

# IsfMonitor 用户手册

Product Name : IsfMonitor

Product Version : V2.0

Release Date : 2026.02.08

Contact : [@李艳青](#) (liyanqing1987@163.com)

# 目录

一、简介 .....	4
1.1 功能简介 .....	4
1.2 快速了解 .....	5
二、环境依赖 .....	10
2.1 操作系统依赖 .....	10
2.2 PYTHON 版本依赖 .....	10
2.3 集群管理工具 .....	10
三、工具安装及配置 .....	11
3.1 工具下载 .....	11
3.2 工具安装 .....	12
3.3 工具配置 .....	15
四、工具使用 .....	17
4.1 数据采集 BSAMPLE .....	17
4.1.1 帮助信息 .....	17
4.1.2 手工采样 .....	18
4.1.3 自动采样 (定时) .....	18
4.1.4 数据库 .....	19
4.2 数据展示 BMONITOR .....	21
4.2.1 工具载入 .....	21
4.2.2 帮助信息 .....	23
4.2.3 菜单栏 .....	24
4.2.4 JOB 页 .....	25
4.2.5 JOBS 页 .....	26
4.2.6 HOSTS 页 .....	27
4.2.7 LOAD 页 .....	28
4.2.8 USERS 页 .....	28
4.2.9 QUEUES 页 .....	29
4.2.10 UTILIZATION 页 .....	30
4.2.11 LICENSE 页 .....	30
4.2.12 其它功能介绍 .....	31
五、辅助工具 .....	36
5.1 AKILL .....	36
5.2 PATCH .....	37
5.3 SEEDB .....	38
5.4 MEMPREDICTION .....	41
六、LSFMONITOR 常见问题及解决 .....	42
6.1 图形显示问题 .....	42
6.2 JOBS 页信息缺失 .....	42

6.3 LICENSE 页信息缺失 .....	42
6.4 HOSTS 页和 LOAD 页中的 MEM 值为什么不一致 .....	43
七、技术支持 .....	44
附录 .....	45
附 1. 变更历史 .....	45
附 2. LSF 任务 EXIT CODE 含义 .....	48

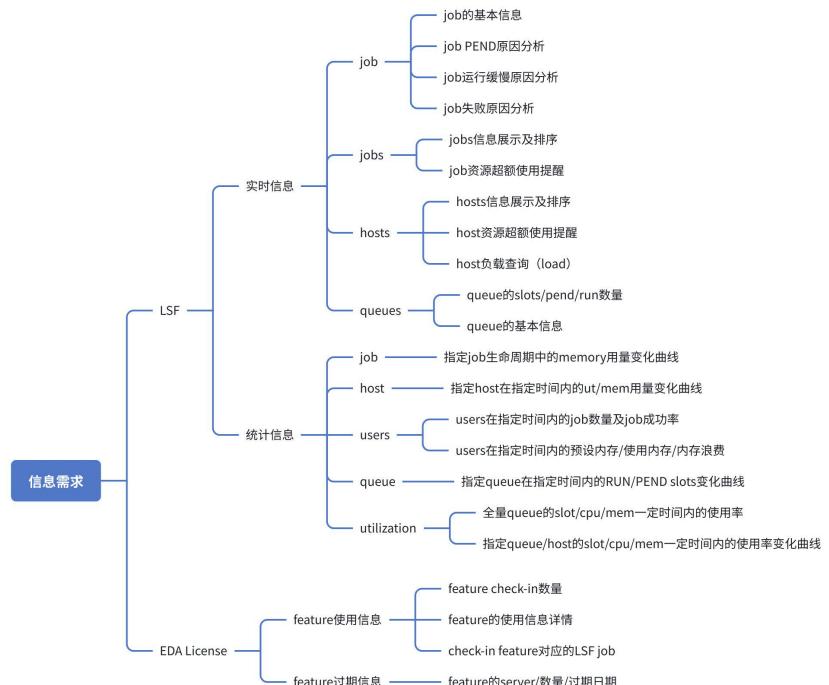
# 一、简介

## 1.1 功能简介

LSF 是 IBM 旗下的一款分布式集群管理系统软件，负责计算资源的管理和批处理作业的调度。它具有良好的可伸缩性和高可用性，支持几乎所有的主流操作系统，是高性能计算的重要基础软件。

lsmMonitor 是一款适用于 LSF/volclava/openlava4 数据收集、分析及展示的开源工具，亦可用于 EDA license 实时信息检索，可以满足集成电路行业用户对于 LSF/license 的绝大部分信息需求，可以平替 RTM 的基本功能。

在集成电路设计 Linux 环境中，关于 LSF 集群和 EDA license 使用状况，用户常常会面临如下信息需求：



lsmMonitor 就是为如上信息需求提供一站式解决方案。

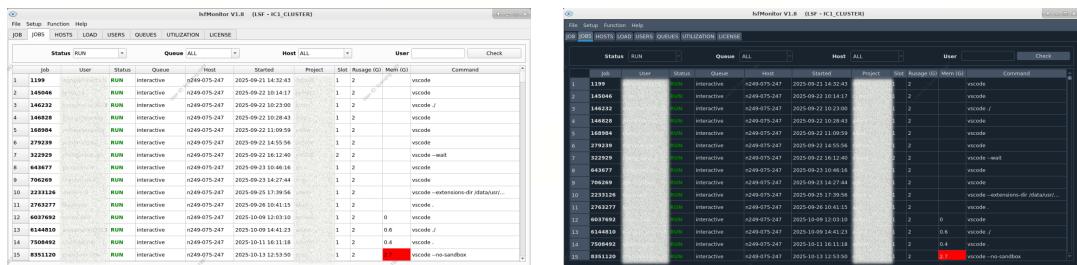
## 1.2 快速了解

lsmMonitor 安装目录下的 monitor/bin 目录内，包含两个主脚本 bsample 和 bmonitor，bsample 用于信息采集，bmonitor 用于信息展示。

