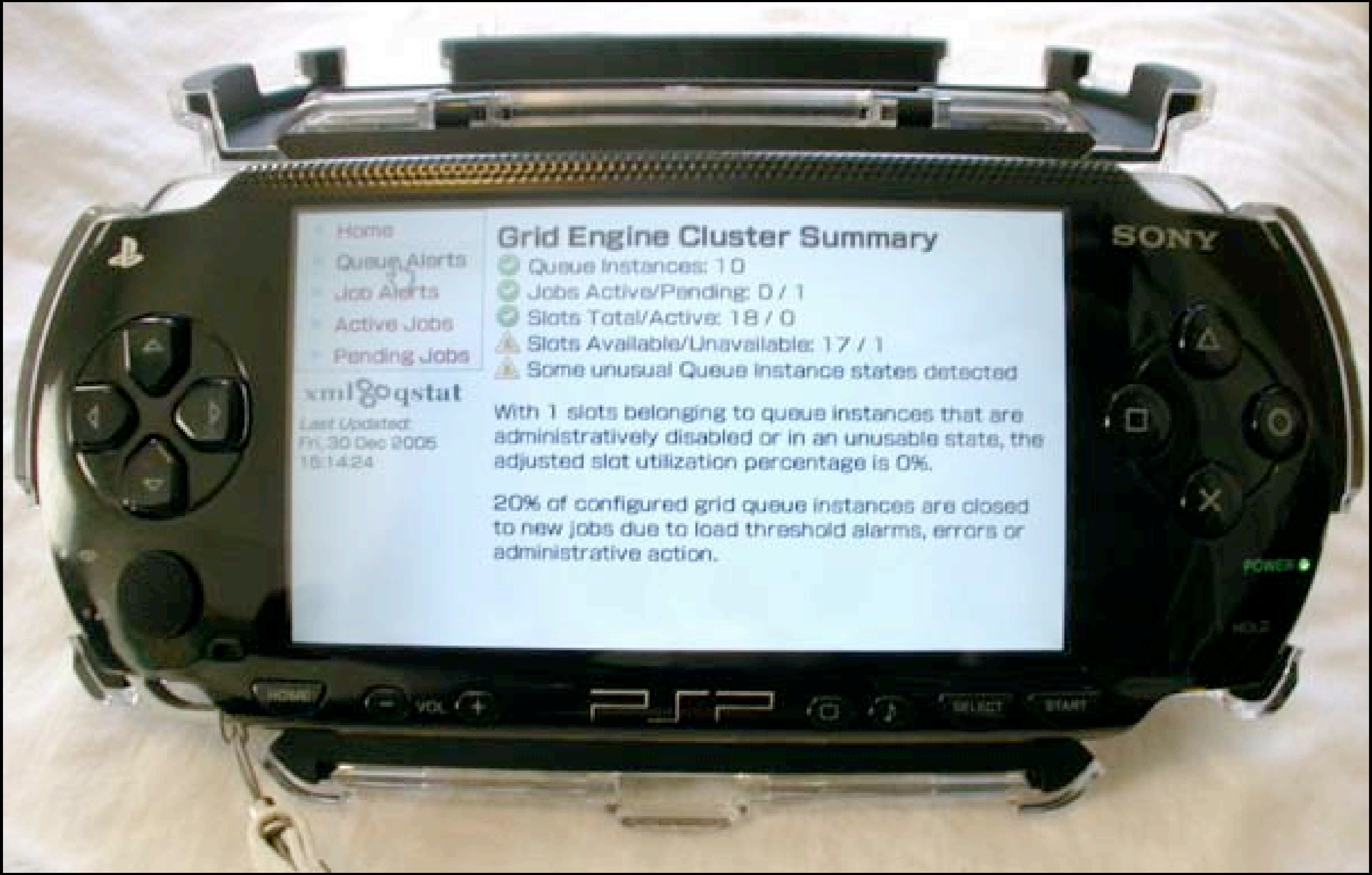
Grid Engine Cluster Summary xmlqstat Today, 06:08 PM

Queue Instances: 10
Jobs Active/Pending: 16 / 3
Slots Total/Active: 18 / 16
Slots Available/Unavailable: 17 / 1
Queue Info: 1 load threshold alarm state 'a'
Queue Info: 1 admin disabled state 'd'

Some unusual Queue Instance states have been detected. Click here for Detailed information.

With 1 slots belonging to queue instances that are administratively disabled or in an unusable state, the adjusted slot utilization percentage is 94.1%. Currently, 20% of configured grid queue instances are closed to new jobs due to load threshold alarms, errors or administrative action.

New way …



Grid Engine Cluster Summary

- ✓ Queue Instances: 10
- ✓ Jobs Active/Pending: 0 / 1
- ✓ Slots Total/Active: 18 / 0
- ⚠ Slots Available/Unavailable: 17 / 1
- ⚠ Some unusual Queue instance states detected

With 1 slots belonging to queue instances that are administratively disabled or in an unusable state, the adjusted slot utilization percentage is 0%.

