Stupid Cluster Tricks

#1 - Grid Engine License Juggling

Chris Dagdigian
BioTeam Inc.

This is an applied talk…

- Making a Grid Engine cluster "license aware"
- Custom application integration examples

Chris Dagdigian
BioTeam Inc.

The project

- Novartis Institutes for Biomedical Research (NIBR)
- New linux cluster in Cambridge, MA
- NIBR Chose SGEEE over other commercial and internal alternatives

Chris Dagdigian
BioTeam Inc.

Primary user problem

- The most important cluster applications are FLEXlm licensed
- Significant percentage of jobs are failing due to license related issues
- Researchers must parse their own job output data to find missing results; then manually resubmit -- lots of time wasted

Goals

1. Make Grid Engine “license aware” when making resource allocation and scheduling decisions
2. Minimize license related job failures
3. Automatic detection and handling of the license errors that do occur

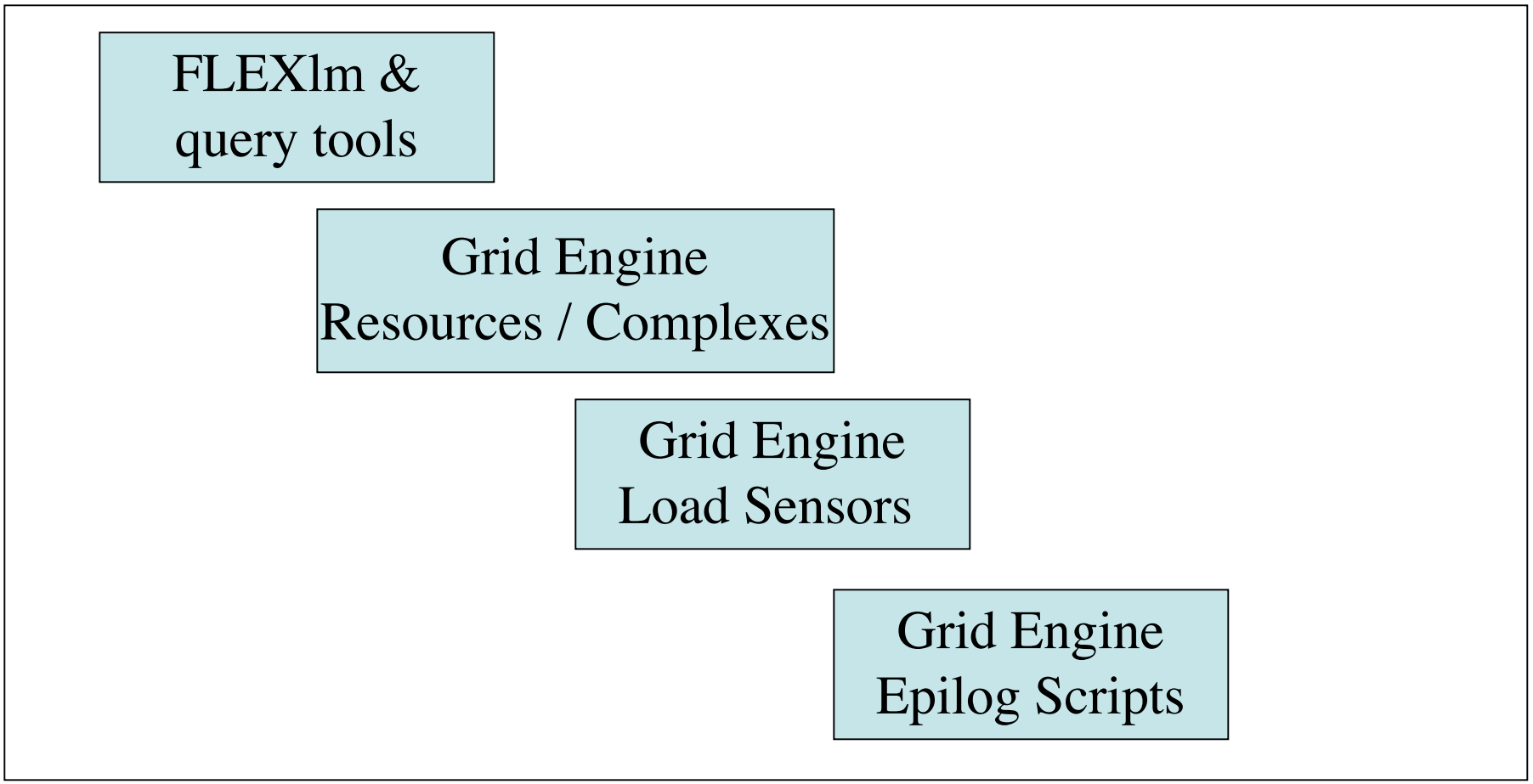
Chris Dagdigian
BioTeam Inc.

