IBM Spectrum LSF
What’s New in LSF Service Pack 12

Bill.McMillan@uk.ibm.com
Principal Product Manager
Spectrum Computing

Mingliang Zu
Technical Account Manager
Spectrum Computing

Yun Gao
Technical Support Professional
Spectrum Computing

Yi Sun
Technical Account Manager
Spectrum Computing

Michael Spriggs
STSM Spectrum Computing
Spectrum Computing

© 2021 IBM Corporation
Upcoming Seminars
Details and Registration on www.ibm.biz/LSFCommunitySeminars

September 2, 2021
What's New in LSF Service Pack 12
Speaker: LSF Support Team

September 16, 2021
Best Practices for Upgrading your LSF Clusters
Speaker: Larry Adams, Expert Labs

September 30, 2021
What if....? Using the LSF Simulator to answer those hard questions
Speakers: John Welch & Renita Leung, LSF SME

October 14, 2021
Simplifying HPC – Just push the button! Tips & Tricks
Speaker: Gabor Samu, Product Management

October 25-28, 2021
IBM TechU: https://www.ibm.com/training/events
• Applied AI and HPC: Case Studies from IBM
• Deploying LSF with OCP: Tips and Tricks
• HPC Cloud Bursting on IBM Cloud: Tips and Tricks

November 11, 2021
Expediting PMR’s with the LSF Support Tools
Speaker: LSF Support Team

November 29, 2021
High Performance Computing - Health Check Services from Lab Services
Speakers(s): Larry Adams, Expert Labs
Agenda

- Security Enforcement in FP12
- Nvidia A100 MIG Support in FP12
- Resource Connector Enhancement for Google Cloud
- Quick Deployment of LSF on IBM Cloud
- Deprecated Features
- Q&A: You can ask questions at any time in the Q&A panel
Security Enforcement in FP12
Security Enforcement in FP12

• A Brief Review of Major Security Features Introduced Prior to FP12
• New Parameters
• New Command Option
• New Enforced Default Value to The Security Related Parameters
• Compatibility
A Brief Review of Major Security Features Introduced Prior to FP12

- eauth.cve added (site-specific default encryption key and time difference check) in FP2
  - User customized eauth encryption key (LSF_EAUTH_KEY)

- eauth.cve upgrading in FP7
  - New parameter LSF_EXT_SERVERDIR in lsf.conf
  - New parameter LSF_ENV_OVERRIDE in lsf.conf
  - New hostsetup script options (--ext-serverdir --eauth-key)

- eauth.cve replaces the default eauth in FP10
A Brief Review of Major Security Features Introduced Prior to FP12

• Root privilege disabled by default in FP10
  - LSF_ROOT_REX removed in lsf.conf
  - New parameter LSF_ROOT_USER in lsf.conf

• Authentication of query commands (LSF_AUTH_QUERY_COMMANDS) in FP11

• Secured authorization requests (LSF_STRICT_CHECKING=enhanced) in FP11
New Parameters

• LSF_EAUTH_OLDKEY (lsf.sudoers)
  Specifies the previous key that eauth used to encrypt and decrypt user authentication data after you specify a new eauth key

• LSF_EAUTH_OLDKEY_EXPIRY (lsf.sudoers)
  Specifies the expiry date for the previous eauth key (LSF_EAUTH_OLDKEY_EXPIRY parameter), after which the previous key no longer works and only the new LSF_EAUTH_KEY parameter works.
New Command Option

- `badmin` security view

<table>
<thead>
<tr>
<th>No</th>
<th>Category</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Authentication</td>
<td>Check <code>eauth version</code>, <code>LSF_EAUTH_KEY</code>, <code>LSF_EAUTH_OLDKEY_EXPIRY</code></td>
</tr>
<tr>
<td>2</td>
<td>Message Integrity</td>
<td>Check <code>LSF_STRIC_CHEKCING</code></td>
</tr>
<tr>
<td>3</td>
<td>Secure Startup</td>
<td>Check <code>LSF_STARTUP_USERS</code>, <code>set-user-id</code> of <code>bctrld</code>, <code>lsadmin</code>, <code>badmin</code></td>
</tr>
<tr>
<td>4</td>
<td>Secure Configuration</td>
<td>Check <code>/etc/lsf.conf</code></td>
</tr>
<tr>
<td>5</td>
<td>Disallow Root User</td>
<td>Check <code>LSF_ROOT_USER</code></td>
</tr>
<tr>
<td>6</td>
<td>Kerberos Impersonation</td>
<td>Check <code>LSF_KRB_IMPERSONATE</code></td>
</tr>
<tr>
<td>7</td>
<td>Interactive Preload Protection</td>
<td>Check <code>LSF_LD_SECURITY</code></td>
</tr>
<tr>
<td>8</td>
<td>Custom Plugin Location</td>
<td>Check <code>LSF_EXT_SERVERDIR</code></td>
</tr>
<tr>
<td>9</td>
<td>Query Authentication</td>
<td>Check <code>LSF_AUTH_QUERY_COMMANDS</code></td>
</tr>
</tbody>
</table>
New Command Option

