Information Technology
Infrastructure Branch

WEXAC workshop

Presented by: Vadim Malkin
Back to the Future:
From WEIZAC to WEXAC
WEIZAC, the first Israeli computer was fully operational in 1957.
60 years later :
The WEXAC (Weizmann EXAscale cluster) cluster in 2017, also fully operational.
60 years later (contd.):
The WEXAC (Weizmann EXAscale cluster) cluster in 2017, also fully operational.
The WEXAC (Weizmann EXAscale cluster) cluster is a part of the new data center.
Current WEXAC Status

• Computing power: more than 300 servers, above 6000 cores – the biggest Weizmann Institute of Science campus cluster

• Isilon Storage capacity: 1.6 PByte total capacity, 1.05 PByte in use
  • 633 TB in StorWEX, 522 TB in StorWIS

• Modules: more than 600 scientific applications with different versions installed in WEXAC.

• Migration to LSF10 and CentOS 7: Done at Mar, 2017.

• ScaleMP: 1TB RAM, 128 compute cores in single machine
Best practices to compile serial and parallel applications.

- Use Latest Intel Compiler – `module load icompiler/2017`
- Use the newest servers for compilation
- Use a fast interconnect (Infiniband)
- In parallel application cases use a MPI library with good collectiveness (MVAPICH2 or Intel MPI)
- Don’t re-invent the wheel – use optimized libraries, like NetCDF, HDF5, Intel MKL
- Use a last FFT library (FFTW or Intel FFTW wrappers)
- Use productivity tools like DDT, mpiP, OpenMPI, STAT for debugging and application profiling
Ganglia Monitoring and Graphics Tool

• Ganglia monitoring tool is primarily built for monitoring clusters of servers, and it does its job at the best.

• It represents the overall performance of a cluster of servers in an overview.

• Suppose you have multiple clusters in a datacenter, then in that case you can arrange them and call it as a grid and have an overall performance overview of that grid.

• Performance monitoring to the base machine of a collection of virtual machines, can be done using ganglia monitoring tool.
MobaXterm for Windows users

• Enhanced terminal for Windows with X11 server, tabbed SSH client, network tools and much more

• Download: http://mobaxterm.mobatek.net/download.html
New WEXAC Access Servers

access.wexac.weizmann.ac.il
access1.wexac.weizmann.ac.il
access2.wexac.weizmann.ac.il
access3.wexac.weizmann.ac.il
MobaXterm User Interface

- MobaXterm Professional v8.6
  (X server, SSH client and network tools)

- Your computer drives are accessible through the /drives path
- Your DISPLAY is set to 132.77.134.71:0.0
- When using SSH, your remote DISPLAY is automatically forwarded
- Your HOME folder is not persistent: it will be erased on restart
- Each command status is specified by a special symbol (√ or ×)

Registered to Weizmann Institute of Science (6 users)

[2017-02-12 17:48.47] ~
[vadim.vadim-laptop] ➤
MobaXterm – Create SSH session
MobaXterm – WEXAC prompt

[vadim.vadim-laptop] > ssh vadimm@access.wexac.weizmann.ac.il

Welcome to access.
This server has been upgraded to RedHat 7.3.
If you have any problems write an email to hpc@weizmann.ac.il.
Enjoy.

Last login: Sun Feb 12 17:48:00 2017 from v-lap.weizmann.ac.il
[vadimm@access ~]$
Map Network Drive – Windows workstation
Enter Lab Name (StorWEX)
Enter Lab Name (StorWIS)
Universal WIS credentials
How StorWEX / StorWIS drives appears on Windows workstation
Access WEXAC from Ubuntu
Access WEXAC from Ubuntu
Universal DIS credentials used to access WEXAC from Ubuntu
Access WEXAC from MAC
Access WEXAC from MAC using universal DIS credentials
### WEXAC Modules

<table>
<thead>
<tr>
<th>Module Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>artemis/3.4.7</td>
<td>nepenthe/3.10.2</td>
</tr>
<tr>
<td>aspen/4.3.4</td>
<td>numpy/1.15.0</td>
</tr>
<tr>
<td>assymbiotics/2018Dec11</td>
<td>oscar/3.0.0</td>
</tr>
<tr>
<td>baxcomb/2</td>
<td>paper/2.0.0</td>
</tr>
<tr>
<td>blast/2.2.26</td>
<td>python/3.7.3</td>
</tr>
<tr>
<td>cglib/3.3.3</td>
<td>python-checker/3.7.3</td>
</tr>
<tr>
<td>cellranger/1.2.1</td>
<td>python-incremental/3.7.3</td>
</tr>
<tr>
<td>chimera/1.12.2</td>
<td>pytorch/3.7.3</td>
</tr>
<tr>
<td>circRNA/3.49-1</td>
<td>qsub/2.2.6</td>
</tr>
<tr>
<td>coda/7.5</td>
<td>qa/f/2.0.0</td>
</tr>
<tr>
<td>cuda/8.0.44</td>
<td>qa/test/2.0.0</td>
</tr>
<tr>
<td>cytoscape/3.4.0</td>
<td>qtpdf/2.2.7</td>
</tr>
<tr>
<td>eigen/3.3.0</td>
<td>qtpdf/api/2.2.7</td>
</tr>
<tr>
<td>fastqc/0.11.5</td>
<td>qtpdf/api2/2.2.7</td>
</tr>
<tr>
<td>tasi_q/screener/8.9.5</td>
<td>qtpdf/baselib/2.2.7</td>
</tr>
<tr>
<td>QAT/3.7</td>
<td>qtpdf/baselib2/2.2.7</td>
</tr>
</tbody>
</table>

Note: The above list is not exhaustive and only represents a selection of modules available in WEXAC.
[vitalyg@access ~]# module --help

Modules Release 3.2.10 2012-12-21 (Copyright GNU GPL v2 1991):