lsmMonitor 的正确使用姿势是：安装 -> 配置 -> bsample 采样 -> bmonitor 展示



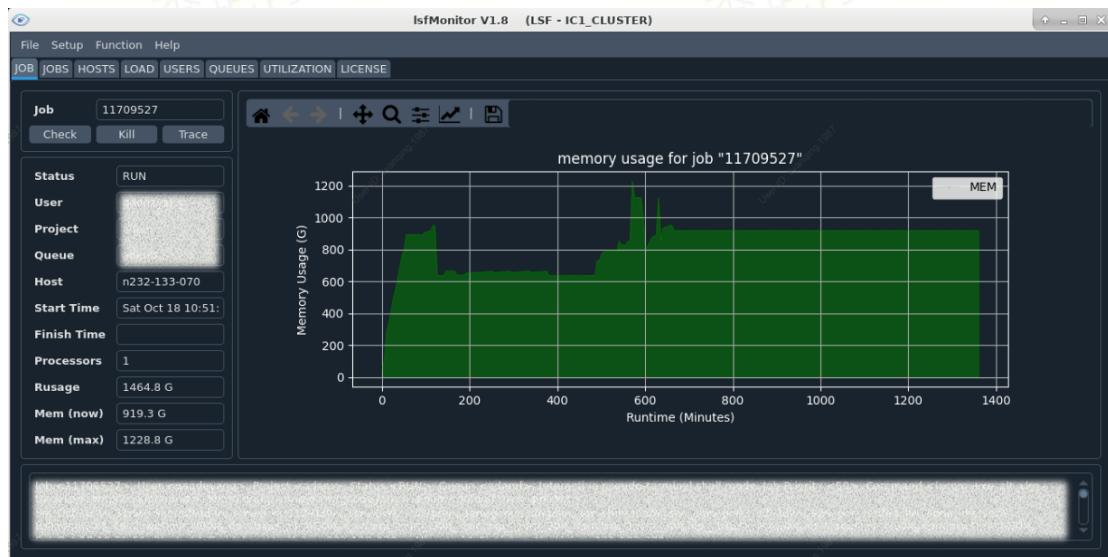
bmonitor 支持“亮色”和“暗黑”两种界面模式。快速上手章节我们采用暗色系展示，工具使用章节则采用默认的亮色系展示。



lsmMonitor 共包含 JOB/JOB/HOSTS/LOAD/USERS/QUEUES/UTILIZATION/LICENSE 八个页面，打开初始默认为 JOBS 页，下面来依次展示不同页面。

## JOB 页：

展示单个 job 的信息。在 Job 文本框输入 jobid，点击 Check 按钮，左侧会显示 job 的关键信息，(如已采样) 右侧会显示 job 生命周期内的内存用量曲线，下侧会显示 job 的详细信息。



## JOBS 页：

展示所有 RUN 状态的 job 及其关键信息，也可以按照 Status/Queue/Host/User 来精准筛选 job。

The screenshot shows the LsfMonitor V1.8 interface with the 'JOBS' tab selected. The main area is a table listing 15 jobs, all currently in a 'RUN' state. The columns in the table are: Job, User, Status, Queue, Host, Started, Project, Slot, Rusage (G), Mem (G), and Command. Most jobs are running on host n249-075-247, with one job (Job 8351120) running on host n249-075-248. The 'Command' column shows that most jobs are running 'vscode' or 'vscode .' except for one job which is running 'vscode --no-sandbox'.

	Job	User	Status	Queue	Host	Started	Project	Slot	Rusage (G)	Mem (G)	Command
1	1199		RUN	interactive	n249-075-247	2025-09-21 14:32:43		1	2		vscode
2	145046		RUN	interactive	n249-075-247	2025-09-22 10:14:17		1	2		vscode
3	146232		RUN	interactive	n249-075-247	2025-09-22 10:23:00		1	2		vscode ./
4	146828		RUN	interactive	n249-075-247	2025-09-22 10:28:43		1	2		vscode
5	168984		RUN	interactive	n249-075-247	2025-09-22 11:09:59		1	2		vscode
6	279239		RUN	interactive	n249-075-247	2025-09-22 14:55:56		1	2		vscode
7	322929		RUN	interactive	n249-075-247	2025-09-22 16:12:40		2	2		vscode --wait
8	643677		RUN	interactive	n249-075-247	2025-09-23 10:46:16		1	2		vscode
9	706269		RUN	interactive	n249-075-247	2025-09-23 14:27:44		1	2		vscode
10	2233126		RUN	interactive	n249-075-247	2025-09-25 17:39:56		1	2		vscode --extensions-dir /data/usr/...
11	2763277		RUN	interactive	n249-075-247	2025-09-26 10:41:15		1	2		vscode .
12	6037692		RUN	interactive	n249-075-247	2025-10-09 12:03:10		1	2	0	vscode
13	6144810		RUN	interactive	n249-075-247	2025-10-09 14:41:23		1	2	0.6	vscode ./
14	7508492		RUN	interactive	n249-075-247	2025-10-11 16:11:18		1	2	0.4	vscode .
15	8351120		RUN	interactive	n249-075-247	2025-10-13 12:53:50		1	2	2.7	vscode --no-sandbox

## HOSTS 页：

展示所有 LSF 服务器的静态/动态信息，也可以按照 Status/Queue/MAX/MaxMem/Host 来精准筛选 host。

	Host	Status	Queue	MAX	Njobs	Ut (%)	MaxMem (G)	aMem (G)	saMem (G)	MaxSwp (G)	Swp (G)	Tmp (G)
1	ic-lsf-main-m1	ok	n019-028-021	32	0	3	123	109	109	0	0	459
2	ic-lsf-main-m2	ok	n019-028-022	32	0	0	123	121	121	0	0	462
3	ic-lsf-main-m3	closed_busy	n019-028-023	32	0	17	123	48	48	0	0	376
4	n019-028-021	ok	n019-028-024	120	112	24	992	443	338	127	127	1780
5	n019-028-024	ok	n019-028-025	57	21	2	958	933	314	127	127	1799
6	n019-028-025	ok	n019-028-026	57	20	2	958	932	334	127	127	1799
7	n019-028-026	ok	n019-028-027	57	21	2	958	927	314	127	127	1799
8	n019-028-027	ok	n019-028-028	57	20	2	958	933	324	127	127	1799
9	n019-028-028	ok	n019-028-029	57	21	5	958	911	313	127	127	1799
10	n019-028-029	ok	n019-028-030	57	22	3	958	903	284	127	127	1799
11	n019-028-030	ok	n019-028-031	57	20	3	958	887	323	127	127	1799
12	n019-028-031	ok	n019-028-032	57	21	1	958	929	193	127	127	1799
13	n019-028-032	ok	n019-028-033	57	20	2	958	924	224	127	127	1799
14	n019-028-033	ok	n019-028-036	57	20	1	958	934	243	127	127	1799
15	n019-028-036	ok		96	9	0	744	728	491	127	127	1799

