ADEM- Application Software Automatic Deployment and Management on OSG

(I) For the Application Software End-users on OSG

The end users can directly use a few commands of the ADEM tool to deploy and manage application software. Sites.xml and tc.data files for the Swift execution can also be generated.

1. Download ADEM tool

wget http://www.ci.uchicago.edu/~houzx/adem-osg.tar.gz
tar –zxvf adem-osg.tar.gz
OR
svn co https://svn.ci.uchicago.edu/svn/vdl2/SwiftApps/adem-osg adem-osg

2. Set the environment

cd adem-osg
source setup.sh (for Bash)
source setup.csh (for Csh)

3. Usage

(a) Get the grid sites automatically and dynamically of a given VO within the grids

```
[houzx@login bin]$ auto-get-sites
(1)Please input the GRID name as the first parameter, which can be:
osg osg-itb
(2)Please input the virtual organization name as the second parameter.Just use a space to separate the two parameters. For osg, vo can be:
CDF CMS CompBioGrid DES DOSAR DZero Engage Fermilab fMRI GADU geant4 GLOW GPN GRASE GridChem GridEx GROW i2u2 iVDGL LIGO mariachi MIS nanoHUB NWICG Ops OSG OSGEDU SDSS STAR USATLAS
E. G.
[houzx@login bin]$ auto-get-sites osg osg
It's waiting for the authentication tests results. A few seconds please!
Now it's $DATE. Altogether, there are # available grid sites in the cache (started from OSG VO), and # new possible grid sites.
The available sites file for the application deployment is: /home/houzx/adem-osg/logs/sites/osg-osg-avail-sites-$DATE.txt
The property information of new possible grid sites are being checked and will be added into the CACHE file after confirmation.
```

(b) Automatically deploy a given application to the available grid sites of a given VO

```
[houzx@login bin]$ auto-deploy-app
It's checking the available applications in the repository...
(1)Please input the Application software name to be deployed as the first parameter. If it is a new application software, Please submit it to the application administrator. Or the already packaged application software includes:
afni-2.0
angle-1.0
blast-2.2.18
dock-6.2
Flex-2.5.4a
freesurfer-4.0.4
lib-2.3.4
mpiblast-1.5.0
mpich-1.2.7
nab-5.1.2
```

octave-3.0.1

oops-0.1

R-2.5.1

Yacc-0.5.9

(2)Please input the Virtual Organization name as the second parameter.Just use a space to separate the parameters.For osg, vo can be:

CDF CMS CompBioGrid DES DOSAR DZero Engage Fermilab fMRI GADU geant4 GLOW GPN GRASE GridChem GridEx GROW i2u2 iVDGL LIGO mariachi MIS nanoHUB NWICG Ops OSG OSGEDU SDSS STAR USATLAS. For Teragrid, vo just use: teragrid.

(3)And please input the grid sites file as the third parameter. Just use a space to separate the parameters. You can use auto-get-sites to create the available grid sites automatically.

E. G.

[houzx@login bin]$ auto-deploy-app dock-6.2 osg ../logs/sites/osg-osg-avail-sites-$DATE.txt

(c) Automatically check the deployment results of all kinds of applications on the grid sites of a given VO

[houzx@login bin]$ auto-check-app

(1)Please input the Virtual Organization name which you belong to,as the first parameter.For osg, vo can be:

CDF CMS CompBioGrid DES DOSAR DZero Engage Fermilab fMRI GADU geant4 GLOW GPN GRASE GridChem GridEx GROW i2u2 iVDGL LIGO mariachi MIS nanoHUB NWICG Ops OSG OSGEDU SDSS STAR USATLAS.

(2)Please input the grid sites file as the second parameter. Just use a space to separate the two parameters. You can use auto-get-sites to create the available grid sites file automatically.

(3)Please input the application name as the third parameter. So, you can check just one application. Otherwise, it will check all of the applications. This is optional.

E. G.

[houzx@login bin]$ auto-check-app osg ../logs/sites/osg-osg-avail-sites-$DATE.txt

(d) Automatically remove a given application from the selected grid sites

[houzx@login bin]$ auto-rm-app

It's checking the available application names in the repository...

(1)Please input the Application software name to be removed as the first parameter. 'ALL' Or the already packaged application software are:

afni-2.0

angle-1.0

blast-2.2.18

dock-6.2

Flex-2.5.4a

freesurfer-4.0.4

lib-2.3.4

mpiblast-1.5.0

mpich-1.2.7

nab-5.1.2

octave-3.0.1

oops-0.1

R-2.5.1

Yacc-0.5.9

(2)Please input the Virtual Organization you belong to as the second parameter. For osg, vo can be:

CDF CMS CompBioGrid DES DOSAR DZero Engage Fermilab fMRI GADU geant4 GLOW GPN GRASE GridChem GridEx GROW i2u2 iVDGL LIGO mariachi MIS nanoHUB NWICG Ops OSG OSGEDU SDSS STAR USATLAS. For Teragrid, vo just use: teragrid.

(3)And please input the grid sites file as the third parameter. Just use a space to separate the parameters. You can use auto-get-sites to create the available grid sites automatically.

E. G.

[houzx@login bin]$ auto-rm-app dock-6.2 osg ../logs/sites/osg-osg-avail-sites-$DATE.txt

(e) Automatically update a given application from the selected grid sites

[houzx@login bin]$ auto-update-app

It's checking the available application names in the repository...

(1)Please input the Application software name to be updated as the first parameter. The already packaged application software are:

afni-2.0

angle-1.0

blast-2.2.18

dock-6.2

Flex-2.5.4a

freesurfer-4.0.4

lib-2.3.4

mpiblast-1.5.0

mpich-1.2.7

nab-5.1.2

octave-3.0.1

oops-0.1

R-2.5.1

Yacc-0.5.9

(2)Please input the Virtual Organization you belong to as the second parameter. For osg, vo can be:

CDF CMS CompBioGrid DES DOSAR DZero Engage Fermilab fMRI GADU geant4 GLOW GPN GRASE GridChem GridEx GROW i2u2 iVDGL LIGO mariachi MIS nanoHUB NWICG Ops OSG OSGEDU SDSS STAR USATLAS. For Teragrid, vo just use: teragrid.

(3)And please input the grid sites file as the third parameter. Just use a space to separate the two parameters. You can use auto-get-sites to create the available grid sites automatically.

E. G.

[houzx@login bin]$ auto-update-app dock-6.2 osg ../logs/sites/osg-osg-avail-sites-$DATE.txt

(f) Get the sites.xml file for swift execution

[houzx@login bin]$ swift-auto-sites-xml

(1)Please input the GRID name as the first parameter, which can be:

osg osg-itb

(2)Please input the virtual organization name as the second parameter.Just use a space to separate the two parameters. For osg, vo can be:

CDF CMS CompBioGrid DES DOSAR DZero Engage Fermilab fMRI GADU geant4 GLOW GPN GRASE GridChem GridEx GROW i2u2 iVDGL LIGO mariachi MIS nanoHUB NWICG Ops OSG OSGEDU SDSS STAR USATLAS

E. G.
[houzx@login bin]$ swift-auto-sites-xml osg osg
The general sites file for swift execution is: /home/houzx/adem-osg/swift-execution-example/swift-sites.xml

(g) Get the tc.data file for swift execution

[houzx@login bin]$ swift-auto-tc-data

(1)Please input the Virtual Organization name which you belong to,as the first parameter.For osg, vo can be:

CDF CMS CompBioGrid DES DOSAR DZero Engage Fermilab fMRI GADU geant4 GLOW GPN GRASE GridChem GridEx GROW i2u2 iVDGL LIGO mariachi MIS nanoHUB NWICG Ops OSG OSGEDU SDSS STAR USATLAS.