Usage: module [ switches ] [ subcommand ] [subcommand-args ]

Switches:
- H | --help                      this usage info
- v | --version                   modules version & configuration options
- f | --force                     force active dependency resolution
- t | --terse                     terse format avail and list format
- l | --long                      long format avail and list format
- h | --human                     readable format avail and list format
- v | --verbose                   enable verbose messages
- s | --silent                    disable verbose messages
- c | --create                    create caches for avail and apropos
- i | --icase                     case insensitive
-u | --userlvl <lvl>              set user level to (nov[ice],exp[ert],adv[anced])

Available SubCommands and Args:
  + add|load modulefile [modulefile ...]
  + rm|unload modulefile [modulefile ...]
  + switch|swap [modulefile1] modulefile2
  + display|show modulefile [modulefile ...]
  + avail modulefile [modulefile ...]
  + use [-a|--append] dir [dir ...]
  + unuse dir [dir ...]
  + update
  + refresh
  + purge
  + list
  + clear
  + help [modulefile [modulefile ...]]
  + whatis [modulefile [modulefile ...]]
  + apropos|keyword string
  + initadd modulefile [modulefile ...]
  + initprepend modulefile [modulefile ...]
  + initrm modulefile [modulefile ...]
  + initswitch modulefile1 modulefile2
  + initlist
  + initclear
[vitalyg@access ~]$ module list
Currently Loaded Modulefiles:
   1) matlab/R2016b  2) mathematica/11.0  3) R/3.3.2  4) python/2.7
[vitalyg@access ~]$ module load bowtie2/2.2.9
[vitalyg@access ~]$ module list
Currently Loaded Modulefiles:
   1) matlab/R2016b  3) R/3.3.2  5) bowtie2/2.2.9
   2) mathematica/11.0  4) python/2.7
[vitalyg@access ~]$ module load samtools/1.3.1
[vitalyg@access ~]$ module list
Currently Loaded Modulefiles:
   1) matlab/R2016b  3) R/3.3.2  5) bowtie2/2.2.9  7) samtools/1.3.1
   2) mathematica/11.0  4) python/2.7  6) icompiler/2017
WEXAC Job Submission & Control commands

bsub - submits a batch job to LSF
bjobs - displays information about LSF jobs
bhist - displays historical information about jobs
bkill - sends signals to kill, suspend, or resume unfinished jobs
bmod - modifies job submission options of a job
bpeek - displays the stdout and stderr output of an unfinished job
bstop - suspends unfinished jobs
bresume - resumes unfinished jobs
bswitch - switches jobs from one queue to another
Commonly Used Options

- **-q qname** submits the job to the specified queue
- **-o file** redirect stdout, stderr and resource usage information of the job to the specified *output file*
- **-e file** redirect stderr to the specified error file
- **-o/-e file** same as -o/-e, but overwrite file if it exists
- **-i filename** use the specified file as standard input for the job
- **-n number** specify number of job slots
- **-g jobgroup** submit job to specified group
- **-J jobname** assigns the specified name to the job
- **-R res_req** runs job on a host that meets the specified resource requirements
- **-L shell** Initializes the execution environment using the specified login shell
bsub – Methods for Submitting Jobs

• By script or command
  $ bsub -q new-all.q -J example -o example-%J.o -e example-%J.e date

• By job spooling
  $ bsub < job_file

job_file example:
#BSUB -q new-all.q
#BSUB -J example
#BSUB -o example-%J.o
#BSUB -e example-%J.e
date
bsub – Methods for Submitting Jobs

Interactively

$ bsub
bsub> #BSUB -q new-all.q
bsub> #BSUB -J example
bsub> #BSUB -o example-%J.o
bsub> #BSUB -e example-%J.e
bsub> date
bsub> ^D
Job <2387> is submitted to queue <new-all.q>.
Resource Requirements (-R)

Resource requirement string is divided into following sections:

- **Selection** - select[selection_string]
- **Usage** - rusage[rusage_string]
- **Ordering** - order[order_string]
- **Locality** - span[span_string]
- **Same** - same[same_string]
- **CU** - cu[cu_string]

Span and same sections are used for parallel jobs
Example 1. Select execution host candidates that have at least 2GB free RAM
$ bsub -R "select[mem>2GB]" myJob

Example 2. Select execution host candidates with Infiniband interconnect
$ bsub -R "defined(ib)" myJob

Example 3. Select candidate hosts that have 1000MB free swap and order by amount of available memory
$ bsub -R "select[swp>1000] order[mem]" myJob
Example 4. Candidate hosts should have min 500MB free RAM, job will reserve 400MB RAM.

$ bsub -n 4 -R "select[mem>500] rusage[mem=400]" myJob

Example 5. All slots required for a parallel job should reside on the same host

$ bsub -n 4 -R "span[hosts=1]" parallelJob

Example 6. 12-CPU parallel job should run on up to 4 CPUs per host, all candidate hosts must have the same CPU model

$ bsub -n 12 -R "span[ptile=4] same[model]" parallelJob
Array Jobs – example of usage