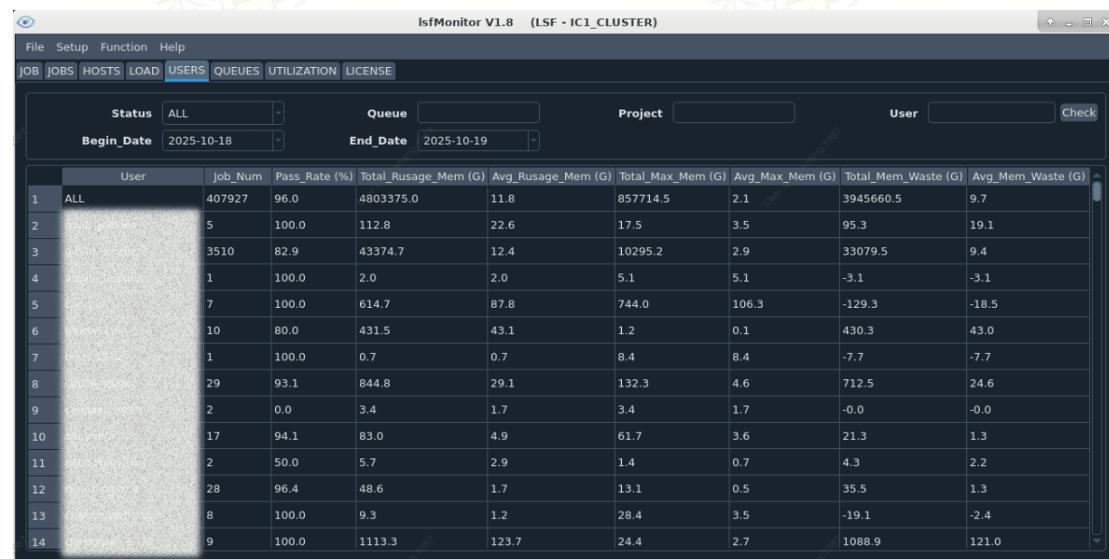
## LOAD 页：

展示指定服务器在指定时间段的 ut/mem 负载变化。



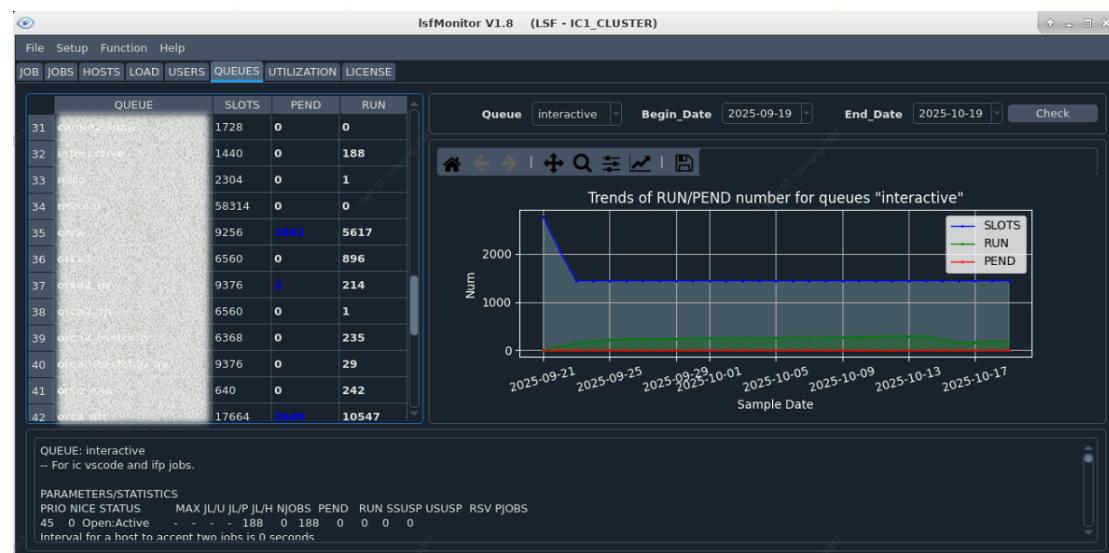
## USER 页：

展示指定时间段内用户任务的基本汇总信息，包括 job 数目、成功率、预设内存、最大内存用量、内存浪费，也可以按照 Status/Queue/Project/User 来精准筛选用户数据。



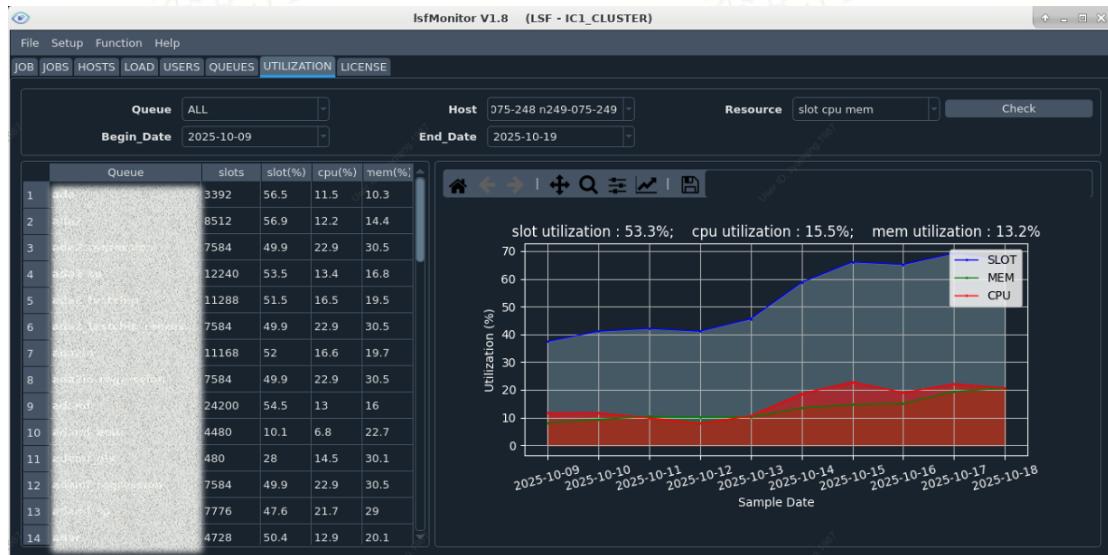
## QUEUES 页：

展示所有 queue 瞬时的 SLOTS/PEND/RUN job 数目，如果点击指定的 queue，则右侧展示指定时间段内的 SLOTS/PEND/RUN job 数据变化趋势，下侧展示 queue 的详细信息。