(2)Please input the grid sites file as the second parameter. Just use a space to separate the two parameters. You can use auto-get-sites to create the available grid sites file automatically.

E. G.

[houzx@login bin]$ swift-auto-tc-data osg ../logs/sites/osg-osg-avail-sites-$DATE.txt

(II) For the Application Software Repository Administrator

The application administrator is in charge of maintaining the application software repository, and pacman Cache, including writing and updating pacman files, uploading and managing application software source code or pre-built application binary tarballs.

1. Repository

The current repository is:

ADEM	http://www.ci.uchicago.edu/~houzx/pac-cache/	Zhengxiong Hou / Greg Cross	zhengxiongh@uchicago.edu

The potential repository can be:

YOUR OWN REPOSITORY			
VTB	http://osg-vtb.uchicago.edu/vtb/	Suchandra Thapa	sthapa@ci.uchicago.edu
OSG	http://software.grid.iu.edu/pacman/	Leigh Grundhoefer	leighg@indiana.edu
TeraGrid	http://software.teragrid.org/pacman/	JP Navarro	navarro@mcs.anl.gov

2. Maintenance

(a) Upload application software source code tarball into the repository.

(b-optional) Build the application software by NMI B&T system (http://nmi.cs.wisc.edu/), then get the pre-built application software binary tarball.

E.g. R application

```
[nmi-s005] /home/houzx/R-ftp > ls
r-build.sh  R.ftp  R.scp  R.submit
[nmi-s005] /home/houzx/R-ftp > cat R.submit
project = adem
component = R
component_version = 2.5.1
description = R B&T using scp and ftp
run_type = build

inputs = R.scp, R.ftp
remote_task = r-build.sh
platforms = x86_64_rhas_4, x86_64_rhas_3
notify = houzhx@hotmail.com
[nmi-s005] /home/houzx/R-ftp > cat R.scp
method=scp
scp_file=/home/houzx/R-ftp/r-build.sh

[nmi-s005] /home/houzx/R-ftp > cat R.ftp
method = ftp
ftp_root = http://www.ci.uchicago.edu/~houzx/pac-cache/
ftp_target = R-2.5.1.tar.gz
[nmi-s005] /home/houzx/R-ftp > cat r-build.sh
tar -zxvf R-2.5.1.tar.gz
cd R-2.5.1
./configure --prefix=`pwd`
make && make install
cd ../
tar -czv results.tar.gz R-2.5.1
```

(c) Write pacman files for different site signatures and put them into the repository

The pacman file mainly includes the description of the application software, what to download, how to install, dependencies, and how to test.

E.g1. Pacman file for OOPS source code tarball (oops-linux-x86_64.pacman)

```
#Description of the package
description = 'Oops-folding-i686'
# What to download?
platformGE('unix')
downloadUntarzip('http://www.ci.uchicago.edu/~houzx/pac-cache/OOPS.v01.tar.gz','Oops_TAR')
# How to install
cd('$Oops_TAR/src')
shell('./install > install.log')
#How to test
cd('$Oops_TAR/tests')
shell('./doopstest.sh > test-oops.log')
cd()
```

E.g2. Pacman file for dock binary tarball (dock-linux-i686.pacman)

```
#Description of the package
description = 'DOCK Binary for Molecular Dynamics on (i686)x86_rhas_3 platform'
# What to download?
platformGE('unix')
downloadUntarzip('http://www.ci.uchicago.edu/~houzx/pac-cache/dock6-i686_cent_4.2.tar.gz','DOCK_TAR')
#How to install
# just unpack and it's OK.
#How to test
# Use real applications
cd()
```

E.g.3 Pacman file for R source code tarball(R-linux-i686.pacman)

```
#Description of the package
description = 'R-2.5.1'
#check /usr/bin/g77 or g77 in workpac
{ exists('/usr/bin/g77') OR package('http://www.ci.uchicago.edu/~houzx/pac-cache:Gcc') }
# What to download?
platformGE('unix')
downloadUntarzip('http://www.ci.uchicago.edu/~houzx/pac-cache/R-2.5.1.tar.gz','R_TAR')
# How to install
cd('$R_TAR')
shell('./configure --with-x=no --with-readline=no --prefix=$R_TAR > config.log')
shell('make > R-make.log')
shell('make install > R-make-install.log')
# How to test
# use angle to test it
cd()
```

E.g.4 Pacman file for R binary tarball (R-linux-i686.pacman)

```
#Description of the package
description = 'R-2.5.1 for i686_cent_4.2'
# What to download?
platformGE('unix')
downloadUntarzip('http://www.ci.uchicago.edu/~houzx/pac-cache/R-i686_cent_4.2.tar.gz','R_TAR')
# How to install
# just unpack and it's OK.
# How to test
# use angle to test it
cd()
```

(d) Update pacman files

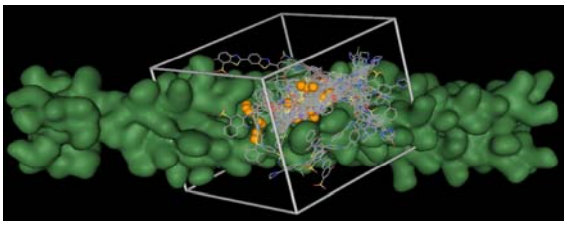
Just modify and update the pacman file.

E.g. To update the OOPS.v01 to OOPS.v03 in the above oops-linux-x86_64.pacman, it just needs to modify this line:

```
downloadUntarzip('http://www.ci.uchicago.edu/~houzx/pac-cache/OOPS.v03.tar.gz','Oops_TAR')
```

(III) Application Execution Examples by Swift

3.1 Molecular Dynamics: DOCK



This image graphically depicts a typical DOCK setup used toward the discovery of a potential disease inhibitor. The target (green) is an HIVgp41 protein, which mediates membrane fusion between the HIV virus and the host cell. During DOCKing, spheres (orange) are used to guide the placement of potential compounds (colored by atom type). These compounds are then scored using a grid (gray box) and ranked by their potential to bind to the target. (Courtesy of Dr. Robert Rizzo)

1. DOCK6 application running description

This application, executed on the OSG grid sites, screens KEGG compounds and drugs against important metabolic protein targets using the DOCK6 application to simulate the "docking" of small molecules, or ligands, to the "active sites" of large macromolecules of known structure called "receptors". A compound that interacts strongly with a receptor (such as a protein molecule) associated with a disease may inhibit its function and thus act as a beneficial drug. The economic and health benefits of speeding drug development by rapidly screening for promising compounds and eliminating costly dead-ends is significant in terms of both resources and human life. In this application run, nine proteins that perform key enzymatic functions in the core metabolism of bacteria and humans were selected for screening against a database of 15,351 natural compounds and existing drugs in KEGG's ligand database.