20% of configured grid queue instances are closed to new jobs due to load threshold alarms, errors or administrative action.

Last Updated:
Fri, 30 Dec 2005
15:14:24

What I’ve learned …

XML is good

- Initial misconception:
  - “*Structured output solves the parsing problem only*”
- The real benefit of Grid Engine XML:
  - *Transform* XML into *anything* you need
    - Textfiles, Complex PDFs, Speadsheets, SMS alerts, manpages, web pages, RSS feeds, …
  - Can also search, parse, manipulate, etc.

Transforming XML

- Works like this
  1. Take any XML source
  2. Apply a XSL stylesheet
  3. Run both through a XSLT Engine
  4. Profit!

Buzzword Overkill

- XML
  - “*Extensible Markup Language*”
- XSLT
  - “*Extensible Stylesheet Language Transformation*”
- XPath
  - “*XML Path Language*”
- Web stuff
  - XHTML / DHTML
  - CSS
  - AJAX

XSLT & XPath

- Your source XML is transformed according to rules laid out in a XSLT stylesheet
- XPath “language”
  - Traverse XML document trees, parse data and run simple functions
- XPath statements are used within XSLT stylesheets

XSLT Example: Source

```
<?xml version="1.0" ?>
<persons>
  <person username="JS1">
    <name>John</name>
    <family_name>Smith</family_name>
  </person>
  <person username="MI1">
    <name>Morka</name>
   <family_name>Ismincius</family_name>
  </person>
</persons>
```

Source: Wikipedia XSLT article

XSLT Example: XSL

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">

<xsl:template match="/persons">
        <html xmlns="http://www.w3.org/1999/xhtml">
        <head> <title>Testing XML Example</title> </head>
        <body>
                <h1>Persons</h1>
                <ul>
                <xsl:apply-templates select="person">
                        <xsl:sort select="family_name" />
                </xsl:apply-templates>
                </ul>
        </body>
        </html>
</xsl:template>

<xsl:template match="person">
        <li>
                <xsl:value-of select="family_name"/>,
                <xsl:value-of select="name"/>
        </li>
</xsl:template>

</xsl:stylesheet>
```

XSLT Example: XHTML Result

```
<?xml version="1.0" encoding="UTF-8"?>
<html xmlns="http://www.w3.org/1999/xhtml">
<head> <title>Testing XML Example</title> </head>
<body>
        <h1>Persons</h1>
        <ul>
           <li>Ismincius, Morka</li>
           <li>Smith, John</li>
        </ul>
</body>
</html>
```

Source: Wikipedia XSLT article

Grid Engine XSLT Examples

- Count number of pending jobs:

```
<xsl:variable
  name=”pending_job_count”
  select="count(//job_info/job_list[@state='pending']/>
```

Grid Engine XSLT Examples

- Cluster wide slot utilization

```
<xsl:variable name="slotsUsed"
select="sum(//Queue-List/slots_used)"/>

<xsl:variable name="slotsTotal"
select="sum(//Queue-List/slots_total)"/>

  <xsl:variable name="slotsPercent"
select="($slotsUsed div $slotsTotal)*100" />
```

Grid Engine XSLT Examples

- Nested conditionals

```
<xsl:choose>
  <xsl:when test="//job_list[@state='pending']">
         <!-- HANDLE PENDING JOBS HERE -->
  </xsl:when>
  <xsl:otherwise>
         <!-- DO OTHER STUFF HERE -->
   </xsl:otherwise>
</xsl:choose>
```

Grid Engine XSLT Examples

- Better conditional example
  - Problem: queues in state “au” have no reported load_average. This was screwing up a HTML table
  - Embedded HTML shown in BLUE

```
<xsl:choose>
    <xsl:when test="load_avg">
     <!-- When load_avg is reported, output as usual -->
     <td class="boldcode">
         <xsl:value-of select="load_avg"/>
     </td>
    </xsl:when>
    <xsl:otherwise>
     <!-- otherwise, add a placeholder -->
     <td class="alarmcode">unknown</td>
    </xsl:otherwise>
</xsl:choose>
```

Rolling your own …

- Good online and paper references for XML, XSLT and XPath are all available
- There are *many* XSLT engine implementations …
  - Xalan (C++, Java)
  - XML:LibXSLT XML:LibXML (Perl)
  - Gnome libxml2 and libxslt (C)

Apache Cocoon

- http://cocoon.apache.org
    - xml-qstat (finally!) standardized on the Cocoon XML web publishing framework
    - Why Cocoon?
        - It just works
        - Most cross-platform solution I’ve found
            - Sole requirement: Java
        - Handles “web stuff” I don’t want to care about
            - Cookies, URL handling, parameter passing, etc.

Apache Cocoon

- All in one XML publishing framework
  - XML and XSLT transformations handled internally
  - Just point it at your XML and your XSL stylesheet and it does the rest
- Easily drops behind Apache or Tomcat
- Web apps guided by a sensible configuration file
  - “sitemap.xmap”

Cocoon & SGE XML: qstat.html

- From the sitemap.xmap file …