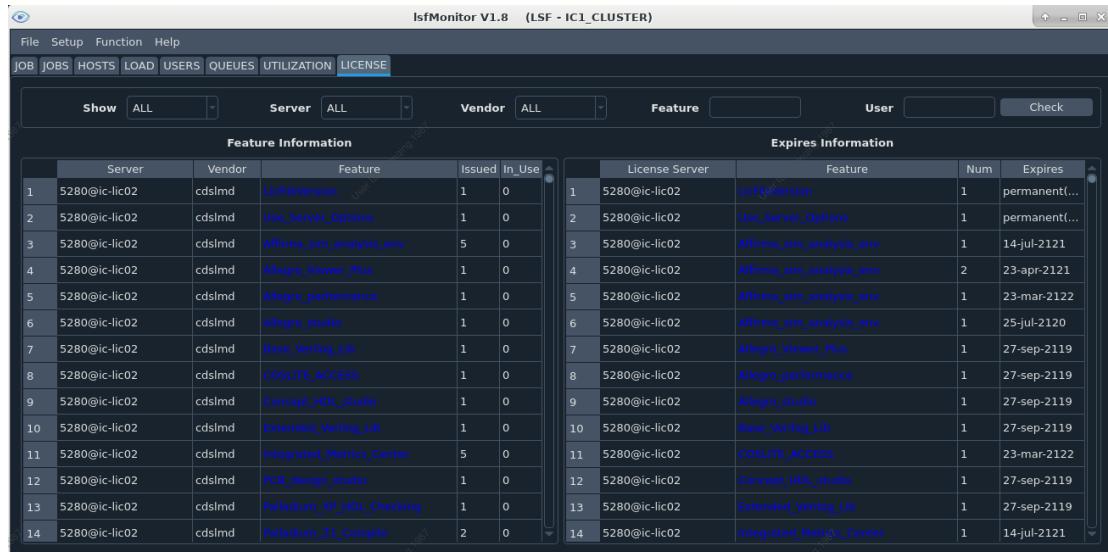
## UTILIZATION 页：

展示指定时间段内所有 queue 的 slots/slots 利用率/cpu 利用率/mem 利用率信息，可以按照 Queue/Host 来指定逻辑单元，则会在右侧显示指定服务器单元在指定时间段内的利用率信息。



## LICENSE 页：

会根据当前 Terminal 中用户的 **LM\_LICENSE\_FILE** 环境变量的设置，展示相关 license server 上所有 feature 的总量/用量信息。



## 二、环境依赖

### 2.1 操作系统依赖

lsmMonitor 的开发和测试操作系统为 **CentOS Linux release 7.9.2009 (Core)** 和 **Rocky Linux release 8.10 (Green Obsidian)**, 这也是 IC 设计常用的操作系统版本。

centos7/centos8/rocky8/redhat7/redhat8 理论上都可以运行, 主要的潜在风险在于系统库版本差异可能会影响部分组件的正常运行。

GCC 和 OpenSSL 需要满足如下版本要求, 最好使用 NAS 安装的版本, 以做到跨服务器、跨操作系统版本通用。

GCC >= 8.5

OpenSSL = 1.1.1

### 2.2 python 版本依赖

lsmMonitor 基于 python 开发, 其开发和测试的 python 版本为 **python3.12.12** (早期为 3.8.8)。

python 安装时可以参照如下环境载入方式和编译命令。

```
./configure \
  --prefix=${PYTHON_ROOT}/python3.12.12 # 指定安装路径（建议自定义, 避免覆盖系统 Python）
  --enable-optimizations           # 启用优化（提升 Python 运行速度）
  --with-ssl=${OPENSSL_ROOT}        # 强制启用 SSL 模块（依赖 libssl-dev）

make -j8
make altinstall
```

### 2.3 集群管理工具

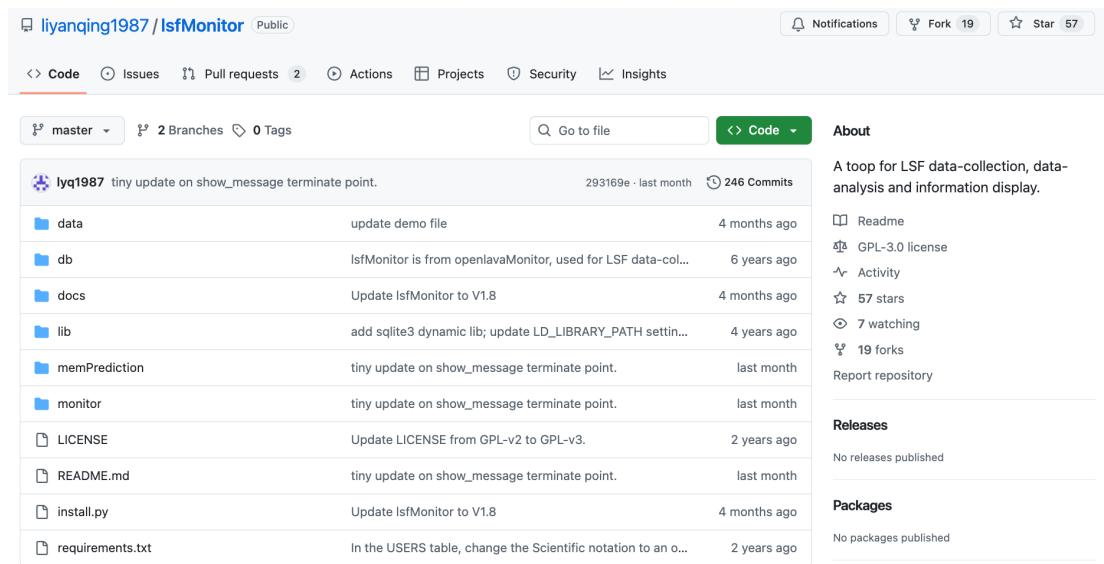
lsmMonitor 依赖 LSF/volclava/Openlava 集群管理系统, 暂不支持其它集群管理系统。

LSF 9.1.3 及以上的版本良好支持, volclava 支持良好, Openlava 几个版本间输出信息格式有一定差异, 仅支持 openlava4.0 的主流版本。

## 三、工具安装及配置

### 3.1 工具下载

lsfMonitor 的 github 路径位于 <https://github.com/liyanqing1987/lsfMonitor>



The screenshot shows the GitHub repository page for 'liyanqing1987/lsfMonitor'. The repository is public and has 246 commits. The code tab is selected. The repository description is: 'A tool for LSF data-collection, data-analysis and information display.' It includes links to Readme, GPL-3.0 license, Activity, 57 stars, 7 watching, 19 forks, and a Report repository. There are sections for Releases (No releases published) and Packages (No packages published).