• A job array is submitted using the syntax:
  job_name_spec[index | start_index-end_index:step]

```
[vitaly@access mathematica]$ cat array_math.job
#BSUB -q new-short
#BSUB -J mathematica[1-10]
#BSUB -o results.%j
#BSUB -e results.%j
math -script fullCluster.m $[LSB_JOBINDEX]
[vitaly@access mathematica]$ bsub < array_math.job
Memory reservation is (MB): 2048
Memory Limit is (MB): 2048
Job <618114> is submitted to queue <new-short>.
[vitaly@access mathematica]$ bjobs
JOBID USER STAT QUEUE FROM_HOST EXEC_HOST JOB_NAME SUBMIT_TIME
[vitaly@access mathematica]$ `
Array Jobs

[vitalyg@access mathematica]$ cat array_math.job
#BSUB -q new-short
#BSUB -J mathematica[1-10:2]
#BSUB -o results.%J
#BSUB -e results.%J
math -script fullCluster.m ${LSB_JOBINDEX}
[vitalyg@access mathematica]$ bsub < array_math.job
Memory reservation is (MB): 2048
Memory Limit is (MB): 2048
Job <618115> is submitted to queue <new-short>.
[vitalyg@access mathematica]$ bjobs

<table>
<thead>
<tr>
<th>JOBID</th>
<th>USER</th>
<th>STAT</th>
<th>QUEUE</th>
<th>FROM_HOST</th>
<th>EXEC_HOST</th>
<th>JOB_NAME</th>
<th>SUBMIT_TIME</th>
</tr>
</thead>
</table>
-a Display information about jobs in all states, including recently finished jobs
-A Displays summarized information about job arrays
-d Display information about jobs that finished recently
-l| -w Display information in long or wide format
-p Display information about pending jobs
-r Display information about running jobs
-g job_group Display information about jobs in specified group
-J job_name Display information about specified job or array
-q queue Display information about jobs in specified queue
-u user Display information about jobs for specified users/groups
```
[vitalyg@access ~]$ bjobs

<table>
<thead>
<tr>
<th>JOBID</th>
<th>USER</th>
<th>STAT</th>
<th>QUEUE</th>
<th>FROM_HOST</th>
<th>EXEC_HOST</th>
<th>JOB_NAME</th>
<th>SUBMIT_TIME</th>
</tr>
</thead>
<tbody>
<tr>
<td>478504</td>
<td>vitalyg</td>
<td>RUN</td>
<td>short</td>
<td>access.wexa</td>
<td>cn165.wexac</td>
<td>example3</td>
<td>Jan 17 10:47</td>
</tr>
<tr>
<td>478502</td>
<td>vitalyg</td>
<td>RUN</td>
<td>new-all.q</td>
<td>access.wexa</td>
<td>cn165.wexac</td>
<td>example1</td>
<td>Jan 17 10:47</td>
</tr>
<tr>
<td>478503</td>
<td>vitalyg</td>
<td>USUSP</td>
<td>new-all.q</td>
<td>access.wexa</td>
<td>cn165.wexac</td>
<td>example2</td>
<td>Jan 17 10:47</td>
</tr>
<tr>
<td>478505</td>
<td>vitalyg</td>
<td>RUN</td>
<td>short</td>
<td>access.wexa</td>
<td>cn160.wexac</td>
<td>example4</td>
<td>Jan 17 10:48</td>
</tr>
<tr>
<td>478506</td>
<td>vitalyg</td>
<td>RUN</td>
<td>short</td>
<td>access.wexa</td>
<td>cn160.wexac</td>
<td>example5</td>
<td>Jan 17 10:48</td>
</tr>
<tr>
<td>478507</td>
<td>vitalyg</td>
<td>RUN</td>
<td>short</td>
<td>access.wexa</td>
<td>cn160.wexac</td>
<td>example6</td>
<td>Jan 17 10:48</td>
</tr>
<tr>
<td>478509</td>
<td>vitalyg</td>
<td>PEND</td>
<td>short</td>
<td>access.wexa</td>
<td></td>
<td>example7</td>
<td>Jan 17 10:50</td>
</tr>
</tbody>
</table>
```
### bjobs – Example 2: detailed job view

```
[vitalyg@access ~]$ bjobs -l 478509

Job <478509>, Job Name <example7>, User <vitalyg>, Project <default>, Status <PEND>, Queue <short>, Command <sleep 999>
Tue Jan 17 10:50:42: Submitted from host <access.wexac.weizmann.ac.il>, CWD <$HOME>, Output File </dev/null>, Re-runnable, Requested Resources <rusage[mem=2048]>, Specified Hosts <cn149>:

**MEMLIMIT**
2 G

**PENDING REASONS:**
There are no suitable hosts for the job;

**SCHEDULING PARAMETERS:**

<table>
<thead>
<tr>
<th>r15s</th>
<th>r1m</th>
<th>r15m</th>
<th>ut</th>
<th>pg</th>
<th>io</th>
<th>ls</th>
<th>it</th>
<th>tmp</th>
<th>swp</th>
<th>mem</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**loadSched**

- - - - - - - - -

**loadStop**

- - - - - - - - -

**ngpus**

- - - - - - - - -

**ngpus_shared**

- - - - - - - - -

**ngpus_excl_t**

- - - - - - - - -

**ngpus_excl_p**

- - - - - - - - -

**ngpus_prohibited**

- - - - - - - - -

**loadSched**

- - - - - - - - -

**loadStop**

- - - - - - - - -

**gpu_temp0**

- - - - - - - - -

**gpu_ecc0**

- - - - - - - - -

**gpu_temp1**

- - - - - - - - -

**gpu_ecc1**

- - - - - - - - -

**gpu_temp2**

- - - - - - - - -

**gpu_ecc2**

- - - - - - - - -

**nmics**

- - - - - - - - -

**loadSched**

- - - - - - - - -

**loadStop**

- - - - - - - - -

**mic_temp0**

- - - - - - - - -

**mic_temp1**

- - - - - - - - -

**mic_freq0**

- - - - - - - - -

**mic_freq1**

- - - - - - - - -

**mic_power0**

- - - - - - - - -

**mic_power1**

- - - - - - - - -

**loadSched**

- - - - - - - - -

**loadStop**

- - - - - - - - -

**mic_freeemem0**

- - - - - - - - -

**mic_freeemem1**

- - - - - - - - -

**mic_util0**

- - - - - - - - -

**mic_util1**

- - - - - - - - -

**mic_ncores0**

- - - - - - - - -

**mic_ncores1**

- - - - - - - - -

**loadSched**

- - - - - - - - -

**loadStop**

- - - - - - - - -

**RESOURCE REQUIREMENT DETAILS:**
Combined: select[(type = any ) && (type == any)] order[r15s:pg] rusage[mem=2048.00]
*Effective: -*
```
bhist – Jobs History options

-a Display information about all jobs
-b|-l|-w Display information in brief, long, or wide format
-d Display information about finished jobs
-p Display information about pending jobs
-s Display information about suspended jobs
-t Display job events chronologically
-C|-D|-S|-T start_time,end_time

Display information about completed, dispatched, submitted, or all jobs in specified time window

-q queue Display information about jobs submitted to specified queue
-u username|all Display information about jobs submitted by user or all users
### bhist – Example 1