[houzx@login databases]$ ls Targets/NaturalLigands
1F9Y 1G97 1JBW 1KQP 1KZL 1OD6 1SNN 1SUW 1VHT
[houzx@login databases]$ ls Targets/NaturalLigands/1F9Y/
grid.bmp grid.nrg selected_spheres.sph

[houzx@login databases]$ ls KEGG_and_Drugs/
all 15352 possibilities (15351 natural compounds and existing drugs, 1 debug file)
[houzx@login dock-run]$ cat databases/KEGG_and_Drugs/C00001.mol2
@<TRIPOS>MOLECULE
C00001
3 2 0 0 0
SMALL
USER_CHARGES

@<TRIPOS>ATOM
1 O1 -0.6068 0.2382 0.7169 O.3 1 <0> -0.8600
2 H1 -0.3293 1.1312 0.9713 H 1 <0> 0.4300
3 H2 0.0000 -0.0000 0.0000 H 1 <0> 0.4300
@<TRIPOS>BOND
1 1 2 1
2 1 3 1

2. Regenerate the parameters

Step 1: Run /dock6/bin/grid -i grid.in -o grid.out

Required files:
grid.in - points to location of 2 files
dock6/parameters/vdw_AMBER_parm99.defn
receptor_charged.mol2 - created manually with Chimera
rec_box.pdb - created manually with DOCK6 utilities

Step 2: Run /dock6/bin/dock -i dock.in -o dock.out

required files:
dock.in - edit to point to compound database,
selected_spheres.sph, dir/prefix of grid files, location of /dock6/parameters
selected_spheres.sph - created manually with DOCK6 utilities
grid.bmp - output file from Step 1
grid.nrg - output file from Step 1

3. Steps for running the Dock6 application on the OSG grid sites by Swift

(1) Download and setup adem-osg tool
(It just needs one time. Please refer to Part I)

(2) Get the available grid sites
auto-get-sites $GRID $VO (get the available grid sites within a given virtual organization in osg or osg-itb)
e.g. auto-get-sites osg osg

(3) Get the sites.xml for swift
swift-auto-sites-xml *$GRID $VO*
e.g. swift-auto-sites-xml osg osg
(swift-sites.xml will be generated in $ADEM_HOME/swift-execution-example for different users, e.g. sites-10.xml)

(4) prepare-for-dock-swift-submit
Include: Transmit the Target files and Parameters (dockKAGxNL.tar.gz) to the grid sites; Modify and transfer the execution wrapper scripts (rundock) to the grid sites; prepare tc.data file (E.g. tc-dock.data)
cd $ADEM_HOME/swift-execution-example/dock
./prepare-for-dock-swift-submit $VO $Grid-sites-file
(e.g. ./prepare-for-dock-swift-submit osg $ADEM_HOME/logs/sites/osg-osg-avail-sites-$DATE.txt)

(5) Edit the swift code and pick the ligands input files

```
[houzx@login results]$ cat grid-many-dock6-auto.swift
type file;
type DockIn;
type DockOut;

(file t,DockOut tarout) dockcompute (DockIn infile, string targetlist) {
  app {
    rundock @infile targetlist stdout=@filename(t) @tarout;
    }
}

type params {
     string  ligandsfile;
```

```
    string  targetlist;
}

#params pset[] <csv_mapper;file="paramslist.txt">;
doall(params pset[])
{
  foreach params,i in pset {
  DockIn infile < single_file_mapper; file=@strcat("/home/houzx/dock-run/databases/KEGG_and_Drugs/",pset[i].ligandsfile)>;
  file sout <single_file_mapper; file=@strcat("/home/houzx/dock-run/databases/results/stdout/",pset[i].targetlist,"-",i,"-stdout.txt")>;
  DockOut tout <single_file_mapper; file=@strcat(pset[i].ligandsfile,"-result.tar.gz")>;
  (sout,tout) =  dockcompute(infile,pset[i].targetlist);
 }
}

params p[];
p = readdata("paramslist.txt");
doall(p);

[houzx@login results]$ head paramslist.txt
ligandsfile targetlist
C10000.mol2 1F9Y
C10001.mol2 1F9Y
C10002.mol2 1F9Y
C10003.mol2 1F9Y
C10004.mol2 1F9Y
C10005.mol2 1F9Y
C10006.mol2 1F9Y
C10007.mol2 1F9Y
C10008.mol2 1F9Y
```

*Note: The red places need to be modified by the individual users.

(6) Submit the job

```
[houzx@login results]$ swift -sites.file ../sites-10.xml -tc.file ./dock-tc.data grid-many-dock6-auto.swift
```

4. Experiment Results

- How to plot for the results log file

On a client computer, type:

```
svn co https://svn.ci.uchicago.edu/svn/vdl2/log-processing
cd log-processing/bin
export PATH=$(pwd):$PATH
```

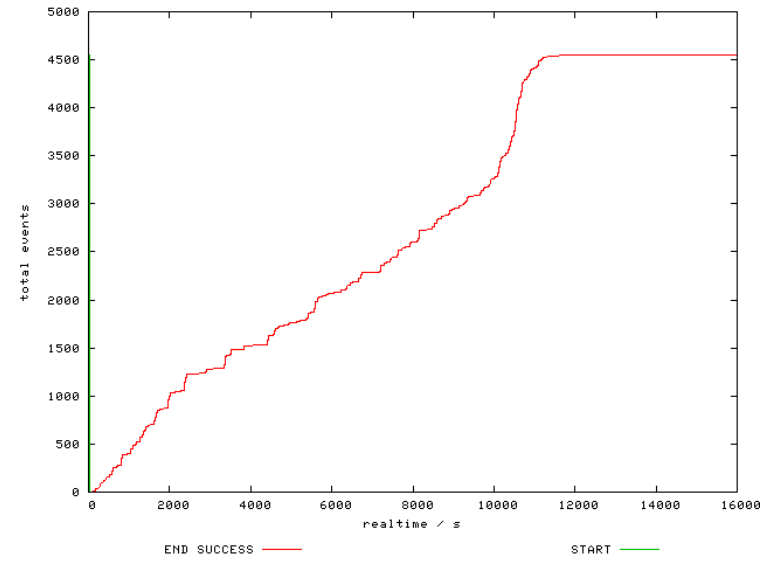
swift-plot-log mylog-01234456-8922-abcdef.log
It will generate a directory report-mylog-01234456-8922-abcdef/

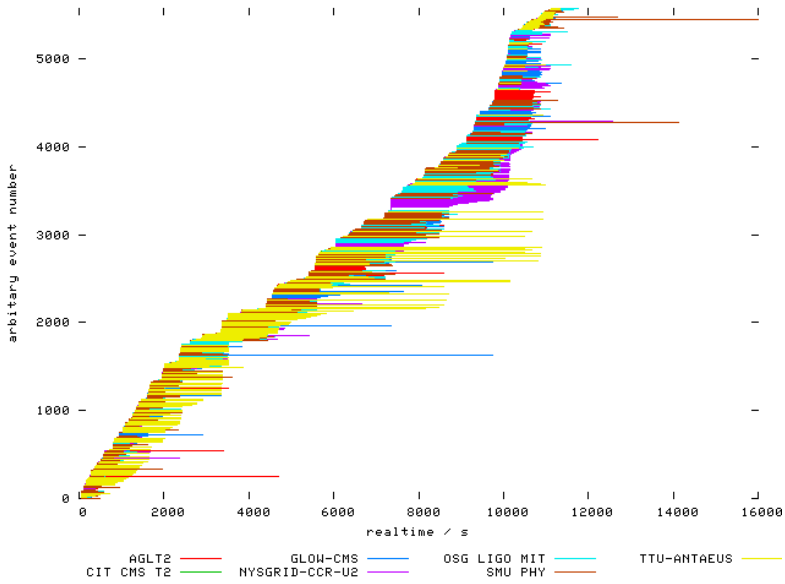
```
mv report-mylog-01234456-8922-abcdef    ~$USER/public_html/
```