File	Description	Last Commit
data	update demo file	4 months ago
db	lsfMonitor is from openlavaMonitor, used for LSF data-col...	6 years ago
docs	Update lsfMonitor to V1.8	4 months ago
lib	add sqlite3 dynamic lib; update LD_LIBRARY_PATH settin...	4 years ago
memPrediction	tiny update on show_message terminate point.	last month
monitor	tiny update on show_message terminate point.	last month
LICENSE	Update LICENSE from GPL-v2 to GPL-v3.	2 years ago
README.md	tiny update on show_message terminate point.	last month
install.py	Update lsfMonitor to V1.8	4 months ago
requirements.txt	In the USERS table, change the Scientific notation to an o...	2 years ago

可以采用“`git clone https://github.com/liyanqing1987/lsfMonitor.git`”的方式拉取源代码。

```
[liyanqing.1987@LQ2KW66YXF ~/test]#git clone  
git@github.com:liyanqing1987/lsfMonitor.git  
Cloning into 'lsfMonitor'...  
remote: Enumerating objects: 1320, done.  
remote: Counting objects: 100% (477/477), done.  
remote: Compressing objects: 100% (248/248), done.  
remote: Total 1320 (delta 267), reused 434 (delta 226), pack-  
reused 843 (from 1)  
Receiving objects: 100% (1320/1320), 44.45 MiB | 975.00 KiB/s,  
done.  
Resolving deltas: 100% (816/816), done.
```

也可以在 lsfMonitor 的 github 页面上，Code -> Download ZIP 的方式拉取代码包。

The screenshot shows the GitHub repository page for 'liyanqing1987/lsfMonitor'. The 'Code' tab is selected. On the right, there's a 'Clone' section with options for 'HTTPS' and 'GitHub CLI', and a 'Download ZIP' button which is highlighted with a red box. Below this are sections for 'About', 'Releases', and 'Packages'.

## 3.2 工具安装

工具安装之前，首先参照第二章“环境依赖”满足 lsfMonitor 的环境依赖关系。

安装包下的文件和目录如下。

```
[root@ic-admin01 tools]# cd lsfMonitor/
[root@ic-admin01 lsfMonitor]# ls -p
data/  db/  docs/  install.py  lib/  LICENSE  memPrediction/
monitor/  README.md  requirements.txt
```

确认 python 版本正确 (Python 3.12.12)，并基于安装包中的 requirements.txt 安装 python 依赖库。(此步需要 root 权限并且支持 pip install，推荐在 IC 内网配置 pip 源实现)

```
[root@ic-admin01 lsfMonitor]# python3 --version
Python 3.12.12
[root@ic-admin01 lsfMonitor]#
[root@ic-admin01 lsfMonitor]# pip3 install -r requirements.txt
Looking in indexes: http://mirrors.ic/bytedance.com/pypi/simple
Collecting qdarkstyle==3.2.3 (from -r requirements.txt (line 1))
  Using cached
  http://mirrors.ic/bytedance.com/pypi/packages/93/7d/c3c10498430dad
  cea4def5faddf71cd199e577d20a125e7ef1e9d7bdbbf/QDarkStyle-3.2.3-
  py2.py3-none-any.whl (871 kB)
```

```
Collecting PyQt5==5.15.11 (from -r requirements.txt (line 2))
...
Successfully installed PyQt5-5.15.11 PyQt5-Qt5-5.15.18 PyQt5-sip-12.18.0 aniso8601-10.0.1 anyio-4.12.1 blinker-1.9.0 certifi-2026.1.4 charset_normalizer-3.4.4 click-8.3.1 contourpy-1.3.3 cycler-0.12.1 elastic-transport-9.2.1 elasticsearch-9.2.1 es_pandas-0.0.23 flask-3.1.2 flask_restful-0.3.10 fonttools-4.61.1 gensim-4.3.3 glove-py-0.1.0 idna-3.11 imbalanced-learn-0.14.1 imblearn-0.0 itsdangerous-2.2.0 jinja2-3.1.6 joblib-1.5.3 kiwisolver-1.4.9 markupsafe-3.0.3 matplotlib-3.10.8 numpy-1.26.4 nvidia-nccl-cu12-2.29.3 packaging-26.0 pandas-2.3.3 pillow-12.1.0 pybind11-3.0.1 pyparsing-3.3.2 python-dateutil-2.9.0.post0 pytz-2025.2 pyyaml-6.0.3 qdarkstyle-3.2.3 qtpy-2.4.3 requests-2.32.5 scikit-learn-1.7.2 scipy-1.13.1 screeninfo-0.8.1 setuptools-80.10.2 six-1.17.0 sklearn-compat-0.1.5 smart-open-7.5.0 sniffio-1.3.1 tabulate-0.9.0 threadpoolctl-3.6.0 tqdm-4.67.1 typing-extensions-4.15.0 tzdata-2025.3 urllib3-2.6.3 werkzeug-3.1.5 wrapt-2.1.1 xgboost-3.0.5
```

在安装目录下，使用命令“python3 install.py”安装 lsfMonitor。

```
[root@ic-admin01 lsfMonitor]# python3 install.py
>>> Check python version.
      Required python version : 3.12.12
      Current  python version : 3.12.12

>>> Generate shell tools
      Generate
      "/ic/data/usr/liyanqing.1987/tools/lsfMonitor/monitor/bin/bmonitor"
".
      Generate
      "/ic/data/usr/liyanqing.1987/tools/lsfMonitor/monitor/bin/bsample"
".
      Generate
      "/ic/data/usr/liyanqing.1987/tools/lsfMonitor/monitor/tools/akill"
".
      Generate
      "/ic/data/usr/liyanqing.1987/tools/lsfMonitor/monitor/tools/check_
issue_reason".
      Generate
      "/ic/data/usr/liyanqing.1987/tools/lsfMonitor/monitor/tools/patch"
".
      Generate
```

```
"/ic/data/usr/liyanqing.1987/tools/lsfMonitor/monitor/tools/process_tracer".
    Generate
"/ic/data/usr/liyanqing.1987/tools/lsfMonitor/monitor/tools/seedb"
.
    Generate
"/ic/data/usr/liyanqing.1987/tools/lsfMonitor/monitor/tools/show_license_feature_usage".

>>> Generate config file
"/ic/data/usr/liyanqing.1987/tools/lsfMonitor/monitor/conf/config.py".

Done, Please enjoy it.
```