```
[vitalyg@access ~]$ bhist
Summary of time in seconds spent in various states:

<table>
<thead>
<tr>
<th>JOBID</th>
<th>USER</th>
<th>JOB_NAME</th>
<th>PEND</th>
<th>PSUSP</th>
<th>RUN</th>
<th>USUSP</th>
<th>SSUSP</th>
<th>UNKWN</th>
<th>TOTAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>478502</td>
<td>vitalyg</td>
<td>example1</td>
<td>0</td>
<td>0</td>
<td>539</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>539</td>
</tr>
<tr>
<td>478503</td>
<td>vitalyg</td>
<td>example2</td>
<td>1</td>
<td>0</td>
<td>46</td>
<td>486</td>
<td>0</td>
<td>0</td>
<td>533</td>
</tr>
<tr>
<td>478504</td>
<td>vitalyg</td>
<td>example3</td>
<td>1</td>
<td>0</td>
<td>520</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>521</td>
</tr>
<tr>
<td>478505</td>
<td>vitalyg</td>
<td>example4</td>
<td>0</td>
<td>0</td>
<td>471</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>471</td>
</tr>
<tr>
<td>478506</td>
<td>vitalyg</td>
<td>example5</td>
<td>1</td>
<td>0</td>
<td>467</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>468</td>
</tr>
<tr>
<td>478507</td>
<td>vitalyg</td>
<td>example6</td>
<td>0</td>
<td>0</td>
<td>464</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>464</td>
</tr>
<tr>
<td>478509</td>
<td>vitalyg</td>
<td>example7</td>
<td>342</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>342</td>
</tr>
</tbody>
</table>
```

[vitalyg@access ~]$
bhist – Example 2 : detailed job information

[vitalyg@access ~]$ bhist -l 478502

Job <478502>, Job Name <example1>, User <vitalyg>, Project <default>, Command <sleep 999>
Tue Jan 17 10:47:25: Submitted from host <access.wexac.weizmann.ac.il>, to Queue <new-all.q>, CWD <$HOME>, Output File </dev/null>, Re-runnable, Requested Resources <rusage[mem=2048]>

MEMLIMIT
2 G
Tue Jan 17 10:47:25: Dispatched 1 Task(s) on Host(s) <cn165.wexac.weizmann.ac.il>, Allocated 1 Slot(s) on Host(s) <cn165.wexac.weizmann.ac.il>, Effective RES_REQ <select[[(type = any ) && (type = any)]] order[r15s:pg] rusage[mem=2048.00] >;
Tue Jan 17 10:47:25: Starting (Pid 13646);
Tue Jan 17 10:47:25: Running with execution home </home/labs/training/vitalyg>, Execution CWD </home/labs/training/vitalyg>, Execution Pid <13646>;

Summary of time in seconds spent in various states by Tue Jan 17 10:58:15
PEND PSUSP RUN USUSP SSUSP UNKNW TOTAL
0 0 650 0 0 0 650
Manipulating Jobs

**bkill** - send signals to kill, suspend, or resume unfinished jobs
  Tip: use JobID 0 to kill all your jobs

**bmod** - modify job submission options of a job

**bpeek** - display the stdout and stderr output of an unfinished job

**bstop** - suspend unfinished jobs

**bresume** - resume unfinished jobs

**bswitch** - switch jobs from one queue to another
Manipulating Jobs

[vitalyg@access ~]$ bsub -q new-all.q -oo output -eo error -J example sleep 999
Memory reservation is (MB): 2048
Memory Limit is (MB): 2048
Job <478532> is submitted to queue <new-all.q>.

[vitalyg@access ~]$ bjobs 478532
JOBID USER STAT QUEUE FROM_HOST EXEC_HOST JOB_NAME SUBMIT_TIME
478532 vitalyg RUN new-all.q access.wexa cn169.wexac example Jan 17 12:54

[vitalyg@access ~]$ bstop 478532
Job <478532> is being stopped

[vitalyg@access ~]$ bjobs 478532
JOBID USER STAT QUEUE FROM_HOST EXEC_HOST JOB_NAME SUBMIT_TIME
478532 vitalyg SUSP new-all.q access.wexa cn169.wexac example Jan 17 12:54

[vitalyg@access ~]$ bresume 478532
Job <478532> is being resumed

[vitalyg@access ~]$ bjobs 478532
JOBID USER STAT QUEUE FROM_HOST EXEC_HOST JOB_NAME SUBMIT_TIME
478532 vitalyg RUN new-all.q access.wexa cn169.wexac example Jan 17 12:54
Manipulating Jobs

Example 2

```
[vitalyg@access ~]$ bjobs
JOBID USER STAT QUEUE FROM_HOST EXEC_HOST JOB_NAME SUBMIT_TIME
478529 vitalyg RUN new-all.q access.wexa cn146.wexac example Jan 17 12:41
478530 vitalyg PEND new-all.q access.wexa example Jan 17 12:43

[vitalyg@access ~]$ bmod -q new-short 478530
Parameters of job <478530> are being changed

[vitalyg@access ~]$ bjobs
JOBID USER STAT QUEUE FROM_HOST EXEC_HOST JOB_NAME SUBMIT_TIME
478530 vitalyg RUN new-short access.wexa cn146.wexac example Jan 17 12:43
478529 vitalyg RUN new-all.q access.wexa cn146.wexac example Jan 17 12:41
```

Information Systems
GPU Jobs – Is the gain worth the pain?