- badmin security view

```
$ yxu@ib20b01-684: badmin security view -v
Authentication:  Medium
You are using an up-to-date version of eauth with an automatically generated encryption key.

Enhanced Message Integrity:  Enabled
The default value of LSF STRICT CHECKING = ENHANCED is set, ensuring that communications cannot be tampered with.

Secure Startup:  Disabled
It is strongly recommended to remove the setuid bit from lsadmin, badmin, bcrtld and eauth. Daemon startup/shutdown should be performed with "bcrtld" and setuid should only be set on bcrtld and eauth when LSF STARTUP_USERS is defined in lsf.sudoers.

Secure Configuration:  Enabled
Secure configuration is enabled and cluster configuration will always be read from /etc/lsf.conf and $LSF_ENVDIR will be ignored.

Disallow Root User:  Enabled
The root user is not allowed to perform actions as a valid user from the LSF command line.

Interactive Preload Protection:  Disabled
Preload libraries are allowed when running interactive workload. This may weaken security and is not recommended.

Custom Plugin Location:  Disabled
Your e*plugins (eauth/elim etc) are stored in $LSF_SERVERDIR. This means they could be accidentally overridden during upgrades.

Query Authentication:  Enabled
Authentication is enabled for LSF query commands.
```
## New Enforced Default Value to the Security Related Parameters

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Prior FP12</th>
<th>FP12</th>
</tr>
</thead>
<tbody>
<tr>
<td>LSF_STRICT_CHECKING</td>
<td>Not defined</td>
<td>ENHANCED</td>
</tr>
<tr>
<td>LSF_AUTH_QUERY_COMMANDS</td>
<td>N</td>
<td>Y</td>
</tr>
<tr>
<td>LSF_ENV_OVERRIDE</td>
<td>Y</td>
<td>N</td>
</tr>
</tbody>
</table>
# Compatibility

<table>
<thead>
<tr>
<th>LSF</th>
<th>[Prior FP2]</th>
<th>[FP2 ~ FP11]</th>
<th>FP12</th>
</tr>
</thead>
<tbody>
<tr>
<td>[10.1 GA ~ FP2)</td>
<td>✓</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>[FP2 ~ FP11]</td>
<td>✗</td>
<td>✓</td>
<td>✗</td>
</tr>
<tr>
<td>FP12</td>
<td>✗</td>
<td>Condition</td>
<td>✓</td>
</tr>
</tbody>
</table>

**Condition**

1. If LSF\_SIMPLE\_CHECKING is not defined in lsf.conf, "N" should be configured to resolve the compatibility issue.

2. If LSF\_AUTH\_QUERY\_COMMANDS is not defined in lsf.conf, "N" should be configured to resolve the compatibility issue.

3. If above 2 parameters are defined in lsf.conf, everything should be working fine.
Nvidia A100 MIG Support in FP12
Nvidia A100 MIG Support in FP12

• MIG Introduction
• LSF MIG Support Pre-request
• LSF New Parameters for MIG Support
• New command options for MIG Support
• Use Case of MIG Support
MIG introduction

- NVIDIA releases GPU A100 with Multi-Instance GPU (MIG) feature.
MIG introduction

- **GPU Instances (GI)**
  - Each GPU can be split to 7 GPU instances (GI) at most.
  - The name of GI indicates the GPU Slices number and GPU memory.

  "2g.10gb" means the GI have 2 GPU Slices and 10GB GPU memory.