请注意，此处的 `install.py` 是支持多个参数的，如果是初次安装且仅使用 `LsfMonitor`，则不需要加任何参数。

```
python3 install.py -h
usage: install.py [-h] [-p PREFIX] [-c] [-f] [-m]

options:
-h, --help            show this help message and exit
-p PREFIX, --prefix PREFIX
                      Specify lsfMonitor install path on config
file, default is current directory.
-c, --clean           Cleanup old installation.
-f, --force            Install by force.
-m, --memPrediction   Install memPrediction the same time.
```

--clean：清理旧的安装操作。

--force：强制初始化 `config.py`，用于二次安装的情况不留旧的设置，慎用。

--memPrediction：同时安装 `memPrediction` 这个附加工具。

### 3.3 工具配置

安装目录下主要的配置文件为 monitor/conf/config.py，用于配置工具的一些基本设置和验证规则。

安装后默认配置如下，一般需要重新配置。

```
# Specify the database directory.  
db_path = "/ic/software/tools/lsfMonitor/db"  
  
# Specify EDA license administrators.  
license_administrators = "all"  
  
# Specify lmstat path, example  
"/eda/synopsys/scl/2021.03/linux64/bin/lmstat".  
lmstat_path = "/ic/software/tools/lsfMonitor/monitor/tools/lmstat"  
  
# Specify lmstat bsub command, example "bsub -q normal -Is".  
lmstat_bsub_command = ""  
  
# Excluded license servers, format is "27020@lic_server  
5280@lic_server".  
excluded_license_servers = ""
```

**db\_path**: 采样数据的数据库存放路径，默认为 lsfMonitor 安装路径下的 db 目录。

如果 lsfMonitor 用版本管理工具管理，那么建议把 db\_path 修改到独立的数据存放路径。

**license\_administrators**: 设定 license 信息的管理员，仅指定的用户可见 LICENSE 页，默认为“all”，即所有的人均可见。

**lmstat\_path**: lsfMonitor 通过工具 lmstat 获取 EDA license 信息，此处用于配置 lmstat 工具的路径。

**lmstat\_bsub\_command**: lsfMonitor 一般在 Linux 环境的 login server 上运行，而 login server 一般会通过 iptables 等方法设置禁止 lmstat 等 EDA 相关的工具运行，所以执行 lmstat 的时候需要 bsub 出去，此处用于指定执行 lmstat 时候的 bsub 命令。

**excluded\_license\_servers**: license 信息检索时排除指定的 license servers，被排除的 license server 信息将不会显示在 LICENSE 页，默认为“”，即不排除任何 license server。

一个使用的 demo 配置如下。

```
# Specify the database directory.  
db_path = "/ic/software/tools/lsfMonitor/db"  
  
# Specify EDA license administrators.  
license_administrators = "root liyanqing.1987"  
  
# Specify lmstat path, example  
"/eda/synopsys/scl/2021.03/linux64/bin/lmstat".  
lmstat_path = "/ic/software/tools/lsfMonitor/monitor/tools/lmstat"  
  
# Specify lmstat bsub command, example "bsub -q normal -Is".  
lmstat_bsub_command = ""  
  
# Excluded license servers, format is "27020@lic_server  
5280@lic_server".  
excluded_license_servers = "27020@ic-lic01 5280@ic-lic02"
```

## 四、工具使用

lsmMonitor 工具包括“数据采集”和“数据展示”两大部分，对应的执行脚本分别为 bsample 和 bmonitor，均位于 lsmMonitor 安装路径下的 monitor/bin 子目录中。

### 4.1 数据采集 bsample

#### 4.1.1 帮助信息

bsample 用于采集 LSF/openlava 的 job/queue/host/load/user/utilization 信息。

```
[root@ic-admin01 lsmMonitor]# monitor/bin/bsample -h
usage: bsample.py [-h] [-j] [-m] [-q] [-H] [-l] [-u] [-U] [-UD]

optional arguments:
  -h, --help            show this help message and exit
  -c, --cleanup         Clean up database with entries limitation.
  -j, --job              Sample (finished) job info with command
"bjobs -u all -d -UF".
  -m, --job_mem         Sample (running) job memory usage
information with command "bjobs -u all -r -UF".
  -q, --queue            Sample queue info with command "bqueues".
  -H, --host             Sample host info with command "bhosts".
  -l, --load              Sample host load (ut/tmp/swp/mem) info
with command "lsload".
  -u, --user              Sample user (finished) job info with
command "bjobs -u all -d -UF".
  -U, --utilization      Sample utilization (slot/cpu/mem) info
with command "lsload/bhosts/lshosts".
  -UD, --utilization_day
                        Count and save utilization-day info with
utilization data.
```

**--help:** 打印帮助信息。

**--cleanup:** 清理超出条目限制的数据库表单。

**--job:** 采集 job 信息并存储。

**--job\_mem:** 采集 job 的 MEM 用量信息并存储。

**--queue:** 采集 queue 信息并存储。

- host:** 采集 host 信息并存储。(bmonitor 暂时不需要)
- load:** 采集 host load 信息并存储。
- user:** 采集 user 信息并存储。
- utilization:** 采集 slot/cpu/memory 的 utilization 信息。
- utilization\_day:** 根据 utilization 数据计算按天核算的 utilization 值。

注意，在 V1.5 及更早的版本中，“bsample -j”是采集 job MEM 用量信息，而在 V1.5.1 及以后版本中，这个功能变更为“bsample -m”。

### 4.1.2 手工采样

下面是一个手工采样示例，我们使用 monitor/bin/bsample 采样 job MEM 用量的信息。

```
[root@ic-admin01 lsfMonitor]# monitor/bin/bsample -m
>>> Sampling job mem usage info ...
      Done (5744 jobs).
```

### 4.1.3 自动采样（定时）

首先，建议 LSF 的 CLEAN\_PERIOD 参数至少设置为 86400 (一天)，默认值一般为 3600 (一小时)，这个值过小的话容易导致部分采样中出现数据缺失。