```
[vadim.vadim-laptop] $ ssh root@cn151
Warning: Permanently added 'cn151' (RSA) to the list of known hosts.
X11 forwarding request failed on channel 0
Last login: Wed Feb 8 13:58:47 2017
[root@cn151 ~]# module load cuda
[root@cn151 ~]# nvidia-smi
Sun Feb 12 17:43:55 2017

+-----------------------------------------------------------------------------+
<table>
<thead>
<tr>
<th>NVIDIA-SMI 367.55</th>
<th>Driver Version: 367.55</th>
</tr>
</thead>
<tbody>
<tr>
<td>GPU  Name  Persistence-M</td>
<td>Bus-Id</td>
</tr>
<tr>
<td>Fan  Temp  Perf  Pwr:Usage/Cap</td>
<td>Memory-Usage</td>
</tr>
<tr>
<td>-------------------</td>
<td>--------------------------</td>
</tr>
</tbody>
</table>
+-------------------|--------------------------|
|   0  Tesla K40m   Off | 0000:09:00.0  Off | 0  |
| N/A 24C  P8 19W / 235W | 2MiB / 11439MiB | 0% Default |
+-------------------|--------------------------|

Processes:
```

```
GPU    PID  Type  Process name
-----------------------------

No running processes found
```
ScaleMP: Monster Machine

```
root@cn202:~# top - 09:11:31 up 489 days, 17:42, 1 user, load average: 9.45, 9.57, 9.55
Tasks: 310 total, 1 running, 308 sleeping, 0 stopped, 0 zombie
Cpu0 : 0.0%us, 0.3%sy, 99.7%ni, 0.0%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st
Cpu1 : 0.0%us, 0.0%sy, 100.0%ni, 0.0%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st
Cpu2 : 0.0%us, 0.0%sy, 100.0%ni, 0.0%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st
Cpu3 : 0.0%us, 0.0%sy, 100.0%ni, 0.0%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st
Cpu4 : 0.0%us, 0.0%sy, 100.0%ni, 0.0%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st
Cpu5 : 0.0%us, 0.0%sy, 100.0%ni, 0.0%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st
Cpu6 : 0.0%us, 0.0%sy, 100.0%ni, 0.0%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st
Cpu7 : 0.0%us, 0.0%sy, 100.0%ni, 0.0%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st
Mem: 16333312k total, 2971566k used, 13361756k free, 396040k buffers
Swap: 24575996k total, 13744k used, 24562252k free, 1592916k cached

PID USER      PR NI  VIRT  RES  SHR  S %CPU %MEM  TIME+ COMMAND
8490 asafmi   39  19  586m  76m  18m R 100.3  0.5  1962:07 python
```

```
root@cn133:~# top - 12:39:20 up 39 days, 18:33, 2 users, load average: 0.06, 0.06, 0.00
Tasks: 2495 total, 1 running, 2494 sleeping, 0 stopped, 0 zombie
Cpu(s):  0.0%us,  0.0%sy,  99.9%ni,  0.0%wa,  0.0%hi,  0.0%si,  0.0%st
Mem: 941339888k total, 68658140k used, 872681748k free, 470776k buffers
Swap:  0k total,  0k used,  0k free, 51987424k cached

Sorry, terminal is not big enough
```

```
PID USER      PR NI  VIRT  RES  SHR  S %CPU %MEM  TIME+ COMMAND
293982 root   20  0   14000 3000  808  R  5.3  0.0  054.99 top
142733 root   0 -20  25492 4788 1968 S  1.6  0.0  597:26.45 lim
```

```
1TB RAM

128 cores
```
Cluster Query Commands

lsinfo   - displays load sharing configuration information
lshosts  - displays hosts and their static resource information
lsload   - displays load information for hosts
bhosts   - displays hosts and their static and dynamic resources
bqueues  - displays information about queues
bmgroup  - displays information about host groups and compute units
busers   - displays information about users and user groups
bugroup  - displays information about user groups
Example 1. Show resource information for compute nodes cn141 and cn142

[vitalyg@access ~]$ lshosts cn141 cn142

Example 2. Show load information for compute nodes cn141 and cn142

[vitalyg@access ~]$ lsload cn141 cn142

Example 3. Show statistics for hosts cn141 and cn142

[vitalyg@access ~]$ bhosts cn141 cn142
Query Commands

Example 4. Show queues

```
[vitaly@access ~]$ bqueues
 QUEUE_NAME  PRI  STATUS  MAX J1/U J1/P J1/M NJOBS PEND  RUN  SUSP
 bio-nem     85   Open:Active  -   -   -   0   0   0   0
 sorek-high-prio 88   Open:Active  -   6   -   -   0   0   0   0
 bio         88   Open:Active  -   -   -   -   0   0   0   0
 schwartz    88   Open:Active  -   -   -   0   0   0   0
 feulf       88   Open:Active  -   -   -   1   1   0   0
 bio-sert    75   Open:Active  -   -   -   -   0   0   0   0
 bio-guest   78   Open:Active  -   -   -   0   0   0   0
 wicc-priority 65  Open:Active  -   -   -   0   0   0   0
 fleishman-prior 68  Open:Active  -   -   -   0   0   0   0
 hana-priority 68  Open:Active  -   -   -   0   0   0   0
 koren-priority 68  Open:Active  -   -   -   0   0   0   0
 lancet-priority 68  Open:Active  -   -   -   0   0   0   0
 schneidman-prio 68  Open:Active  -   -   -   0   0   0   0
 all-high    68  Open:Active  -   1000   -   -   0   0   0   0
 new-short   68  Open:Active  -   -   -   0   0   0   0
```

Example 5. Display information about all hosts