<table>
<thead>
<tr>
<th>Profile Name</th>
<th>Fraction of Memory</th>
<th>Fraction of SMs</th>
<th>Number of Instances Available</th>
</tr>
</thead>
<tbody>
<tr>
<td>MIG 1g.5gb</td>
<td>1/8</td>
<td>1/7</td>
<td>7</td>
</tr>
<tr>
<td>MIG 2g.10gb</td>
<td>2/8</td>
<td>2/7</td>
<td>3</td>
</tr>
<tr>
<td>MIG 3g.20gb</td>
<td>3/8</td>
<td>3/7</td>
<td>2</td>
</tr>
<tr>
<td>MIG 4g.20gb</td>
<td>4/8</td>
<td>4/7</td>
<td>1</td>
</tr>
<tr>
<td>MIG 7g.40gb</td>
<td>Full</td>
<td>7/7</td>
<td>1</td>
</tr>
</tbody>
</table>

© 2021 IBM Corporation
**MIG introduction**

- **Compute Instances (CI)**
  - Each GPU instance (GI) can be divided to several GPU Compute Instances (CI)

<table>
<thead>
<tr>
<th>Memory</th>
<th>20gb</th>
<th>20gb</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>GPU Instance</strong></td>
<td>3g</td>
<td>3g</td>
</tr>
<tr>
<td><strong>Compute Instance</strong></td>
<td>1c</td>
<td>1c</td>
</tr>
<tr>
<td><strong>MIG Device</strong></td>
<td>1c.3g.20gb</td>
<td>1c.3g.20gb</td>
</tr>
<tr>
<td></td>
<td>GPC</td>
<td>GPC</td>
</tr>
</tbody>
</table>

*The above shows that two different “3g.20gb” GPU Instances (GI) can be divided to 3 Compute Instances (CI) named 1c/1c/1c and 2 Compute Instances (CI) named 2c/1c respectively.*
MIG introduction

- **MIG Device**
  - A CI and its parent GI can be called a MIG device. And CUDA applications treat the MIG device as a normal GPU.

For example,
- A MIG device called 3g.20gb means the device have 3 GPU slices and 20 GB GPU memory, all compute instance are used.

- A MIG device called 2c.3g.20gb means the device have 3 GPU SM slices and 20 GB GPU memory, and only 2 compute instances(GPU SM slices) of total 3 are used.
LSF MIG Support Pre-request

- Enable Nvidia MIG mode
  
  ```
  $ sudo nvidia-smi -i 0 -mig 1
  Enabled MIG Mode for GPU 00000000:36:00.0
  All done.
  ```

- LSF GPU related parameters setting
  
  - LSF_GPU_RESOURCE_IGNORE=Y
  - LSF_GPU_AUTOCONFIG=Y
  - LSB_GPU_NEW_SYNTAX=extend
  - LSF_MANAGE_MIG=Y/N
LSF New Parameters for MIG Support

- **LSF_MANAGE_MIG=N** (Default Value)

The GPU instance (GI) and Compute instance (CI) is configured by the administrator on each host. LSF dispatches the job to the host based on its GI/CI topology.

For example:
The administrator creates GI/CI with size 1c.3g.20gb, 2c.3g.20gb, and 3g.20gb on host1, and the user submit a job with ‘-gpu num=1:mig=3/2’ request. LSF searches the available MIG resource among the hosts in the cluster, allocate the MIG resource and dispatch the job to host1, run the job on the MIG device 2c.3g.20gb
LSF New Parameters for MIG Support

- LSF_MANAGE_MIG=Y

LSF dynamically manages the GI/CI allocation/deallocation for the GPU hosts in the cluster. LSF will create GI/CI according to the job’s requirement and destroy GI/CI after job execution finishes.

For example:
The administrator doesn’t need to configure any GI/CI on GPU hosts in advance. If the user requests a job ‘mig=3/2’, LSF allocates a host that MIG 3/2 can be created, dispatch the job to that host, create GI/CI for this job, run job on the MIG device, finally destroy the MIG device.
New command option for MIG Support

- **lshosts –mig**
  - Use ‘`lshosts –gpu –mig`’ to check the MIG instances on each GPU hosts.
  - ‘`lshosts –gpu`’ indicate which gpu enabled MIG mode on each hosts

```bash
classic@rl-dgxa-c18:~:$ lshosts -gpu -mig
```