我们推荐用 crontab 来自动定时采样，job/user 采样较慢所以一天两次即可，其它采样建议 5~10 分钟一次。下面是一个示例。（crontab -e）

bsample -c 则用于自动清理 database 中的旧数据，以防止数据库尺寸过大。

```
SHELL=/bin/bash
PATH=/usr/local/bin:/bin:/usr/bin:/usr/local/sbin:/usr/sbin:/ic/software/tools/lsf/10.1/linux3.10-glibc2.17-x86_64/bin
LSF_SERVERDIR=/ic/software/tools/lsf/10.1/linux3.10-glibc2.17-x86_64/etc
LSF_LIBDIR=/ic/software/tools/lsf/10.1/linux3.10-glibc2.17-x86_64/lib
LSF_BINDIR=/ic/software/tools/lsf/10.1/linux3.10-glibc2.17-x86_64/bin
LSF_ENVDIR=/ic/software/tools/lsf/conf
```

```
LSF_TOP=/ic/software/tools/lsf

# For lsfMonitor
3 0 * * * /ic/software/tools/lsfMonitor/monitor/bin/bsample -c
10 11,23 * * * /ic/software/tools/lsfMonitor/monitor/bin/bsample -
j
*/5 * * * * /ic/software/tools/lsfMonitor/monitor/bin/bsample -m
*/5 * * * * /ic/software/tools/lsfMonitor/monitor/bin/bsample -q
*/5 * * * * /ic/software/tools/lsfMonitor/monitor/bin/bsample -H
*/5 * * * * /ic/software/tools/lsfMonitor/monitor/bin/bsample -l
30 11,23 * * * /ic/software/tools/lsfMonitor/monitor/bin/bsample -
u
*/10 * * * * /ic/software/tools/lsfMonitor/monitor/bin/bsample -U
55 23 * * * /ic/software/tools/lsfMonitor/monitor/bin/bsample -UD
```

请注意，`crontab` 中默认是没有任何环境的，所以需要在 `crontab` 中设置好 `PATH` 及 `LSF_*` 等变量，否则 `bsample` 中引用的 `bjobs` 等工具无法生效。这些变量可以通过如下方式获取。

```
[root@ic-admin01 lsfMonitor]# echo $PATH
/usr/local/bin:/bin:/usr/bin:/usr/local/sbin:/usr/sbin:/ic/software/tools/lsf/10.1/linux3.10-glibc2.17-x86_64/bin
[root@ic-admin01 lsfMonitor]#
[root@ic-admin01 lsfMonitor]# env | grep "LSF_"
LSF_SERVERDIR=/ic/software/tools/lsf/10.1/linux3.10-glibc2.17-x86_64/etc
LSF_LIBDIR=/ic/software/tools/lsf/10.1/linux3.10-glibc2.17-x86_64/lib
LSF_BINDIR=/ic/software/tools/lsf/10.1/linux3.10-glibc2.17-x86_64/bin
LSF_ENVDIR=/ic/software/tools/lsf/conf
LSF_TOP=/ic/software/tools/lsf
```

#### 4.1.4 数据库

`lsfMonitor` 支持多 LSF/openlava clusters，会根据 `cluster` 来存放数据，所以有可能在 `db_path` 下面看到多个 `cluster` 的采样数据目录。

测试环境的 `cluster` 信息为“IC1\_CLUSTER”。

```
[root@ic-admin01 lsfMonitor]# lsid
```

```
...
My cluster name is IC1_CLUSTER
My master name is ic-lsf-main-m1
```

所以采样目录为同名目录。

```
[root@ic-admin01 lsfMonitor]# ls db
IC1_CLUSTER
```

采样目录下的数据如下。

```
[root@ic-admin01 lsfMonitor]# ls -p db/IC1_CLUSTER/
host.db  job/  job_mem/  load.db  queue.db  user/
utilization_day.db  utilization.db
[root@ic-admin01 lsfMonitor]#
[root@ic-admin01 lsfMonitor]# ls -p db/IC1_CLUSTER/job/
20250825  20250826  20250908  20250910  20250925  20250926
20250927
[root@ic-admin01 lsfMonitor]#
[root@ic-admin01 lsfMonitor]# ls -p db/IC1_CLUSTER/job_mem/
73970000_73979999.db  90700000_90709999.db
93550000_93559999.db ...
[root@ic-admin01 lsfMonitor]#
[root@ic-admin01 lsfMonitor]# ls -p db/IC1_CLUSTER/user/
20250825  20250826  20250908  20250910  20250925  20250926
20250927
```

**host.db:** 记录 host 的静态信息，由“bsample -H”生成。

**job/<date>:** 记录 job 历史信息，文件为 json 格式，由“bsample -j”生成。

**job\_mem/\*.db:** 记录 job 的 mem usage 信息，由“bsample -m”生成。

**load.db:** 记录 host 的 load 信息，由“bsample -l”生成。

**queue.db:** 记录 queue 的 run/pending slot 信息，由“bsample -q”生成。

**user/<date>:** 记录用户的 job 关键信息，有“bsample -u”生成。

**utilization\_day.db:** 记录 slot/cpu/mem 的 utilization 信息，按天汇聚，由“bsample -UD”生成。

**utilization.db:** 记录 slot/cpu/mem 的 utilization 信息，由“bsample -U”生成。

### 特别注意：

V1.3 版本开始，**bsample** 采样数据库的数据格式跟旧版本有变化，且不向前兼容，不能复用旧的数据库。

V1.4 版本开始，**bsample** 采样 queue.db 数据库格式再次有了变化，且不向前兼容，不能复用旧的数据库。

V1.4.2 版本开始，**bsample** 采样存储目录名从“monitor”切换为 LSF/openlava 的 cluster 名字，这个切换是无感的，唯一的影响是旧的“monitor”目录下的数据会失效，可以通过手工拷贝到新路径复用旧数据。

V1.5.1 版本开始，**bsample** 采样 job MEM 用量的存储目录名从“job”切换为“job\_mem”，这个切换是无感的，唯一的影响是旧的“job”目录下的数据会失效，可以通过手工拷贝到“job\_mem”路径下来复用旧数据。

V1.7 版本开始，**bsample** 采样 job 的数据库格式，由 json 转变为 sqlite3 数据库，且不向前兼容，不能复用旧的数据库。

## 4.2 数据展示 **bmonitor**

### 4.2.1 工具载入

lsfMonitor 的核心工具叫做 **bmonitor**，是一个图形界面工具，其载入方式有多种。

- 引用 **bmonitor** 绝对路径。
- 将 **bmonitor** 的路径加入到环境变量 PATH 中，直接执行 **bmonitor** 即可。
- 采用 modules 管理和加载环境，直接执行 **bmonitor** 即可。
- 将 **bmonitor** link 到 LSF 的 bsub 脚本路径中，直接执行 **bmonitor** 即可。