```
[vitaly@access ~]$ bhosts
 HOST_NAME          STATUS  J1/U  MAX NJOBS  RUN  SSUSP  USUSP  RSV
 cn088.wexac.welma ok   -   24   10  10  0   0   0   0
 cn081.wexac.welma ok   -   24   20  20  0   0   0   0
 cn083.wexac.welma unavall   -   32   0   0   0   0   0   0
 cn084.wexac.welma ok   -   32   30  30  0   0   0   0
 cn101.wexac.welma ok   -   24   20  20  0   0   0   0
 cn104.wexac.welma ok   -   24   20  20  0   0   0   0
 cn105.wexac.welma ok   -   24   20  20  0   0   0   0
 cn106.wexac.welma ok   -   24   20  20  0   0   0   0
 cn107.wexac.welma ok   -   24   0   0   0   0   0   0
 cn108.wexac.welma closed   -   24   20  20  0   0   0   0
```
RTM Operation

USERS TRAINING
Why is it important for LSF Cluster’s administrator:

- It’s an “Out of the BOX” and end to end cluster monitoring tool designed for Platform customers!
- It answers the question “how are my clusters doing right now”
- It brings together LSF data from multiple sources into a single interface
- It simplifies the debugging of LSF problems
- It allows Job and Host data to be visualized together
- It can be easily customized to increase penetration to other business needs
How to Manage a computes cluster
# RTM Host Group Info

[Image of the IBM RTM Host Group info interface]

- The interface is displaying the status and details of various host groups.
- It shows the status of hosts such as "All", "High", "Medium", and "Low".
- Each host group has metrics like AvgIP, AvgCPU, AvgIO, etc.
- The interface also includes filters for sorting and viewing the data.

### Host Group Details

| Host Group | Status | AvgIP | AvgCPU | AvgIO | AvgPage | AvgRate | AvgIOPer | AvgPagePer | AvgRatePer | AvgMore | AvgIP | AvgCPU | AvgIO | AvgPage | AvgRate | AvgIOPer | AvgPagePer | AvgRatePer | AvgMore |
|------------|--------|-------|--------|-------|---------|---------|----------|-----------|-----------|---------|-------|-------|-------|-------|---------|---------|-----------|-----------|---------|-------|
| all        | All    | 1.56  | 0.37   | 1.00  | 0.00    | 0.00    | 0.00     | 0.00      | 0.00      | 0.00    | 0.00  | 0.00  | 0.00  | 0.00  | 0.00    | 0.00    | 0.00      | 0.00      | 0.00    | 0.00  |
| high       | High   | 2.80  | 0.54   | 1.20  | 0.00    | 0.00    | 0.00     | 0.00      | 0.00      | 0.00    | 0.00  | 0.00  | 0.00  | 0.00  | 0.00    | 0.00    | 0.00      | 0.00      | 0.00    | 0.00  |
| medium     | Medium | 1.60  | 0.30   | 0.90  | 0.00    | 0.00    | 0.00     | 0.00      | 0.00      | 0.00    | 0.00  | 0.00  | 0.00  | 0.00  | 0.00    | 0.00    | 0.00      | 0.00      | 0.00    | 0.00  |
| low        | Low    | 0.80  | 0.15   | 0.50  | 0.00    | 0.00    | 0.00     | 0.00      | 0.00      | 0.00    | 0.00  | 0.00  | 0.00  | 0.00  | 0.00    | 0.00    | 0.00      | 0.00      | 0.00    | 0.00  |

### Additional Information

- The interface provides options for filtering and sorting the data.
- It also includes a search bar and a bottom navigation bar.
- The layout is user-friendly with clear labels and metrics.
The Cluster Compute Nodes Dashboard
### Host Load Information Dashboard

#### Cluster: **vcloud**
- **Status:** All
- **Type:** All
- **User:** All
- **Reason:** All

#### Actions:
- **Host Name:**
  - **Load/Batch:**
    - **CPU Busy:**
    - **Rev Qty:**
    - **Mem Usage:**
    - **Page Usage:**
    - **Page Ratio:**
    - **Max Slots:**
    - **Free Slots:**
    - **Reserve Slots:**

#### Filter Options:
- **Time:**
- **CPU Busy:**
- **Rev Qty:**
- **Mem Usage:**
- **Page Usage:**
- **Page Ratio:**
- **Max Slots:**
- **Free Slots:**
- **Reserve Slots:**
Please pay attention: User asks for 8GB RAM, but actual usage is below 1.5GB.
RTM Reports
WEXAC submission from GUI: PAC

• Link to PAC: https://pac.wexac.weizmann.ac.il:8443
• Logon using your WEXAC credentials
PAC dashboard
### Hosts

<table>
<thead>
<tr>
<th>Host Name</th>
<th>LSF Status</th>
<th>Host Resources</th>
<th>CPUs</th>
<th>Cores</th>
<th>Job State In Use</th>
<th>CPU Usage</th>
<th>Free Memory</th>
</tr>
</thead>
<tbody>
<tr>
<td>master2.wxaxs.weizmann.ac.il</td>
<td>Closed_Full</td>
<td>mg</td>
<td>2</td>
<td>20</td>
<td>0</td>
<td>0 %</td>
<td>18/256 MB</td>
</tr>
<tr>
<td>master1.wxaxs.weizmann.ac.il</td>
<td>Closed_Full</td>
<td>mg</td>
<td>2</td>
<td>20</td>
<td>0</td>
<td>1 %</td>
<td>19/157 MB</td>
</tr>
<tr>
<td>wz002.wxaxs.weizmann.ac.il</td>
<td>Closed_Full</td>
<td>mg</td>
<td>2</td>
<td>20</td>
<td>0</td>
<td>0 %</td>
<td>10/90 MB</td>
</tr>
<tr>
<td>wz003.wxaxs.weizmann.ac.il</td>
<td>Closed_Full</td>
<td>mg</td>
<td>2</td>
<td>20</td>
<td>0</td>
<td>1 %</td>
<td>12/99 MB</td>
</tr>
<tr>
<td>wz044.wxaxs.weizmann.ac.il</td>
<td>Closed_Full</td>
<td>mg</td>
<td>2</td>
<td>20</td>
<td>0</td>
<td>7 %</td>
<td>19/89 MB</td>
</tr>
<tr>
<td>wz051.wxaxs.weizmann.ac.il</td>
<td>Closed_Full</td>
<td>mg</td>
<td>2</td>
<td>20</td>
<td>0</td>
<td>6 %</td>
<td>19/89 MB</td>
</tr>
<tr>
<td>wz052.wxaxs.weizmann.ac.il</td>
<td>Closed_Full</td>
<td>mg</td>
<td>2</td>
<td>20</td>
<td>0</td>
<td>6 %</td>
<td>19/89 MB</td>
</tr>
<tr>
<td>wz053.wxaxs.weizmann.ac.il</td>
<td>Closed_Full</td>
<td>mg</td>
<td>2</td>
<td>20</td>
<td>0</td>
<td>6 %</td>
<td>19/89 MB</td>
</tr>
<tr>
<td>wz000.wxaxs.weizmann.ac.il</td>
<td>Closed_Full</td>
<td>mg</td>
<td>2</td>
<td>20</td>
<td>0</td>
<td>6 %</td>
<td>19/89 MB</td>
</tr>
<tr>
<td>wz045.wxaxs.weizmann.ac.il</td>
<td>Closed_Full</td>
<td>mg</td>
<td>2</td>
<td>20</td>
<td>0</td>
<td>6 %</td>
<td>19/89 MB</td>
</tr>
<tr>
<td>wz046.wxaxs.weizmann.ac.il</td>
<td>Closed_Full</td>
<td>mg</td>
<td>2</td>
<td>20</td>
<td>0</td>
<td>6 %</td>
<td>19/89 MB</td>
</tr>
<tr>
<td>wz047.wxaxs.weizmann.ac.il</td>
<td>Closed_Full</td>
<td>mg</td>
<td>2</td>
<td>20</td>
<td>0</td>
<td>6 %</td>