How? - Basic strategy

1. Poll FLEXlm server(s), parse required info and feed data periodically to Grid Engine
2. Configure Grid Engine to consider license usage data when scheduling jobs
3. Monitor job exit status and output files to discover and fix problems as they occur without user or operator intervention

Chris Dagdigian
BioTeam Inc.

Key Concepts Involved

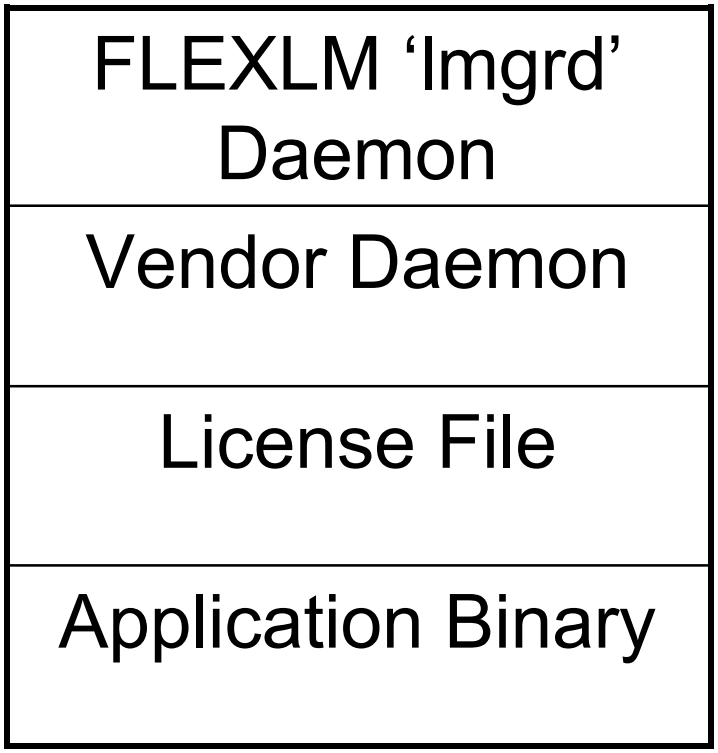


Chris Dagdigian
BioTeam Inc.

FLEXlm License Management

Chris Dagdigian
BioTeam Inc.

FLEXlm Overview



1. Application learns lmgrd hostname and port # from a license file
2. Application queries lmgrd daemon for info on how/where to contact the vendor daemon
3. Application queries vendor daemon for license grant/deny decision

Chris Dagdigian
BioTeam Inc.

Example FLEXlm License File

```
SERVER portal 000bdb9558a4 1700
DAEMON lsf_ld /usr/local/lsf/etc/lsf_ld
FEATURE lsf_base lsf_ld 6.000 08-Dec-2004 20 3D60958644C5F974E1CB
FEATURE lsf_manager lsf_ld 6.000 08-Dec-2004 20 8D9075F6EAEE22CAB7E7
FEATURE lsf_sched_fairshare lsf_ld 6.000 08-Dec-2004 20 BD80350661C0
FEATURE lsf_sched_preemption lsf_ld 6.000 08-Dec-2004 20 0D9085767F0
FEATURE lsf_sched_resource_reservation lsf_ld 6.000 08-Dec-2004 20
FEATURE lsf_sched_parallel lsf_ld 6.000 08-Dec-2004 20
FEATURE lsf_sched_advance_reservation lsf_ld 6.000 08-Dec-2004 20
FEATURE lsf_sla lsf_ld 6.000 08-Dec-2004 20 6D3025E6D613C87E049E
```


Chris Dagdigian
BioTeam Inc.