So that you can view it with a web browser at, http://XXX.XXX.XXX/~$USER/report-mylog-01234456-8922-abcdef/

❖ Example A. 4554 jobs on 7 sites

site	JOB_START	JOB_END	APPLICATION_EXCEPTION	JOB_CANCELED	unknown	total
AGLT2	0	985	4	89	0	1078
CIT_CMS_T2	0	0	20	2	0	22
GLOW-CMS	0	1160	106	194	1	1461
NYSGRID-CCR-U2	0	841	1	335	0	1177
OSG_LIGO_MIT	0	877	1	106	0	984
SMU_PHY	0	522	0	37	0	559
TTU-ANTAEUS	0	168	1	122	1	292





3.2 Blast (Basic Local Alignment Search Tool)

1. Blast application running introduction

This application, executed on the OSG grid sites, has the maximum 500K sequences. The computation needs the database and the sequence input files.

Database:
[houzx@communicado pir]$ ls /disks/ci-gpfs/swift/blast/pir/
UNIPROT_for_blast_14.0.seq UNIPROT_for_blast_14.0.seq.01.psq
UNIPROT_for_blast_14.0.seq.00.phr UNIPROT_for_blast_14.0.seq.02.phr
UNIPROT_for_blast_14.0.seq.00.pin UNIPROT_for_blast_14.0.seq.02.pin
UNIPROT_for_blast_14.0.seq.00.psq UNIPROT_for_blast_14.0.seq.02.psq
UNIPROT_for_blast_14.0.seq.01.phr UNIPROT_for_blast_14.0.seq.pal
UNIPROT_for_blast_14.0.seq.01.pin

[houzx@communicado pir]$ cat seqlist.txt
seqid seqfile
A0AQI4
VWLRRCTHYLFIVVVAVNSTLLTINAGDYIFYTDWAWTSYTVFSISQTLMLIVGATYYLTFTGVPG
TATYYALIMTVYTWIAKAAWFSLGYPYDFIVTPVWLPSAMLLDLVYWATKKNKHSLILFGGVLV
GMSLPLSNMVNLITVADPLETAFKYPRPTLPPYMTPIEPQVGKFYNSPVALGAGAGAVLGCTFAAL
GCKLNTWTYR
… …

2. Steps for running the Blast application on the OSG grid sites by Swift

(1) Download and setup adem-osg toolkits
(It just needs one time. Please refer to Part I)

(2) Get the available grid sites
auto-get-sites $GRID $VO (get the available grid sites within a given virtual organization in osg or osg-itb)
e.g. auto-get-sites osg osg

(3) Get the sites.xml for swift
swift-auto-sites-xml *$GRID $VO*
e.g. swift-auto-sites-xml osg osg
(swift-sites.xml will be generated in $ADEM_HOME/swift-execution-example for different users, e.g. sites-10.xml)

(4) prepare-for-blast-swift-submit
Include: Transfer the Database (pir-db.tar.gz) to the grid sites; prepare blast-tc.data (e.g. tc-blast.data); Modify and transfer the execution wrapper scripts pirblast.sh (e.g. pirblast.sh-grow.cs.uni.edu) to the grid sites
cd $ADEM_HOME/swift-execution-example/blast
./prepare-for-blast-swift-submit $VO $Grid-sites-file
(e.g. ./prepare-for-blast-swift-submit osg $ADEM_HOME/logs/sites/osg-osg-avail-sites-$DATE.txt)

(5) Edit the swift code and pick the sequence input files
[houzx@communicado pir]$ cat blast-pir.swift
type file;
type BlastOut;

```
(BlastOut tarout) blastcompute (string seqid, string infile) {
  app {
    pirblast seqid infile @tarout;
    }
}

type params {
      string  seqid;
      string  seqfile;
}

doall(params pset[])
{
  foreach params,i in pset {

  BlastOut tout <single_file_mapper; file=@strcat(pset[i].seqid,"-result.tar.gz")>;

  (tout) =  blastcompute(pset[i].seqid,pset[i].seqfile);

 }
}

params p[];
p = readData("seqlist.txt");
doall(p);
```

[houzx@communicado pir]$ cat seqlist.txt

```
seqid seqfile
A0AQI4
VWLRRCTHYLFIVVVAVNSTLLTINAGDYIFYTDWAWTSYTVFSISQTLMLIVGATYYLTFTGVPG
TATYYALIMTVYTWIAKAAWFSLGYPYDFIVTPVWLPSAMLLDLVYWATKKNKHSLILFGGVLV
GMSLPLSNMVNLITVADPLETAFKYPRPTLPPYMTPIEPQVGKFYNSPVALGAGAGAVLGCTFAAL
GCKLNTWTYR
… …
```

(6) Submit the job

[houzx@communicado pir]$ swift -tc.file blast-20-tc.data -sites.file swift-13-sites.xml blast-pir.swift

3. Experiment Results

- How to plot for the results log file
  (Please refer to the same part for dock application)