<table>
<thead>
<tr>
<th>HOST_NAME</th>
<th>gpu_id</th>
<th>gpu_model</th>
<th>gpu_driver</th>
<th>gpu_factor</th>
<th>numa_id</th>
<th>vendor</th>
<th>devid</th>
<th>gid</th>
<th>cid</th>
<th>inst_name</th>
</tr>
</thead>
<tbody>
<tr>
<td>rl-dgxa-c18</td>
<td>450.51.06</td>
<td>TeslaA100_SXM4_</td>
<td>8.0</td>
<td>3</td>
<td>Nvidia</td>
<td>0</td>
<td>11</td>
<td>0</td>
<td>1g.5gb</td>
<td></td>
</tr>
<tr>
<td></td>
<td>450.51.06</td>
<td>TeslaA100_SXM4_</td>
<td>8.0</td>
<td>3</td>
<td>Nvidia</td>
<td>1</td>
<td>13</td>
<td>0</td>
<td>1g.5gb</td>
<td></td>
</tr>
<tr>
<td></td>
<td>450.51.06</td>
<td>TeslaA100_SXM4_</td>
<td>8.0</td>
<td>3</td>
<td>Nvidia</td>
<td>0</td>
<td>3</td>
<td>0</td>
<td>2g.10gb</td>
<td></td>
</tr>
<tr>
<td></td>
<td>450.51.06</td>
<td>TeslaA100_SXM4_</td>
<td>8.0</td>
<td>1</td>
<td>Nvidia</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>3g.20gb</td>
<td></td>
</tr>
<tr>
<td></td>
<td>450.51.06</td>
<td>TeslaA100_SXM4_</td>
<td>8.0</td>
<td>1</td>
<td>Nvidia</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>4g.20gb</td>
<td></td>
</tr>
<tr>
<td></td>
<td>450.51.06</td>
<td>TeslaA100_SXM4_</td>
<td>8.0</td>
<td>1</td>
<td>Nvidia</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>7g.40gb</td>
<td></td>
</tr>
<tr>
<td></td>
<td>450.51.06</td>
<td>TeslaA100_SXM4_</td>
<td>8.0</td>
<td>7</td>
<td>Nvidia</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td></td>
<td>450.51.06</td>
<td>TeslaA100_SXM4_</td>
<td>8.0</td>
<td>5</td>
<td>Nvidia</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td></td>
<td>450.51.06</td>
<td>TeslaA100_SXM4_</td>
<td>8.0</td>
<td>5</td>
<td>Nvidia</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td></td>
</tr>
</tbody>
</table>

```bash
classic@rl-dgxa-c18:~:$ lshosts -gpu -factor
```

<table>
<thead>
<tr>
<th>HOST_NAME</th>
<th>gpu_id</th>
<th>gpu_model</th>
<th>gpu_driver</th>
<th>gpu_factor</th>
<th>numa_id</th>
<th>vendor</th>
<th>mig</th>
</tr>
</thead>
<tbody>
<tr>
<td>rl-dgxa-c18</td>
<td>450.51.06</td>
<td>TeslaA100_SXM4_</td>
<td>8.0</td>
<td>3</td>
<td>Nvidia</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td></td>
<td>450.51.06</td>
<td>TeslaA100_SXM4_</td>
<td>8.0</td>
<td>3</td>
<td>Nvidia</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td></td>
<td>450.51.06</td>
<td>TeslaA100_SXM4_</td>
<td>8.0</td>
<td>1</td>
<td>Nvidia</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td></td>
<td>450.51.06</td>
<td>TeslaA100_SXM4_</td>
<td>8.0</td>
<td>1</td>
<td>Nvidia</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td></td>
<td>450.51.06</td>
<td>TeslaA100_SXM4_</td>
<td>8.0</td>
<td>7</td>
<td>Nvidia</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td></td>
<td>450.51.06</td>
<td>TeslaA100_SXM4_</td>
<td>8.0</td>
<td>5</td>
<td>Nvidia</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td></td>
<td>450.51.06</td>
<td>TeslaA100_SXM4_</td>
<td>8.0</td>
<td>5</td>
<td>Nvidia</td>
<td>Y</td>
<td></td>
</tr>
</tbody>
</table>
Use Case of MIG Support