License File Features

```
SERVER portal 000bdb9558a4 1700
DAEMON lsf_ld /usr/local/lsf/etc/lsf_ld
FEATURE lsf_base lsf_ld 6.000 08-Dec-2004 20 3D60958644C5F974E1CB
FEATURE lsf_manager lsf_ld 6.000 08-Dec-2004 20 8D9075F6EAEE22CAB7E7
FEATURE lsf_sched_fairshare lsf_ld 6.000 08-Dec-2004 20 BD80350661C0
FEATURE lsf_sched_preemption lsf_ld 6.000 08-Dec-2004 20 0D9085767F0
FEATURE lsf_sched_resource_reservation lsf_ld 6.000 08-Dec-2004 20
FEATURE lsf_sched_parallel lsf_ld 6.000 08-Dec-2004 20
FEATURE lsf_sched_advance_reservation lsf_ld 6.000 08-Dec-2004 20
FEATURE lsf_sla lsf_ld 6.000 08-Dec-2004 20 6D3025E6D613C87E049E
```

License server hostname & TCP port number

Chris Dagdigian
BioTeam Inc.

License File Features

```
SERVER portal 000bdb9558a4 1700
DAEMON lsf_ld /usr/local/lsf/etc/lsf_ld
FEATURE lsf_base lsf_ld 6.000 08-Dec-2004 20 3D60958644C5F974E1CB
FEATURE lsf_manager lsf_ld 6.000 08-Dec-2004 20 8D9075F6EAEE22CAB7E7
FEATURE lsf_sched_fairshare lsf_ld 6.000 08-Dec-2004 20 BD80350661C0
FEATURE lsf_sched_preemption lsf_ld 6.000 08-Dec-2004 20 0D9085767F0
FEATURE lsf_sched_resource_reservation lsf_ld 6.000 08-Dec-2004 20
FEATURE lsf_sched_parallel lsf_ld 6.000 08-Dec-2004 20
FEATURE lsf_sched_advance_reservation lsf_ld 6.000 08-Dec-2004 20
FEATURE lsf_sla lsf_ld 6.000 08-Dec-2004 20 6D3025E6D613C87E049E
```

Vendor daemon name and full path to binary

Chris Dagdigian
BioTeam Inc.

License File Features

```
SERVER portal 000bdb9558a4 1700
DAEMON lsf_ld /usr/local/lsf/etc/lsf_ld
FEATURE lsf_base lsf_ld 6.000 08-Dec-2004 20 3D60958644C5F974E1CB
FEATURE lsf_manager lsf_ld 6.000 08-Dec-2004 20 8D9075F6EAEE22CAB7E7
FEATURE lsf_sched_fairshare lsf_ld 6.000 08-Dec-2004 20 BD80350661C0
FEATURE lsf_sched_preemption lsf_ld 6.000 08-Dec-2004 20 0D9085767F0
FEATURE lsf_sched_resource_reservation lsf_ld 6.000 08-Dec-2004 20
FEATURE lsf_sched_parallel lsf_ld 6.000 08-Dec-2004 20
FEATURE lsf_sched_advance_reservation lsf_ld 6.000 08-Dec-2004 20
FEATURE lsf_sla lsf_ld 6.000 08-Dec-2004 20 6D3025E6D613C87E049E
```

Vendor feature and total number of tokens

Chris Dagdigian
BioTeam Inc.

lmstat - Getting information

```
lmstat - Copyright (c) 1989-2003 by Macrovision Corporation.
All rights reserved.
Flexible License Manager status on Mon 3/29/2004 16:11

License server status: 1700@portal
    License file(s) on portal: /cluster/lsf/conf/license.dat:

    portal: license server UP (MASTER) v9.2

Vendor daemon status (on portal):

    lsf_ld: UP v9.2
```


Chris Dagdigian
BioTeam Inc.