- 10,000 jobs on 13 grid sites

Appendix A: selected_ spheres.sph

```
[houzx@login databases]$ cat Targets/NaturalLigands/1F9Y/selected_spheres.sph
DOCK spheres within 10 ang of ligands
cluster     1   number of spheres in cluster   158
  980 -13.37595 -12.23035   0.03192   1.451 1668 0  0
  988 -14.35976 -13.72597  -1.17619   2.294 1667 0  0
 1030 -10.90828  -2.53670   5.07282   2.350 1497 0  0
 1037 -10.14357  -2.82658   4.56714   2.636 1497 0  0
 1054 -23.76624  -4.84650   4.76178   3.600 1169 0  0
 1092 -21.36982  -3.27482   4.97162   1.723 1177 0  0
 1095 -20.91722  -2.30599   2.41474   2.636 1169 0  0
 1100 -18.98822  -0.04254   3.42409   2.057 1273 0  0
 1101 -13.55514  -1.84702   6.79415   1.605 1475 0  0
 1103 -11.16369  -1.37966   3.63207   3.018 1497 0  0
 1104 -13.43506  -3.79000   0.37097   2.090 1695 0  0
 1106 -22.52031  -3.57877   6.19879   1.990 1177 0  0
 1154 -12.93507   0.14080   5.24269   1.934 1285 0  0
 1155 -18.12812   0.87900   5.05492   1.773 1175 0  0
 1156 -12.07738  -0.30761   6.43434   1.569 1497 0  0
 1170 -20.95113   3.70520   0.44641   3.260 1276 0  0
 1176 -20.94291   0.70000  -0.07435   3.803 1272 0  0
 1177 -23.40052  -4.94163   3.56824   3.938 1048 0  0
 1180 -20.39265   1.02740   1.01889   3.009 1271 0  0
 1231 -13.70166   1.49674   4.94854   1.437 1282 0  0
 1238 -18.12663   1.99491   5.40664   1.418 1180 0  0
 1245 -10.58430   5.09800   1.10813   2.424 1372 0  0
 1246  -9.21725   0.72006   2.31312   3.622 1814 0  0
 1255 -21.27957   5.10583   2.13466   2.231 1170 0  0
 1270 -12.51257   2.02400   2.98830   1.538 1246 0  0
 1273 -19.29805   0.01400   3.02573   2.259 1169 0  0
 1274 -20.64592   4.33608   1.62902   2.384 1170 0  0
 1275 -12.68553   7.45300   2.68934   1.468 1245 0  0
 1277 -11.59949   5.93800  -0.31353   2.469 1346 0  0
 1282 -12.75248   1.55129   3.72515   1.630 1246 0  0
 1285 -11.55845  -0.02810   3.91850   2.779 1497 0  0
 1286 -11.96993  -2.47429   1.48864   2.882 1695 0  0
 1290 -11.86792   6.85788   1.02810   2.323 1245 0  0
 1296 -11.83228   8.15767   2.22823   2.065 1425 0  0
 1298 -12.04779   9.43125   1.48649   1.541 1322 0  0
 1322 -11.57399   6.59747   0.95089   2.389 1245 0  0
 1346 -11.48148   4.65852  -1.75962   3.140 1281 0  0
 1364  -9.17692   2.52067   1.31493   3.358 1246 0  0
 1372 -10.89615   6.26762   1.58194   2.086 1245 0  0
 1376  -9.23917   3.15083   1.78721   2.825 1246 0  0
 1377  -9.23189   2.20705   1.17633   3.539 1246 0  0
 1384 -12.31823   9.11155   2.88533   1.599 1296 0  0
 1392 -12.65981   8.41901   3.51082   1.813 1296 0  0
 1413 -11.69820   7.35492   2.83211   1.616 1245 0  0
 1417 -13.55926   8.43500   4.02830   1.525 1275 0  0
 1425 -11.73940   7.55966   2.24185   2.019 1245 0  0
 1426 -10.80620   6.01971   2.25954   1.626 1245 0  0
 1466 -12.64708   0.46726   5.09053   2.084 1285 0  0
 1475 -12.63574  -1.69400   5.94771   1.920 1103 0  0
 1496  -9.62628  -2.29191   2.75166   3.798 1695 0  0
```

1498 -7.54548 1.60810 3.74668 2.522 1814 0 0
1500 -10.16236 -3.43974 5.18863 2.196 1037 0 0
1511 -10.09952 -4.58300 5.62340 1.698 1037 0 0
1513 -8.80997 -5.60583 5.90855 1.460 1511 0 0
1624 -0.60454 -10.88708 1.20577 1.862 1655 0 0
1655 -0.55743 -10.15869 1.45661 2.170 1777 0 0
1667 -14.32115 -13.71784 -1.13805 2.263 988 0 0
1668 -13.30928 -12.33531 0.08923 1.421 980 0 0
1672 -14.75848 -13.99360 -1.50880 2.663 988 0 0
1676 -14.16067 -14.02700 -0.77908 2.099 988 0 0
1683 -16.33629 -6.64645 -4.44170 2.222 1708 0 0
1689 -16.75131 -11.30754 -3.97460 3.233 1709 0 0
1690 -14.84088 -2.71639 -3.99098 3.788 1283 0 0
1694 -8.71573 -5.62057 -0.51390 1.722 1745 0 0
1698 -16.30116 -12.07850 -0.34013 1.691 988 0 0
1704 -12.72100 -2.46841 -2.06574 3.269 1283 0 0
1705 -14.24590 -12.10009 -0.85540 1.937 1667 0 0
1707 -15.59803 -3.47665 -1.88850 2.766 1283 0 0
1708 -16.96818 -5.77492 -4.45067 2.799 1685 0 0
1709 -17.65809 -10.41299 -4.72869 3.592 1683 0 0
1711 -17.18399 -3.47148 -0.64360 2.549 1283 0 0
1722 -9.54262 -4.98516 5.53822 1.697 1511 0 0
1730 -9.12302 -4.28029 4.83094 2.396 1511 0 0
1738 -6.43694 -4.33318 3.44258 1.828 1786 0 0
1739 -5.56288 -3.91887 -0.83706 2.129 1801 0 0
1740 -5.39226 -3.84954 -2.16763 2.819 1801 0 0
1741 -0.14326 -9.20658 1.89590 2.151 1777 0 0
1742 -7.79600 -3.64399 2.93929 2.606 1786 0 0
1744 -6.78752 -3.52726 0.69261 2.095 1803 0 0
1745 -8.74478 -5.65122 -0.95120 1.742 1680 0 0
1746 -1.95240 -5.99381 -1.35874 1.471 1800 0 0
1748 -0.81903 -9.56500 2.13578 1.787 1777 0 0
1752 -1.35695 -10.66946 1.74453 1.816 1655 0 0
1782 2.09666 -5.43399 4.86115 1.625 1860 0 0
1786 -9.47374 -2.34300 2.66769 3.801 1695 0 0
1794 -8.89854 -4.17420 4.58278 2.540 1730 0 0
1798 2.29235 -6.32791 3.90714 2.188 1861 0 0
1800 -4.28765 -4.84477 -1.83570 2.083 1740 0 0
1801 -5.67054 -3.86202 -0.48407 2.013 1744 0 0
1803 -8.67108 -2.74011 0.76294 3.614 1695 0 0
1809 0.63401 -8.66940 1.85730 2.574 1777 0 0
1810 -1.00655 -6.58326 -0.73901 1.606 1746 0 0
1811 -5.35282 -4.02404 -0.00382 1.628 1744 0 0
1826 3.24102 -2.92968 4.14291 1.401 1857 0 0
1852 2.17045 -6.26151 4.44445 2.031 1782 0 0
1855 2.35977 -4.71512 4.59827 1.497 1798 0 0
1860 2.48246 -5.12400 4.50762 1.736 1799 0 0
1864 2.30343 -7.53865 4.85802 1.415 1773 0 0
700 -3.51783 10.78924 -1.98694 2.013 1348 0 0
1343 -0.63592 7.14704 -2.65151 2.250 1370 0 0
1352 -1.53174 7.71529 -1.95771 1.646 1370 0 0
1354 0.73510 3.53300 -1.78394 2.194 1371 0 0
1357 0.22682 3.35607 -1.21966 1.704 1381 0 0
1368 -3.11309 9.75814 -1.24695 1.888 1348 0 0
1369 -0.22613 2.49757 0.10663 1.400 1815 0 0
1370 -1.70575 8.33642 -1.91657 1.827 1352 0 0