- Use ‘mig=gi_size/ci_size’ in job submission to request a MIG device
- Use ‘bjobs -g’ shows the GPU_ALLOCATION with MIG info

```
rladmin@rl-dgxa-c18-u19:~/migtest/lsftop/conf$ bsub -gpu num=1:mig=1/1 ...
Job <416> is submitted to default queue <normal>
rladmin@rl-dgxa-c18-u19:~/migtest/lsftop/conf$ bjobs
JOBID USER STAT QUEUE FROM_HOST EXEC_HOST JOB_NAME SUBMIT_TIME
416 rladmin RUN normal rl-dgxa-c18 rl-dgxa-c18 w/n/e06-gpu Nov 25 06:19
rladmin@rl-dgxa-c18-u19:~/migtest/lsftop/conf$ bjobs -l -g
Job <416>, User <rladmin>, Project <default>, Status <RUN>, Queue <normal>, Command <./mig_bin/e06-gpu>, Share group charged </rladmin>
Wed Nov 25 06:19:01: Submitted from host <rl-dgxa-c18-u19-epn266s0>, CWD </HOME/migtest/lsftop/conf>, Requested GPU <num=1:mig=1/1/1>
Wed Nov 25 06:19:01: Started 1 Task(s) on Host(s) <rl-dgxa-c18-u19-epn266s0>, Allocated 1 Slot(s) on Host(s) <rl-dgxa-c18-u19-epn266s0>, Execution Home </home/rladmin>, Execution CWD </home/rladmin/migtest/lsftop/conf/1>

SCHEDULING PARAMETERS:

<table>
<thead>
<tr>
<th>loadSched</th>
<th>loadStop</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

RESOURCE REQUIREMENT DETAILS:

Combined: select{ngpus=0} && (type == local) order[r15s:pg] rusage[ngpus physical=1.00:mig=1/1]
Effective: select{((ngpus=0))} && (type == local) order[r15s:pg] rusage[ngpus physical=1.00:mig=1/1]

GPU REQUIREMENT DETAILS:

Combined: num=1:mode=shared:mps=no:exclusive=yes:gvendor=nvidia:mig=1/1
Effective: num=1:mode=shared:mps=no:exclusive=yes:gvendor=nvidia:mig=1/1

GPU_ALLOCATION:

<table>
<thead>
<tr>
<th>HOST</th>
<th>TASK</th>
<th>GPU_ID</th>
<th>GI_ID/SIZE</th>
<th>CI_ID/SIZE</th>
<th>MODEL</th>
<th>MTOTAL</th>
<th>FACTOR</th>
<th>MRSV</th>
<th>SOCKET</th>
<th>NVLINK/XGMI</th>
</tr>
</thead>
<tbody>
<tr>
<td>rl-dgxa-c18-u19</td>
<td>0</td>
<td>0</td>
<td>4/1</td>
<td>4/1</td>
<td>TeslaA100_SX</td>
<td>39.5G</td>
<td>8.0</td>
<td>0M</td>
<td>3</td>
<td>-</td>
</tr>
</tbody>
</table>
Use Case of MIG Support

- Use ‘bhosts -gpu -l’ to show the MIG info on each hosts.