lmstat - vendor daemon down

```
License server status: 1716@llmaster.na.novartis.net
    License file(s) on llmaster.na.novartis.net:
/cl/sw/XXX/license:

llmaster.na.novartis.net: license server UP (MASTER) v7.2

Vendor daemon status (on llmaster.na.novartis.net):

   XXX: The desired vendor daemon is down (-97,121)

Feature usage info:

Users of XXX_A: Cannot get users of XXX_A: No such feature exists (-5,222)
Users of XXX_B: Cannot get users of XXX_B: No such feature exists (-5,222)
Users of XXX_C: Cannot get users of XXX_C: No such feature exists (-5,222)
```


Chris Dagdigian
BioTeam Inc.

lmstat - license feature usage

```
lmstat - Copyright (c) 1989-2003 by Macrovision Corporation.
All rights reserved.
Flexible License Manager status on Mon 3/29/2004 16:16

Users of LSF_BASE:  (Total of 20 licenses issued;  Total of 16
licenses in use)

  "lsf_base" v6.000, vendor: lsf_ld
  floating license

    root portal /dev/tty (v6.0) (portal/1700 102), start Tue
3/23 13:08, 16 licenses
```


Chris Dagdigian
BioTeam Inc.

Grid Engine Complexes

Chris Dagdigian
BioTeam Inc.

Grid Engine Complexes

"a set of attributes that can be associated with a queue, a host or the entire cluster"

- Part of the 'resource' framework within Grid Engine
  - Scope can vary: Global, Host, Queue & User Defined
- Such attributes can be static
  - 'ARCH', 'HOSTNAME', 'NUM_PROC'
- They can also refer to dynamic values
  - 'MEMFREE', 'SWAP', 'LOADAVG'. 'UTILIZATION'

Chris Dagdigian
BioTeam Inc.

Grid Engine Complexes

"a set of attributes that can be associated with a queue, a host or the entire cluster"

- Grid Engine understands that some elements of a complex can be configured to be:
  - User requestable
  - Consumable
- As a general rule, almost everything is requestable while very few attributes are 'consumable'

Chris Dagdigian
BioTeam Inc.

Requesting Resources

Resources can be collected together using arithmetic and boolean operators to form resource requirement strings:

```
$ qsub -l arch=solaris64,h_mem_free=800M,swap_free=50M
./MyClusterJob.sh
```

Creating and managing our own resource(s) would allow us to do cool things with Grid Engine like this:

```
$ qsub -hard -l "License_A=1" ./MyClusterJob.sh
```


Chris Dagdigian
BioTeam Inc.

Configuring the complex

1. For each license feature you wish to monitor:
   (Some applications may checkout several license features)
2. Create a new entry within the GLOBAL complex
3. The attribute should have the following primary settings:

```
Requestable (req=Yes)
Consumable (cons=Yes)
Initial value = 0
Type = Integer
Relop = ‘<=‘
```


Chris Dagdigian
BioTeam Inc.

```
Name        shortnam TYP   VAL RELOP    REQ   CONS   INITVAL
-------------------------------------------------------------
License_A   appa     INT     0    <=     YES    YES      0
License_B   appb     INT     0    <=     YES    YES      0
License_C   appc     INT     0    <=     YES    YES      0
-------------------------------------------------------------
```


Chris Dagdigian
BioTeam Inc.

Show current values:

```
bash-2.04$ qconf -se global

 hostname                    global
 load_scaling                NONE
 complex_list                NONE
 complex_values              License_A=0,License_B=0,License_C=0
 load_values                 License_A=0,License_B=0,License_C=0
 processors                  0
 user_lists                  NONE
 xuser_lists                 NONE
 projects                    NONE
 xprojects                   NONE
 usage_scaling               NONE
 resource_capability_factor 0.000000
```


Chris Dagdigian
BioTeam Inc.

Grid Engine Load Sensors

Chris Dagdigian
BioTeam Inc.

Grid Engine Load Sensors

A load sensor is an external script run periodically by Grid Engine. The output of the sensor script is used to update or change the value of one or more Grid Engine Complexes

- A load sensor script can query the FLEXlm server for license data and feed the results into Grid Engine
- This is generally done by running the FLEXlm utility 'lmstat' and parsing the resulting output

Chris Dagdigian
BioTeam Inc.