<td>19/89 MB</td>
</tr>
<tr>
<td>wz048.wxaxs.weizmann.ac.il</td>
<td>Closed_Full</td>
<td>mg</td>
<td>2</td>
<td>20</td>
<td>0</td>
<td>6 %</td>
<td>19/89 MB</td>
</tr>
<tr>
<td>wz049.wxaxs.weizmann.ac.il</td>
<td>Closed_Full</td>
<td>mg</td>
<td>2</td>
<td>20</td>
<td>0</td>
<td>6 %</td>
<td>19/89 MB</td>
</tr>
<tr>
<td>wz050.wxaxs.weizmann.ac.il</td>
<td>Closed_Full</td>
<td>mg</td>
<td>2</td>
<td>20</td>
<td>0</td>
<td>6 %</td>
<td>19/89 MB</td>
</tr>
<tr>
<td>wz051.wxaxs.weizmann.ac.il</td>
<td>Closed_Full</td>
<td>mg</td>
<td>2</td>
<td>20</td>
<td>0</td>
<td>6 %</td>
<td>19/89 MB</td>
</tr>
<tr>
<td>wz052.wxaxs.weizmann.ac.il</td>
<td>Closed_Full</td>
<td>mg</td>
<td>2</td>
<td>20</td>
<td>0</td>
<td>6 %</td>
<td>19/89 MB</td>
</tr>
<tr>
<td>wz053.wxaxs.weizmann.ac.il</td>
<td>Closed_Full</td>
<td>mg</td>
<td>2</td>
<td>20</td>
<td>0</td>
<td>6 %</td>
<td>19/89 MB</td>
</tr>
</tbody>
</table>

### Host: master2.wxaxs.weizmann.ac.il

- **Host Name**: master2.wxaxs.weizmann.ac.il
- **LSF Status**: Closed_Full
- **CPU**: 2
- **Memory**: 105728 / 195478 MB
- **Swap**: 19087 MB
- **Disk IO Rate**: 36.3 MB/s
- **8/2/8/64**: 5

### Footer

© Copyright IBM Corporation 2014
The Jobs view is where users would go to view, manage and monitor ALL their jobs. By default, only pending and running jobs will be displayed.

Users can use the Filter to change viewing options such as viewing jobs by all users and all job states.
Basic job management
Job submission

- Generic job submission
Job submission

• Specific application submission: User only sees the applications they are allowed to see.

• Out of the box application templates: “Built in” application templates provided out of the box.
- Custom application templates created by the Administrator/Owner are “Custom type”.
- Only published application templates are available to users.
Job submission

- Specific application submission: Example ABAQUS submission form:
Job submission

• Job submission using generic template
Job monitoring and control

IBM Platform Application Center 9.1

Jobs
- Submission Forms
  - ABAQUS
  - ANSYS
  - EXCEED Desktop
  - MATLAB
  - NASTRAN
  - STAR-CCM
  - Xmanager
- Generic

Jobs
- By State
- Job Data
- Remote Job Consoles

Job: MyJob (349)
- Job ID: 349
- Job Name: MyJob
- Job Status: Pending

Pending Reasons: New job is waiting for scheduling.

More Details
- Tail
- View
- Download
- Copy To
- Move To
- More Actions

File Details
- Location: /home/isadmin/MyJob_1390327141758/LOG
- File Name
- File Size
- File Type
- Date Modified

Started: -
Description: -

Options
Advanced jobs management
Each job submission using application template will have a minimum of the following fields:

• Job name
  • The job name is something that should be meaningful for the user. This job name is used to create the job directory where all of the job files are kept.

• Version
  • The version of the application.

• CPU’s
  • The number of CPUs needed to the job.

• Wall clock time
  • The wall clock time is an estimated run time that the job will run for.

• Queue
  • Which LSF queue the job is to be submitted to.

• Input file(s)
  • Any input files the job may need. This will accept single, multiple and zip files.
Job submission details (2 of 2)

• With the application submission forms, there are 3 sections by default.
Job submission details: File selection

- generic_1390152948091MRRg
- generic_1390152953020L.kml
- generic_1390152957487MNqop
- generic_1390152959656tw058
- generic_13901529650659cgCb
- generic_139015793268dlinW
- generic_1390158132208nsIry
- generic_13901581454577eJ27

Path: /home/lsfadmin
File of Types: All
Job status details

Jobs page
Once a job is submitted, users will see the following job information. This information will only contain information from the job that was just submitted.
Job control and management

- The Jobs page will have many different actions that users can perform on the jobs.
Job control and management

- Users can check a file and then use the View button to see file contents. This can be done while job is running or when complete.
Job filter

- User can query jobs finished in 14 days ago.
- Job filter end time by default: done in past hour.
- Job filter end time available options:
  - any time, past hour, past 24 hours, past 7 days, past 14 days, past 30 days.
Job list view preferences

- Users are able to use the Options drop down to customize their display.
WEXAC Analytics: PA (Platform Analytics)

- Link to Platform Analytics: [http://analytics.wexac.weizmann.ac.il/](http://analytics.wexac.weizmann.ac.il/)
- Contact us for username/password at [hpc@weizmann.ac.il](mailto:hpc@weizmann.ac.il)
Platform Analytics
Architecture Components:

**Analytics Data Loaders:**
- Fast, Efficient, Reliable
- Load from Remote Locations

**Analytics Server:**
- ETL’s the RAW Data into a useable RPT table structure
- purges old RAW and RPT data
- adds business data
- used to customize the data structure for reporting

**Business Intelligence (ROLAP):**
- **workbooks** interfaces directly with the database – No Cubes!
- roll up and aggregation is done in real time in the reporting layer
- data visualizations are delivered through the format of graphically rich multi-functional dashboards
- extremely flexible
- data can be drilled down to database transactional level
- customizations can be done quickly and easily in the reporting workbooks

**Analytics Database:**
- fast data loading of large amounts of data - fast ad hock query capabilities
- good data compression - high availability
- horizontally scalable - runs on commodity hardware
- low cost - practically zero maintenance
Pending job statistics workbooks
Hands-on examples
**Bowtie**

Prerequisites -

Set a work directory:

```
$ WORKDIR=/home/labs/training/Shared/Bowtie
```

- **NOTE:** Setting a work directory is only needed for this training.

Load the required module:

```
$ module load bowtie/2.3.3
```

Submit a job -

```
$ bsub -q new-short -R "rusage[mem=4096]" -o output.%J -e error.%J "bowtie2 -x $WORKDIR/H_sapiens_build36_1/build36.1 --end-to-end --sensitive -U $WORKDIR/Inputab -S output-ab.sam"
```
Submit using a job file:

```bash
$ bsub < $WORKDIR/job.sub
```

Job file example:

```
#BSUB -q new-short
#BSUB -R "rusage[mem=4096]"
#BSUB -o output.%J
#BSUB -e error.%J
Bowtie2 -x $WORKDIR/H_sapiens_build36_1/build36.1 --end-to-end -sensitive -U $WORKDIR/Inptab -S output-ab.sam
```

Avg. runtime: 8 minutes

---

**mpi-blast**

Prerequisites -

Load init.sh:

```
$ source /home/labs/training/Shared/mpiblast/init.sh
```

Load the required modules:

```
$ module load mpiblast blast
```
Submit a job -

```bash
```

 Avg. runtime: 2:30 hours

---

**RSEM**

Prerequisites -

Set a work directory:
```
$ WORKDIR=/home/labs/training/Shared/rsem
```

Load the required modules:
```
$ module load bowtie2 rsem perl
```

Submit using a job file -
```
$ bsub < $WORKDIR/job.sub
```
Job file example:

```bash
#BSUB -q new-short
#BSUB -n 2
#BSUB -R "rusage[mem=3000]"
#BSUB -o out.%J
rssem-calculate-expression --bowtie2 /home/labs/training/Shared/rsem/data/R1_ES-26556613.R1.fastq.gz
/home/labs/training/Shared/rsem/mm9/mm9.refseq_genes.11072015.rsem out-putRSEM_refseqbam_R1_ES-26556613-$(LSB_JOBINDEX)
```

Avg. runtime: 3 hours

---

**Rosetta**

Prerequisite –

Load init.sh:

```bash
$ source /home/labs/training/Shared/Rosetta/init.sh
```

Submit a job -

```bash
$ bsub -q new-short -R "rusage[mem=2048]" -o out.%J
"$WORKDIR/rosetta_scripts.graphics.linuxgccrelease -s $WORKDIR/in/4EY7_A_renum.pdb
-parser:protocol $WORKDIR/refinement/refine_4EY7_A.xml @$WORKDIR/flags/flags_classic -overwrite"
```

Avg. runtime: 2:40 hours
WEXAC Web Resources

http://www.weizmann.ac.il/hpc
https://rtm.wexac.weizmann.ac.il
https://pac.wexac.weizmann.ac.il:8443
https://analytics.wexac.weizmann.ac.il
https://insightiq.weizmann.ac.il
http://master-ops.wexac.weizmann.ac.il/ganglia/
https://www.facebook.com/HpcAtWeizmann/
Questions?

hpc@weizmann.ac.il
Thank you!

hpc@weizmann.ac.il