1371	0.69831	4.94529	-2.36043	2.243	1354	0	0
1374	-3.66893	9.17601	-0.77707	1.400	1348	0	0
1381	3.91884	4.93100	1.89195	2.880	1879	0	0
1815	1.06095	1.10764	-0.13578	1.651	1823	0	0
1822	0.43557	1.89315	0.10550	1.472	1815	0	0
1823	2.39567	3.28041	0.65252	1.873	1381	0	0
1824	2.78995	4.86623	3.22600	1.901	667	0	0
1872	3.44275	4.47919	5.07100	1.412	676	0	0
1879	4.31951	5.43834	3.02990	2.422	674	0	0
87	-5.93546	2.75994	10.90610	1.811	826	0	0
91	-5.39556	2.66767	10.95830	1.715	1930	0	0
95	-6.74151	2.77249	10.77042	1.979	1467	0	0
118	-7.11970	5.39900	9.38253	1.727	1424	0	0
624	-2.67469	8.51742	10.81342	1.502	1919	0	0
656	-4.92246	10.80631	5.94818	1.400	1361	0	0
669	-3.88222	8.13631	8.87575	1.711	1923	0	0
672	-4.61857	8.81186	6.60355	1.591	783	0	0
679	-4.77014	6.39600	5.83593	1.787	1364	0	0
778	-3.40262	7.87390	10.49507	1.664	1920	0	0
779	-5.61395	6.07152	9.42367	1.737	1922	0	0
780	-3.99928	8.06119	8.83209	1.731	1923	0	0
781	-5.98606	5.60124	8.62967	2.055	1922	0	0
782	-4.18452	7.91139	8.72445	1.745	1923	0	0
783	-5.33614	6.55858	6.75573	2.444	1923	0	0
784	-5.20737	9.73946	5.93604	1.436	1361	0	0
819	-3.51899	8.05938	10.85594	1.421	778	0	0
820	-5.45688	2.76777	11.66037	1.456	91	0	0
826	-5.85784	2.75278	10.78237	1.859	87	0	0
1359	-6.14424	7.79976	5.88290	1.731	784	0	0
1361	-4.92291	10.80659	5.94795	1.400	656	0	0
1363	-5.44567	7.14893	6.21658	2.244	783	0	0
1373	-5.87196	7.38889	5.98240	2.000	1411	0	0
1375	-5.56068	7.50074	6.09919	2.026	783	0	0
1411	-5.78365	6.12203	6.26700	2.163	1364	0	0
1412	-5.62229	6.01400	6.47956	2.349	1364	0	0
1423	-5.53868	5.90493	6.87963	2.409	1923	0	0
1467	-6.11587	5.06492	8.63866	2.082	1922	0	0
1890	-4.47831	5.29403	5.32591	1.411	1364	0	0
1902	-5.52563	5.79601	6.65199	2.403	1423	0	0
1905	-4.69657	5.11872	5.33258	1.433	1364	0	0
1919	-3.21799	8.23139	10.33662	1.696	778	0	0
1920	-3.80815	7.57023	10.54809	1.564	778	0	0
1921	-3.38844	8.34482	9.39567	1.676	780	0	0
1922	-5.98663	5.02678	7.99520	2.162	1467	0	0
1923	-5.54677	5.78304	6.89450	2.392	1423	0	0
1930	-5.26915	2.77815	10.71706	1.791	826	0	0
1713	-19.54521	-11.81821	2.44680	1.533	1056	0	0
1648	-10.70100	-15.24635	4.23122	1.400	983	0	0
1653	-4.90723	-9.46866	-2.67031	1.639	1740	0	0
1658	-5.55985	-9.53729	-2.68367	1.582	1740	0	0
1660	-4.55500	-9.10553	-3.26450	2.026	1740	0	0
1666	-5.78608	-9.57718	-4.09368	2.119	1660	0	0

Appendix B: dockKAGxNL.tar.gz

```
[houzx@login tmp]$ ls dockKAGxNL
parameters  targets  template.in
[houzx@login dockKAGxNL]$ ls parameters/
flex.defn  flex_drive.tbl  vdw_AMBER_parm99.defn
[houzx@login dockKAGxNL]$ ls targets/
1F9Y  1G97  1JBW  1KQP  1KZL  1OD6  1SNN  1SUW  1VHT
[houzx@login dockKAGxNL]$ cat template.in
ligand_atom_file                        @INPUT_LIGANDMOL2@
limit_max_ligands                       no
skip_molecule                           no
read_mol_solvation                      no
calculate_rmsd                          no
orient_ligand                           yes
automated_matching                      yes
receptor_site_file                      @RECEPTOR_DIR@/selected_spheres.sph
max_orientations                        1000
critical_points                         no
chemical_matching                       no
use_ligand_spheres                      no
flexible_ligand                         no
bump_filter                             no
score_molecules                         yes
contact_score_primary                   no
contact_score_secondary                 no
grid_score_primary                      yes
grid_score_secondary                    no
grid_score_rep_rad_scale                1
grid_score_vdw_scale                    1
grid_score_es_scale                     1
grid_score_grid_prefix                  @RECEPTOR_DIR@/grid
dock3.5_score_secondary                 no
continuous_score_secondary              no
gbsa_zou_score_secondary                no
gbsa_hawkins_score_secondary            no
amber_score_secondary                   no
minimize_ligand                         yes
simplex_max_iterations                  1000
simplex_max_cycles                      1
simplex_score_converge                  0.1
simplex_cycle_converge                  1.0
simplex_trans_step                      1.0
simplex_rot_step                        0.1
simplex_tors_step                       10.0
simplex_final_min                       no
simplex_random_seed                     0
atom_model                              all
vdw_defn_file                           @PARAMETER_DIR@/vdw_AMBER_parm99.defn
flex_defn_file                          @PARAMETER_DIR@/flex.defn
flex_drive_file                         @PARAMETER_DIR@/flex_drive.tbl
ligand_outfile_prefix                   1
write_orientations                      no
num_scored_conformers_written           1
rank_ligands                            no
```