Load Sensor Output Format:

```
begin
global:License_A:50
global:License_B:160
global:License_C:56
global:License_d:24
end
```


Chris Dagdigian
BioTeam Inc.

Parsing ‘old’ lmstat output

```
License server status: 1716@llmaster.na.novartis.net
llmaster.na.novartis.net: license server UP (MASTER) v7.2

Vendor daemon status (on llmaster.na.novartis.net):
    A_VENDOR: UP v7.2

Feature usage info:

Users of APPLICATION_A:  (Total of 1000 licenses available)

  "APPLICATION_A" v29, vendor: A_VENDOR
  nodelocked license locked to NOTHING (hostid=ANY)

 bioteam node01 (llmaster/1716 109), start Wed 7/9 16:42
 bioteam node39 (llmaster/1716 209), start Wed 7/9 16:43
 bioteam node23 (llmaster/1716 309), start Wed 7/9 16:44
```


Chris Dagdigian
BioTeam Inc.

Parsing ‘old’ lmstat output

1. Construct the ‘lmstat’ command such that the license server is only asked about the single feature you care about

2. Count the number of times the word ‘start’ appears in the resulting output

Chris Dagdigian
BioTeam Inc.

Newer versions are easier

- A single ‘lmstat’ command can provide data on many licensed applications as there is now a unique parsable pattern:

```
Users of LICENSE_A:  (Total of 1000 licenses issued; \
Total of 23 licenses in use)

Users of LICENSE_B:  (Total of 45 licenses issued; \
Total of 0 licenses in use
```


Chris Dagdigian
BioTeam Inc.

```
#!/bin/sh
## load-sensor.sh -- example -- just a loop !
while [ 1 ]; do

        # wait for input
        read input
        result=$?
        if [ $result != 0 ]; then
                exit 1
        fi
        # if you just hit enter, continue
        if [ -z $input ]; then
                input=go
        fi
        if [ $input = "quit" ]; then
                exit 0
        fi
        echo "begin"
        ## These perl scripts actually do the lmstat query and output parsing
        perl /cl/sw/sge/bin/application-a-helper.pl
        perl /cl/sw/sge/bin/application-b-helper.pl
        echo "end"
done
exit 0
```


Chris Dagdigian
BioTeam Inc.

What the helper scripts do

- Run the ‘lmstat’ command and parse the resulting output
- If the vendor daemon is down or FLEXlm itself is unreachable then report “0” licenses available
- If all is well, print formatted data to STDOUT

Chris Dagdigian
BioTeam Inc.

Putting it all together

Chris Dagdigian
BioTeam Inc.

Create complex(es)

```
bash-2.04$ qconf -se global

 hostname                    global
 load_scaling                NONE
 complex_list                NONE
 complex_values              License_A=0,License_B=0,License_C=0
 load_values                 License_A=4,License_B=9,License_C=0
 processors                  0
 user_lists                  NONE
 xuser_lists                 NONE
 projects                    NONE
 xprojects                   NONE
 usage_scaling               NONE
 resource_capability_factor 0.000000
```


Chris Dagdigian
BioTeam Inc.

Load Sensors query FLEXlm

```
begin
global:License_A:50
global:License_B:160
global:License_C:56
global:License_d:24
end
```


Chris Dagdigian
BioTeam Inc.

Launch licensed jobs

Users make a non-negotiable ‘hard’ resource request for the specific complex that tracks the license(s) they need:

```
$ qsub -hard -l “License_A=1,License_B=4”
./MyClusterJob.sh
```

Grid Engine will not schedule a job for execution until all ‘-hard’ resource requests can be met

These resource requests can be embedded into job scripts so that users never actually have to deal with them

Chris Dagdigian
BioTeam Inc.

Error Handling

- Loosely coupling FLEXlm license servers to the Grid Engine scheduler via load sensor scripts is prone to problems and race conditions
- What we really need is an FLEXlm API for advanced license reservation and checkout that cluster schedulers can invoke directly

Chris Dagdigian
BioTeam Inc.