```
radmin@rl-dgxa-c18-u19:~/migtest/lsftop/conf$ bhosts -gpu -l
```

HOST: rl-dgxa-c18-u19-emp226s0
NGPUS NGPUS_SHARED_AVAIL NGPUS_EXCLUSIVE_AVAIL
8 8 8

<table>
<thead>
<tr>
<th>GPU ID</th>
<th>MODEL</th>
<th>MTOTAL</th>
<th>FACTOR</th>
<th>SOCKET</th>
<th>VENDOR</th>
<th>MIG</th>
<th>NVLINK/XGMI</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>TeslaA100_SXM4_40GB</td>
<td>39.5G</td>
<td>8.0</td>
<td>3</td>
<td>Nvidia</td>
<td>Y</td>
<td>/N/N/N/N/N/N/N/N</td>
</tr>
<tr>
<td>1</td>
<td>TeslaA100_SXM4_40GB</td>
<td>39.5G</td>
<td>8.0</td>
<td>3</td>
<td>Nvidia</td>
<td>Y</td>
<td>/N/N/N/N/N/N/N/N</td>
</tr>
<tr>
<td>2</td>
<td>TeslaA100_SXM4_40GB</td>
<td>39.5G</td>
<td>8.0</td>
<td>1</td>
<td>Nvidia</td>
<td>Y</td>
<td>/N/N/N/N/N/N/N/N</td>
</tr>
<tr>
<td>3</td>
<td>TeslaA100_SXM4_40GB</td>
<td>39.5G</td>
<td>8.0</td>
<td>1</td>
<td>Nvidia</td>
<td>Y</td>
<td>/N/N/N/N/N/N/N/N</td>
</tr>
<tr>
<td>4</td>
<td>TeslaA100_SXM4_40GB</td>
<td>39.5G</td>
<td>8.0</td>
<td>7</td>
<td>Nvidia</td>
<td>Y</td>
<td>/N/N/N/N/N/N/N/N</td>
</tr>
<tr>
<td>5</td>
<td>TeslaA100_SXM4_40GB</td>
<td>39.5G</td>
<td>8.0</td>
<td>7</td>
<td>Nvidia</td>
<td>Y</td>
<td>/N/N/N/N/N/N/N/N</td>
</tr>
<tr>
<td>6</td>
<td>TeslaA100_SXM4_40GB</td>
<td>39.5G</td>
<td>8.0</td>
<td>5</td>
<td>Nvidia</td>
<td>Y</td>
<td>/N/N/N/N/N/N/N/N</td>
</tr>
<tr>
<td>7</td>
<td>TeslaA100_SXM4_40GB</td>
<td>39.5G</td>
<td>8.0</td>
<td>5</td>
<td>Nvidia</td>
<td>Y</td>
<td>/N/N/N/N/N/N/N/N</td>
</tr>
</tbody>
</table>

DYNAMIC ATTRIBUTES

<table>
<thead>
<tr>
<th>GPU ID</th>
<th>MODE</th>
<th>MUSED</th>
<th>MRSV</th>
<th>TEMP</th>
<th>ECC</th>
<th>UT</th>
<th>MUT</th>
<th>PSTATE</th>
<th>STATUS</th>
<th>ERROR</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>SHARED</td>
<td>87M</td>
<td>0M</td>
<td>31C</td>
<td>0</td>
<td>0%</td>
<td>0%</td>
<td>0</td>
<td>ok</td>
<td>-</td>
</tr>
<tr>
<td>1</td>
<td>SHARED</td>
<td>7M</td>
<td>0M</td>
<td>27C</td>
<td>0</td>
<td>0%</td>
<td>0%</td>
<td>0</td>
<td>ok</td>
<td>-</td>
</tr>
<tr>
<td>2</td>
<td>SHARED</td>
<td>11M</td>
<td>0M</td>
<td>28C</td>
<td>0</td>
<td>0%</td>
<td>0%</td>
<td>0</td>
<td>ok</td>
<td>-</td>
</tr>
<tr>
<td>3</td>
<td>SHARED</td>
<td>14M</td>
<td>0M</td>
<td>28C</td>
<td>0</td>
<td>0%</td>
<td>0%</td>
<td>0</td>
<td>ok</td>
<td>-</td>
</tr>
<tr>
<td>4</td>
<td>SHARED</td>
<td>0M</td>
<td>0M</td>
<td>31C</td>
<td>0</td>
<td>0%</td>
<td>0%</td>
<td>0</td>
<td>ok</td>
<td>-</td>
</tr>
<tr>
<td>5</td>
<td>SHARED</td>
<td>0M</td>
<td>0M</td>
<td>30C</td>
<td>0</td>
<td>0%</td>
<td>0%</td>
<td>0</td>
<td>ok</td>
<td>-</td>
</tr>
<tr>
<td>6</td>
<td>SHARED</td>
<td>0M</td>
<td>0M</td>
<td>30C</td>
<td>0</td>
<td>0%</td>
<td>0%</td>
<td>0</td>
<td>ok</td>
<td>-</td>
</tr>
<tr>
<td>7</td>
<td>SHARED</td>
<td>0M</td>
<td>0M</td>
<td>30C</td>
<td>0</td>
<td>0%</td>
<td>0%</td>
<td>0</td>
<td>ok</td>
<td>-</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>GPU JOB INFORMATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>GPU_ID</td>
</tr>
<tr>
<td>--------</td>
</tr>
<tr>
<td>0</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td>4</td>
</tr>
<tr>
<td>5</td>
</tr>
<tr>
<td>6</td>
</tr>
<tr>
<td>7</td>
</tr>
</tbody>
</table>

© 2021 IBM Corporation
Use Case of MIG Support

- Request MIG device with GPU memory.
  - "bsub –gpu num=1:mig=3/2:gmem=5G gpu_app"
    
    It will request 1 MIG device with GI size 5 CI size 2 and 5G memory on this GPU Instance(GI).

- Request share/exclusive MIG device.
  - "bsub –gpu num=1:mig=2/1:mode=shared(or mode=exclusive_process or j_exclusive=yes) gpu_app"
    
    Although in MIG mode, only shared mode are supported, LSF has been enhanced to use MIG device exclusively or shared.
Resource Connector Enhancement for Google Cloud
Launch Instance Template

- Instance template is created on Google Cloud Platform portal

- Use **launchTemplateId** in googleprov_templates.json to enable support of launch instance template

- Properties of the instance can be defined within the instance template

- Properties defined in googleprov_templates.json override instance template