Appendix C: rundock

```
[houzx@login bin]$ cat rundock-ce01.cmsaf.mit.edu
#! /bin/bash
PATH=/bin:/usr/bin:$PATH

#
# rundock runid ligfile outfile targetlist
#    targetlist can be "all"
#

if [ $# -lt 2 ]; then
  echo usage: $0 ligands-file target-list
  exit 0
fi

# Get dock parameters from command line

ligpath=$1
targets=$2

time=`which time` # to get [/fuse]/usr/bin/time, not time builtin
tmp=`/bin/pwd`
userdir=/tmp
dock=$userdir/dock
dockpack=/osg/data/osg/houzx/dockKAGxNL.tar.gz
# Dock app version
dock6app=/net/t2dsk0001/d00/osg/app/osg/houzx/work-pac/dock6/bin/dock6
loadfile()
{
    DIR_RAM="$1"
    ARCHIVE="$2"

        mkdir $DIR_RAM
        cd $DIR_RAM
        tar zxf $ARCHIVE

 EXIT_CODE=$?

 if [ "${EXIT_CODE}" -ne "0" ]; then
    echo "Error in untar of ${ARCHIVE} to ${DIR_RAM}/... exit code ${EXIT_CODE}
"
    exit ${EXIT_CODE}
 fi

}

subval()  # insert parameter values into the parameter template file
{
  repl=""
  for v in $*; do
    echo substituting $v = ${!v}
    val=${!v}
    repl="$repl -e 's,@${v}@,$val,'"
  done
  echo Running "sed $repl <$paramtemplate >$paramfile"
  eval sed $repl <$paramtemplate >$paramfile
}

loadfile $dock $dockpack
cd $tmp
cd ../../..
```

```
temp=`/bin/pwd`

ligfile=`basename $ligpath`
lig=$temp/shared/$ligpath

work=`mktemp -d $userdir/dock.XXXXXX`
cd $work
#dd if=$lig of=$ligfile bs=131072
INPUT_LIGANDMOL2=$lig
PARAMETER_DIR=$dock/parameters
paramtemplate=$dock/template.in
if [ "$targets" = all ]; then
  targetdirs="$dock/targets/*"
else
  targetdirs=""
  for t in $targets; do
    tdir=$dock/targets/$t
    if [ -d $tdir ]; then
      targetdirs="$targetdirs $tdir"
    else
      echo $0: Warning: target directory $tdir not found - skipping
    fi
  done
fi

echo "ce01.cmsaf.mit.edu,$ligfile"
echo $0: Docking ligands against targets $targets

for RECEPTOR_DIR in $targetdirs; do
  target=`basename $RECEPTOR_DIR`
  paramfile=$work/dock.$target.in
  outfile=$work/dock.$target.out
  logfile=$work/dock.$target.log

  # insert parameter values into the parameter template file
  subval INPUT_LIGANDMOL2 RECEPTOR_DIR PARAMETER_DIR

  # Run dock on target

  out=dock.$target.out
  echo DOCK starting at `date` on target $target >$out 2>&1
  echo
  ls -l $paramtemplate >>$logfile 2>&1
  ls -l $paramfile >>$logfile 2>&1
  echo
  $time $dock6app -i $paramfile -o $outfile >> $logfile 2>&1
  dockrc=$?
  echo
  mv 1_scored.mol2 $target.scored.mol2
  echo DOCK ended at `date` on target $target RC=$dockrc >>$logfile 2>&1
ls -lt

done

# Copy output files back to persistent location
#rm $ligfile
tar zcf result.tar.gz *
/bin/mv result.tar.gz $tmp/$ligfile-result.tar.gz
#$time dd if=result.tar.gz of=$ofile bs=128k >>dock.debug 2>&1
#echo dd of run $runid `cat dock.debug`

cd /
```

```
rm -r $work
#FIXME - delete output files once copied to make space for next jobs

exit 0
```

Appendix D: sites-10.xml

```
<config>

<!-- sites.xml specifies details of the sites that Swift can run on.

The first entry entry, for localhost, should work on most linux-like systems
without any change.

It may be necessary to change the two occurences of /var/tmp to a different
working directory.

-->

  <pool handle="OSG_LIGO_MIT" sysinfo="INTEL32::LINUX">
    <gridftp url="gsiftp://osg-ligo.mit.edu" storage="/opt/storage/app/houzx" major="2" minor="2"
patch="4">
    </gridftp>
    <jobmanager universe="vanilla" url="osg-ligo.mit.edu/jobmanager-condor" major="2" minor="2"
patch="4" />
    <workdirectory >/opt/storage/app/houzx</workdirectory>
  </pool>

  <pool handle="UFlorida-HPC" sysinfo="INTEL32::LINUX">
    <gridftp  url="gsiftp://iogw1.hpc.ufl.edu" storage="/scratch/ufhpc/osg/data/houzx" major="2" minor="2"
patch="4">
    </gridftp>
    <jobmanager universe="vanilla" url="iogw1.hpc.ufl.edu/jobmanager-condor" major="2" minor="2"
patch="4" />
    <workdirectory >/scratch/ufhpc/osg/data/houzx</workdirectory>
  </pool>

  <pool handle="Nebraska" sysinfo="INTEL32::LINUX">
    <gridftp  url="gsiftp://red.unl.edu" storage="/opt/osg/data/houzx" major="2" minor="2" patch="4">
    </gridftp>
    <jobmanager universe="vanilla" url="red.unl.edu/jobmanager-pbs" major="2" minor="2" patch="4" />
    <workdirectory >/opt/osg/data/houzx</workdirectory>
  </pool>
  <pool handle="GLOW" sysinfo="INTEL32::LINUX">
    <gridftp url="gsiftp://cmsgrid01.hep.wisc.edu" storage="/afs/hep.wisc.edu/osg/home/osg_edu/houzx"
major="2" minor="2" patch="4">
    </gridftp>
    <jobmanager universe="vanilla" url="cmsgrid01.hep.wisc.edu/jobmanager-condor" major="2"
minor="2" patch="4" />
    <workdirectory >/afs/hep.wisc.edu/osg/home/osg_edu/houzx</workdirectory>
  </pool>

    <pool handle="SBGrid-Harvard-Exp" sysinfo="INTEL32::LINUX">
    <gridftp url="gsiftp://abitibi.sbgrid.org" storage="/se/data/vo/osg/houzx" major="2" minor="2"
patch="4">
    </gridftp>
```

```
   <jobmanager universe="vanilla" url="abitibi.sbgrid.org/jobmanager-condor" major="2" minor="2" patch="4" />
   <workdirectory >/se/data/vo/osg/houzx</workdirectory>
  </pool>

  <pool handle="MIT_CMS" sysinfo="INTEL32::LINUX">
   <gridftp url="gsiftp://ce01.cmsaf.mit.edu" storage="/osg/data/houzx" major="2" minor="2" patch="4">
   </gridftp>
   <jobmanager universe="vanilla" url="ce01.cmsaf.mit.edu/jobmanager-condor" major="2" minor="2" patch="4" />
   <workdirectory >/osg/data/houzx</workdirectory>
  </pool>

  <pool handle="CIT_CMS_T2" sysinfo="INTEL32::LINUX">
   <gridftp url="gsiftp://cit-gatekeeper.ultralight.org" storage="/raid2/osg-data/houzx" major="2" minor="2" patch="4">
   </gridftp>
   <jobmanager universe="vanilla" url="cit-gatekeeper.ultralight.org/jobmanager-condor" major="2" minor="2" patch="4" />
   <workdirectory >/raid2/osg-data/houzx</workdirectory>
  </pool>

  <pool handle="GLOW-CMS" sysinfo="INTEL32::LINUX">
   <gridftp url="gsiftp://cmsgrid02.hep.wisc.edu" storage="/afs/hep.wisc.edu/osg/data/houzx" major="2" minor="2" patch="4">
   </gridftp>
   <jobmanager universe="vanilla" url="cmsgrid02.hep.wisc.edu/jobmanager-condor" major="2" minor="2" patch="4" />
   <workdirectory >/afs/hep.wisc.edu/osg/data/houzx</workdirectory>
  </pool>

  <pool handle="UCSDT2-B" sysinfo="INTEL32::LINUX">
   <gridftp url="gsiftp://osg-gw-4.t2.ucsd.edu" storage="/osgfs/data/houzx" major="2" minor="2" patch="4">
   </gridftp>
   <jobmanager universe="vanilla" url="osg-gw-4.t2.ucsd.edu/jobmanager-condor" major="2" minor="2" patch="4" />
   <workdirectory >/osgfs/data/houzx</workdirectory>
  </pool>

   <pool handle="NYSGRID-CCR-U2" sysinfo="INTEL32::LINUX">
    <gridftp url="gsiftp://u2-grid.ccr.buffalo.edu" storage="/san/scratch/grid/grid-tmp/grid-data/houzx" major="2" minor="2" patch="4">
    </gridftp>
    <jobmanager universe="vanilla" url="u2-grid.ccr.buffalo.edu/jobmanager-condor" major="2" minor="2" patch="4" />
    <workdirectory >/san/scratch/grid/grid-tmp/grid-data/houzx</workdirectory>
   </pool>
  </config>
```