Most common error

- SGE internal license counts get out of sync briefly with the real values held by the license server
- Result: Grid Engine launches a job that dies because it is denied a license token by FLEXlm

Chris Dagdigian
BioTeam Inc.

How this happens

- Many organizations run centralized FLEXlm servers
  - Cluster is not the only client
  - Load sensors run at periodic intervals
  - Other license users are active within the organization or cluster users run jobs outside of SGE control
  - SGE can get out of sync just long enough to mess up a few cluster jobs

Chris Dagdigian
BioTeam Inc.

Nasty problem we saw

- Grid Engine launches a licensed app
  - Subtracts 1 from internal license count
- Running application does not check out its license token right away
- While the application crunches data…
  - Load sensor runs and reports back fresh data, internal license count goes up by +1 since the running app never contacted FLEXlm daemon
- Grid engine launches an additional application; one or both apps then fail

Chris Dagdigian
BioTeam Inc.

Dealing with license errors

- If at all possible, have your application exit with status code ‘99’ if license problems are encountered -- Grid Engine will automatically reschedule jobs that exit ‘code 99’

- Other techniques require epilog scripts…

SGE Epilog scripts

Epilog scripts can be configured to run whenever a cluster job exits (normally or abnormally).

Environment variables contain lots of useful info about the actual job and where/how it ran.

Chris Dagdigian
BioTeam Inc.

Simple Epilog Script

```
#!/bin/sh

# Simple epilog script

JOB_EXIT_STATUS="`sed -ne 's/^exit_status=//p' \
   $SGE_JOB_SPOOL_DIR/usage | tail -1`"

echo "--------"
echo "Job exited code: $JOB_EXIT_STATUS"
echo "Job output should be in directory $SGE_WORKDIR"
echo "--------"
```


Chris Dagdigian
BioTeam Inc.

Epilog: finding license errors

```
STARVEDETECT="`grep -c "Licensed number of users \
   already reached" $SGE_O_WORKDIR/*.log `"

  if [ $STARVEDETECT -gt 0 ]
   then
     echo “License Error Pattern Detected in Output!”
      /bin/tcsh -c “cd $SGE_O_WORKDIR; \
     /cl/sw/bin/restart-failed-job.pl “
   else
     echo “No problems detected”
fi
```


Chris Dagdigian
BioTeam Inc.

One epilog; many cluster apps

- Error handling and ‘license problem finding’ techniques are going to be application and workflow specific.
- One epilog script can contain logic to diagnose/debug/troubleshoot many different applications -- the epilog code just needs to figure out which type of job just exited

Chris Dagdigian
BioTeam Inc.

One epilog; many cluster apps

- We enforced the use of SGE “project” strings at the user command-line.
- The project info passed to ‘qsub’ via the command-line ends up in an an environment variable named $SGE_PROJECT
- This is enough info for our epilog code to know what type of job just completed
- Other approaches would work as well…

Chris Dagdigian
BioTeam Inc.

NIBR Cluster End Result

- Load sensor scripts query multiple local and remote FLEXlm license servers for license data
- The global Complex holds values for all the license tokens/features we care about
- Application specific logic within a single epilog script does error catching and automatic job resubmission if license errors are detected
- Works great!

Chris Dagdigian
BioTeam Inc.

Lessons Learned

- Upgrade FLEXlm if possible to make license status parsing easier and less error prone
- Use <port>\@<hostname> instead of path to license file when using lmstat -- avoid license path dependency
- Race conditions are unavoidable given the loosely coupled system we deployed
- Epilog scripts are a great way to catch errors if they do occur
- For license aware cluster applications:
  - Exit code ‘99’ is great - no epilog necessary!
  - Unique exit codes can be easily handled if the vendor is smart enough to throw one for license errors
  - If all else fails parse the output in $SGE_O_WORKDIR/ to try to figure what happened to your applications

Chris Dagdigian
BioTeam Inc.

end;

chris@bioteam.net

Special thanks to NIBR Staff:

- Dmitri Mikhailov
- Chris Harwell
- Goran Pocina
- Wolfgang Zipfel
- Matthew Gabelerlee