```
<!-- Standard View; CACHED Grid Engine XML data -->
<!-- URL will be "/xmlqstat/qstat.html"        -->

<map:match pattern="qstat.html">
  <map:generate src="xml/cached-sge-status.xml" />
      <map:transform src="xsl/qstat-xhtmlCSS-standard-v2.xsl">
          <map:parameter name="showQtable"  value="{cookie:displayQtable}" />
          <map:parameter name="enableResourceQueries" value="no" />
          <map:parameter name="real" value="yes" />
          <map:parameter name="perUserJobSort" value="no" />
       </map:transform>
   <map:serialize type="xhtml" />
</map:match>
```

Cocoon & SGE XML: RSS Feeds

- From the sitemap.xmap file …

```
<!-- ATOM XML FEED -->
<!-- URL pattern example:  /feed/overview   -->

<map:match pattern="feed/*">
   <map:generate src="xml/cached-sge-status.xml" />
     <map:transform src="xsl/feed-atom-{1}.xsl">
       <map:parameter name="ts1" value="{date:yyyy-MM-DD}"/>
       <map:parameter name="ts2” value="{date:HH:mm:ss}"/>
     </map:transform>
   <map:serialize type="atom-xml" />
</map:match>
```

Cocoon & SGE XML: Job view

- From the sitemap.xmap file

```
<!-- SGE JOB rendering for url path /job/<jobID>.html  -->

<map:match pattern="job/*.html">
<map:generate type=“CommandGenerator” src=”qstat -xml -j{1}" />
   <map:transform src="xsl/job-to-xhtml.xsl">
     <map:parameter name="jobID" value="{1}" />
   </map:transform>
 <map:serialize type="xhtml" />
</map:match>
```

XML-Qstat Current Status

Status: xml-qstat

- Original plan: 1.0 release with SGE 6.1
    - Then issue #2335 happened
    - Development effectively halted
- Since then:
    - Petr Jung contributed CommandGenerator
    - Mark Olesen comes to the rescue
        - Code cleanup
        - Discovered the “cocoon not caching” problem
        - Qlicserver integration
        - Qhost data integration

Status: xml-qstat

- Current state
  - Not packaged for general use yet
  - Is usable via SVN checkout from xml-qstat.org
    - Some tweaking required
- To do list:
  1. Solve cache issue or commit to daemon method
  2. Clean code; validate XHTML output
  3. Package & document for general release
  4. Develop iPhone interface*
- Timeframe:
  - Put out “xml-qstat-1.0” when SGE 6.2 leaves beta

Lessons learned

- Handling Grid Engine XML is pretty easy
- XSLT and XPath have a somewhat steep learning curve
- I was wrong about xml-qstat
  - The XML processing is the easy bit
  - 90% of the hard work involves pure, traditional web design
    - XHTML, CSS, DHTML, JavaScript, etc.
    - I’m a geek, not a web designer

Resources I found valuable

- "Learning XSLT" by Michael Fitzgerald, 2nd ed. (2004), O'Reilly
- "Javascript & DHTML Cookbook" by Danny Goodman, 1st ed (2003), O'Reilly
- "Javascript Bible" by Danny Goodman with Michael Morrison, 5th ed. (2003), Wiley
- "Web Designers Reference: An integrated approach to web design with XHTML and CSS" by Craig Grannel, 2005, friendsof
- "Eric Meyer on CSS: Mastering the language of web design" by Eric Meyer, 2003, New Riders Publishing
- "More Eric Meyer on CSS", by Eric Meyer, 2004, New Riders Publishing

How you can help:

- Java gurus
  - Need help making CommandGenerator use Cocoon’s built-in XML caching features
- XML/XSLT gurus
  - Once you stop laughing at my lame XSLT, help make it better
- Web & interface design gurus
  - DHTML, CSS, validation, AJAX assistance
- Large system owners
  - Help us create better “terse” views of SGE data

How you can help:

- Participate
  - Sun SGE devs are soliciting feedback on the future of XML output in Grid Engine
  - We really need feedback from people who want to use XML data for “real world” situations
  - Discussion on mailing lists
  - Wiki pages
    - http://wiki.gridengine.info/wiki/index.php/Qstat_XML
    - http://wiki.gridengine.info/wiki/index.php/GridEngine_XML

End;

- Thanks!

- Questions / Contact
  - Chris Dagdigian
    - Personal: dag@sonsorol.org
    - Corporate: chris@bioteam.net