Appendix E: tc-dock.data

```
# sitename transformation path INSTALLED platform profiles
localhost rundock /home/houzx/dock-run/bin/rundock INSTALLED INTEL32::LINUX null
```

OSG_LIGO_MIT rundock /opt/storage/data/osg/houzx/rundock INSTALLED INTEL32::LINUX null
UFlorida-HPC rundock /scratch/ufhpc/osg/data/osg/houzx/rundock INSTALLED INTEL32::LINUX null
Nebraska rundock /opt/osg/data/osg/houzx/rundock INSTALLED INTEL32::LINUX null
GLOW rundock /afs/hep.wisc.edu/osg/data/osg/houzx/rundock INSTALLED INTEL32::LINUX null
SBGrid-Harvard-Exp rundock /se/data/vo/osg/houzx/rundock INSTALLED INTEL32::LINUX null
MIT_CMS rundock /osg/data/osg/houzx/rundock INSTALLED INTEL32::LINUX null
CIT_CMS_T2 rundock /raid2/osg-data/osg/houzx/rundock INSTALLED INTEL32::LINUX null
GLOW-CMS rundock /afs/hep.wisc.edu/osg/data/osg/houzx/rundock INSTALLED INTEL32::LINUX null
#gk04.swt2.uta.edu rundock /ifs1/osg/app/osg/houzx/rundock INSTALLED INTEL32::LINUX null
UCSDT2 rundock /osgfs/data/osg/houzx/rundock INSTALLED INTEL32::LINUX null
UCSDT2-B rundock /osgfs/data/osg/houzx/rundock INSTALLED INTEL32::LINUX null
NWICG_NotreDame rundock /dscratch/osg/data/osg/houzx/rundock INSTALLED INTEL32::LINUX null
NYSGRID-CCR-U2 rundock /san/scratch/grid/grid-tmp/grid-data/osg/houzx/rundock INSTALLED INTEL32::LINUX null
TTU-ANTAEUS rundock /mnt/hep/osg/houzx/rundock INSTALLED INTEL32::LINUX null
NWICG_NDCCL rundock /dscratch/osg/data/osg/houzx/rundock INSTALLED INTEL32::LINUX null
AGLT2 rundock /atlas/data08/OSG/DATA/houzx/rundock INSTALLED INTEL32::LINUX null
UNI-OSGEDU rundock /data/houzx/rundock INSTALLED INTEL32::LINUX null
cinvestav rundock /raid/osg_data/osg/houzx/rundock INSTALLED INTEL32::LINUX null
SMU_PHY rundock /data/osg/houzx/rundock INSTALLED INTEL32::LINUX null
#uscms1.fltech-grid3.fit.edu rundock /mnt/nas0/OSG/DATA/houzx/rundock INSTALLED INTEL32::LINUX null
isuhep rundock /home/grid/data/osg/houzx/rundock INSTALLED INTEL32::LINUX null

[1] KEGG's Ligand Database: http://www.genome.ad.jp/kegg/ligand.html, 2008.

[2] D.T. Moustakas et al. "Development and Validation of a Modular, Extensible Docking Program: DOCK 5," J. Comput. Aided Mol. Des. 20, 2006, pp. 601-619.

Appendix F: tc-blast.data

[houzx@login bin]$ cat ../swift-execution-example/blast/blast-20-tc.data
#sitename transformation path INSTALLED platform profiles
Purdue-Steele pirblast /apps/osg/houzx/work-pac/pirblast.sh INSTALLED INTEL32::LINUX null
RENCI-Engagement pirblast /nfs/osg-app/osg/houzx/work-pac/pirblast.sh INSTALLED INTEL32::LINUX null
AGLT2 pirblast /atlas/data08/OSG/APP/osg/houzx/work-pac/pirblast.sh INSTALLED INTEL32::LINUX null
GROW-UNI-P pirblast /app/osg/houzx/work-pac/pirblast.sh INSTALLED INTEL32::LINUX null
NWICG_NotreDame pirblast /dscratch/osg/app/osg/houzx/work-pac/pirblast.sh INSTALLED INTEL32::LINUX null
Clemson-ciTeam pirblast /export/osg/app/osg/houzx/work-pac/pirblast.sh INSTALLED INTEL32::LINUX null
SBGrid-Harvard-East pirblast /osg/storage/app/osg/houzx/work-pac/pirblast.sh INSTALLED INTEL32::LINUX null

OSG_LIGO_MIT pirblast /opt/storage/app/osg/houzx/work-pac/pirblast.sh INSTALLED INTEL32::LINUX null
UTA_SWT2 pirblast /ifs1/osg/app/osg/houzx/work-pac/pirblast.sh INSTALLED INTEL32::LINUX null
TTU-ANTAEUS pirblast /mnt/lustre/antaeus/apps/osg/houzx/work-pac/pirblast.sh INSTALLED INTEL32::LINUX null
FLTECH pirblast /mnt/nas0/OSG/APP/osg/houzx/work-pac/pirblast.sh INSTALLED INTEL32::LINUX null
SPRACE-CE pirblast /home/OSG_app/app/osg/houzx/work-pac/pirblast.sh INSTALLED INTEL32::LINUX null
LIGO_UWM_NEMO pirblast /opt/osg/app/osg/houzx/work-pac/pirblast.sh INSTALLED INTEL32::LINUX null
UFlorida-HPC pirblast /scratch/ufhpc/osg/app/osg/houzx/work-pac/pirblast.sh INSTALLED INTEL32::LINUX null

Appendix G: blast wrapper script

```
[houzx@login bin]$ cat ../swift-execution-example/blast/wrapper-bin/pirblast.sh-grow.cs.uni.edu
#!/bin/bash

#TMP=`mktemp -d /dev/shm/pir.blast.d.XXXXXX`
TMP=`pwd`
cd $TMP

SEQID=$1
SEQUENCE=$2

PIR=/data/osg/houzx/pir
blastapp=/app/osg/houzx/work-pac/blast-2.2.10/bin/blastall

OUTDIR=/tmp/out

DB="$PIR/UNIPROT_for_blast_14.0.seq.00 $PIR/UNIPROT_for_blast_14.0.seq.01 $PIR/UNIPROT_for_blast_14.0.seq.02"

echo ">"${SEQID} >blast.query
echo $SEQUENCE >>blast.query

unset LD_LIBRARY_PATH
if [ ! -d $OUTDIR ]
then
  mkdir -p $OUTDIR
fi
time $blastapp -p blastp -F F -d "$DB" -i blast.query -v 300 -b 300 -m8 -o $OUTDIR/$SEQID.out 2>$OUTDIR/$SEQID.err

cd $OUTDIR
tar zcf result.tar.gz *
/bin/mv result.tar.gz $TMP/$SEQID-result.tar.gz
/bin/rm -rf $OUTDIR
```