```
"templateId": "gcloud-VM-1",
"zone": "us-east1-d",
"imageId": "lsfdynamicvm",
"vmType": "f1-micro",
"privateNetworkOnlyFlag": false,
"vpc": "lsf-vpc",
"subnetId": "lsf-vpc-us-east1",
```

```
launchTemplateId defined?  

Y
Launch VM according to attributes defined in template

N
Launch VM according to user specified attribute
```
Local SSDs

- Local SSDs are supported through instance template, not in googleprov_templates.json

- In Google Cloud, attach local SSDs to instance template in **Disks ➔ Add new disk**

- SSDs must be mounted before being used. LSF provides an example of how to mount them in **user_data**: `<LSF_TOP>/<LSF_VERSION>/resource_connector/google/scripts/example_user_data.sh`
Preemptable VM instances

- Preemptable VM instances are supported through instance template, not in `googleprov_templates.json`

- In Google Cloud, set the **Preemptibility** field to **On** when creating the launch instance template to enable preemptible VM instances.

- When an instance is preempted, LSF automatically requeue the jobs running on it by default.
  - Using `RC_RECLAIM_ACTION=REQUEUE | TERMINATE` to change the default behavior
Bulk instance APIs (Bulk API endpoints)

- With bulk API, LSF resource connector creates VM instances and check status in batch
- Specify zone to let LSF call **zonal bulk API**. Instances launch in the specified zone
- Leave zone undefined to let LSF call **regional bulk API**. Zone of the instance is selected by Google Compute Engine based on hardware availability.
- Use **G CLOUD_REGION** in googleprov_config.json to define the region. Can be override by region in googleprov_template.json
Quick Deployment of LSF on IBM Cloud
Quick Deployment of LSF on IBM Cloud

- Deploy LSF with Resource Connector on IBM Cloud with a few clicks
- Automated deployment for compute, network and storage resources
- Easy to auto-scale up and down based on demand
- Allows seamless bursting to IBM Cloud from on-premise clusters with LSF Multi-Cluster
- Transfer data between clusters intelligently with LSF Data Manager
Quick Deployment of LSF on IBM Cloud
Deprecation LSF Features in FP12
## Deprecated LSF Features in FP12

<table>
<thead>
<tr>
<th>Deprecated feature</th>
<th>Corresponding deprecated parameters and commands</th>
<th>Alternative feature</th>
</tr>
</thead>
</table>
| LSF multicluster capability resource leasing model | `lsb.resources` file:  
  - `HostExport` section  
  - `SharedResourceExport` section  
  `lsb.queues` file:  
  - `HOSTS`: allremote and all@`cluster_name` keywords | Updated LSF default behavior. |
| LSF/XL feature and LSF Advanced Edition. | `-cname` option for the following commands: `bacct`, `bhosts`, `bjobs`, `bmggroup`, `lhosts`, `lsload` | Updated LSF default behavior. |
# Deprecated LSF Features in FP12

<table>
<thead>
<tr>
<th>Deprecated feature</th>
<th>Corresponding deprecated parameters and commands</th>
<th>Alternative feature</th>
</tr>
</thead>
<tbody>
<tr>
<td>Job slot pools for fairshare scheduling</td>
<td>• lsb.queues file:</td>
<td>Guarantee SLA (guarantee service class).</td>
</tr>
<tr>
<td></td>
<td>• MAX_SLOTS_IN_POOL</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• SLOT_POOL</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• SLOT_SHARE</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• USE_PRIORITY_IN_POOL</td>
<td></td>
</tr>
<tr>
<td>toplib integrations</td>
<td>• lsf.conf file:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• LSF_TOPD_PORT</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• LSF_TOPD_TIMEOUT</td>
<td></td>
</tr>
<tr>
<td>Chunk job scheduling</td>
<td>• lsf.conf file:</td>
<td>Use the RELAX_JOB_DISPATCH_ORDER parameter in the lsb.params file to enable multiple jobs with common resource requirements to run consecutively on the same allocation.</td>
</tr>
<tr>
<td></td>
<td>• LSB CHUNK RUSAGE</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• lsb.params file:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• CHUNK_JOB_DURATION</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• lsb.applications and lsb.queues files:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• CHUNK_JOB_SIZE</td>
<td></td>
</tr>
</tbody>
</table>
### Deprecated LSF Features in FP12

<table>
<thead>
<tr>
<th>Deprecated feature</th>
<th>Corresponding deprecated parameters and commands</th>
<th>Alternative feature</th>
</tr>
</thead>
<tbody>
<tr>
<td>HPC integrations for the following environments:</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
| - Parallel job support using PAM | - lsf.conf file:  
  - LSF_PAM_APPL_CHKPTN  
  - LSF_PAM_CLEAN_JOB_DELAY  
  - LSF_PAM_HOSTLIST_USE  
  - LSF_PAM_PLUGINDIR  
  - LSF_PAM_USE_ASH  
  - LSF_PE_NETWORK_NUM  
  - LSF_PE_NETWORK_UPDATE_INTERVAL  
  - LSB_CPUSET_BESTCPUS  
  - LSB_CPUSET_DISPLAY_CPULIST  
  - LSF_CPUSETLIB | |
| - IBM Parallel Environment (IBM PE) (See Running MPI workload through IBM Parallel Environment Runtime Edition) | - lsb.params file:  
  - MAX_PROTOCOL_INSTANCES  
  - NETWORK_REQ  
  - STRIPING_WITH_MINIMUM_NETWORK | Use the blaunch command for parallel jobs. |
| - SGI CPUSET | - lsb.queues file:  
  - MAX_PROTOCOL_INSTANCES  
  - NETWORK_REQ  
  - STRIPING_WITH_MINIMUM_NETWORK | |
| | - bsub -network command option | |

© 2021 IBM Corporation
## Deprecated LSF Features in FP12

<table>
<thead>
<tr>
<th>Deprecated feature</th>
<th>Corresponding deprecated parameters and commands</th>
<th>Alternative feature</th>
</tr>
</thead>
</table>
| Automatic CPU frequency selection in energy aware scheduling (See Automatic CPU frequency selection) | • lsf.conf file:  
  - LSF\_STRICT\_RESREQ: Now fixed to Y.                                                                 | Updated LSF default behavior (strict syntax for resource requirement selection strings). |
| Relaxed syntax for resource requirement selection strings                           |                                                                                                              |                                                                                  |
| Task list files and related commands                                               | • lsf.task file  
• lsf.conf file:  
  - LSF\_SHELL\_AT\_USERS  
• ch command  
• lsfmon command  
• lslogin command  
• lsmmon command  
• lstcsh command  
• lsltasks command  
• lsrtasks command  
• lseligible command |                                                                                  |
<table>
<thead>
<tr>
<th>Feature to be deprecated in the next LSF release</th>
<th>Corresponding deprecated parameters and commands</th>
<th>Alternative feature</th>
</tr>
</thead>
</table>
| SLA scheduling except guarantee SLA (See Using goal-oriented SLA scheduling). | - lsb.params file:  
  - ENABLE_DEFAULT_EGO_SLA  
  - lsb.serviceclasses file:  
  - GOALS: THROUGHPUT, VELOCITY, and DEADLINE keywords. | Guarantee SLA (guarantee service class), which will be the default behavior. |
| Slot and package guarantee policy | - lsb.params file:  
  - SIMPLIFIED_GUARANTEES: Will be fixed to Y. | |
| GPU scheduling and features that use ELIM | - lsf.conf file:  
  - LSB_GPU_NEW_SYNTAX: Will be fixed to extend.  
  - LSF_GPU_RESOURCE_IGNORE: Will be fixed to Y.  
  - LSF_GPU_AUTOCONFIG: Will be fixed to Y.  
  - All elim.gpu.* ELIMS: | Updated LSF default behavior (GPU autoconfiguration). |
References

Release Notes of FP12 -

Download Link of FP12 -
http://www.ibm.com/support/fixcentral/swg/selectFixes?product=ibm/Other+software/IBM+Spectrum+LSF&release=All&platform=All&function=fixId&fixids=lsf-10.1.0.12-spk-2021-Jun-build600488&includeSupersedes=0

Detailed Deprecated Features in FP12 -

LSF Deployment on IBM Cloud -

Nvidia A100 MIG user guide -
Thank you

Yun Gao
LSF Support Professional

bjyungao@cn.ibm.com