推荐最后一种方式，下面是具体效果。

```
[root@ic-admin01 lsfMonitor]# which bmonitor  
/ic/software/tools/lsf/10.1/linux2.6-glibc2.3-x86_64/bin/bmonitor
```

其启动效果如下所示（测试环境耗时约 29 秒，时间主要耗费在 LSF 实时/历史数据加载和 License 数据加载上）。

```

[root@ic-admin01 bin]# bmonitor
[2025-10-19 10:00:01] lsfMonitor Version: V1.8 (2025.10.17)
[2025-10-19 10:00:01]
[2025-10-19 10:00:01] LSF (10.1.0.12)
[2025-10-19 10:00:01] My cluster name is "IC1_CLUSTER"
[2025-10-19 10:00:01] My master name is "ic-lsf-main-m1"
[2025-10-19 10:00:01]
[2025-10-19 10:00:01] Loading License information ...
[2025-10-19 10:00:13] Loading LSF bqueues information ...
[2025-10-19 10:00:13] Loading LSF bhosts information ...
[2025-10-19 10:00:13] Loading LSF busers information ...
[2025-10-19 10:00:13] Loading LSF jobs information ...
[2025-10-19 10:00:16] Loading LSF lshosts information ...
[2025-10-19 10:00:16] Loading LSF host_queue information ...
*Warning*: Queue "normal" is not well configured, all of the hosts
are on the same queue.
[2025-10-19 10:00:16] Loading LSF lsload information ...
[2025-10-19 10:00:17] Loading LSF bhosts_load information ...
[2025-10-19 10:00:17] Loading user history info ...
[2025-10-19 10:00:21] Loading LSF queue_host information ...
*Warning*: Queue "normal" is not well configured, all of the hosts
are on the same queue.
[2025-10-19 10:00:22] Loading queue utilization info ...
[2025-10-19 10:00:29] Loading resource utilization information ...

```

lsfMonitor V1.8 (LSF - IC1_CLUSTER)											
JOB		JOBS		HOSTS		LOAD		USERS		QUEUES	
	Job	Status	RUN	Queue	ALL	Host	ALL	User	Check	Project	
1	<a href="#">1199</a>	RUN	interactive	n249-075-247	2025-09-21 14:32:43	1	2	vscode			
2	<a href="#">145046</a>	RUN	interactive	n249-075-247	2025-09-22 10:14:17	1	2	vscode			
3	<a href="#">146232</a>	RUN	interactive	n249-075-247	2025-09-22 10:23:00	1	2	vscode ./			
4	<a href="#">146828</a>	RUN	interactive	n249-075-247	2025-09-22 10:28:43	1	2	vscode			
5	<a href="#">168984</a>	RUN	interactive	n249-075-247	2025-09-22 11:09:59	1	2	vscode			
6	<a href="#">279239</a>	RUN	interactive	n249-075-247	2025-09-22 14:55:56	1	2	vscode			
7	<a href="#">322929</a>	RUN	interactive	n249-075-247	2025-09-22 16:12:40	2	2	vscode --wait			
8	<a href="#">643677</a>	RUN	interactive	n249-075-247	2025-09-23 10:46:16	1	2	vscode			
9	<a href="#">706269</a>	RUN	interactive	n249-075-247	2025-09-23 14:27:44	1	2	vscode			
10	<a href="#">2233126</a>	RUN	interactive	n249-075-247	2025-09-25 17:39:56	1	2	vscode --extensions-dir /data/usr/...			
11	<a href="#">2763277</a>	RUN	interactive	n249-075-247	2025-09-26 10:41:15	1	2	vscode .			
12	<a href="#">6037692</a>	RUN	interactive	n249-075-247	2025-10-09 12:03:10	1	2	0	vscode		
13	<a href="#">6144810</a>	RUN	interactive	n249-075-247	2025-10-09 14:41:23	1	2	0.6	vscode ./		
14	<a href="#">7508492</a>	RUN	interactive	n249-075-247	2025-10-11 16:11:18	1	2	0.4	vscode .		
15	<a href="#">8351120</a>	RUN	interactive	n249-075-247	2025-10-13 12:53:50	1	2	2.7	vscode --no-sandbox		

我们可以看到当前集群的基本信息，以及图形界面启动过程中加载数据的过程。

## 4.2.2 帮助信息

直接执行 bmonitor 会启动图形界面。

执行“bmonitor -h”则可以查看 bmonitor 的帮助信息， bmonitor 的参数主要用于初始化部分信息， 不过这些参数一般也可以在 bmonitor 启动后设置。

```
[root@ic-admin01 lsfMonitor]# bmonitor -h
usage: bmonitor.py [-h] [-j JOBID] [-u USER] [-f FEATURE] [-t
{JOB,JOBS,HOSTS,LOAD,USERS,QUEUES,UTILIZATION,LICENSE}] [--disable_license] [-d]

optional arguments:
  -h, --help            show this help message and exit
  -j JOBID, --jobid JOBID
                        Specify the jobid which show it's
                        information on "JOB" tab.
  -u USER, --user USER  Specify the user show how's job
                        information on "JOBS" tab.
  -f FEATURE, --feature FEATURE
  -t {JOB,JOBS,HOSTS,LOAD,USERS,QUEUES,UTILIZATION,LICENSE}, --tab
{JOB,JOBS,HOSTS,LOAD,USERS,QUEUES,UTILIZATION,LICENSE}
                        Specify current tab, default is "JOBS"
                        tab.
  --disable_license      Disable license check function.
  -d, --dark_mode        Enable dark mode on the main interface.
```

**--help:** 打印帮助信息。

**--jobid:** 指定 jobid， 用于切换到 JOB 页并直接显示指定 jobid 的信息， 此时其它页内容并不加载， 以加快 GUI 打开速度。

**--user:** 指定 user， 用于切换到 JOBS 页并显示指定用户的所有 job 信息。

**--feature:** 指定 license feature， 用于切换到 LICENSE 页并显示指定 license feature 的信息。

**--tab {JOB, JOBS, HOSTS, QUEUES, LOAD, UTILIZATION, LICENSE}:** 指定页面， 会将 bmonitor 打开到指定 GUI 页面。

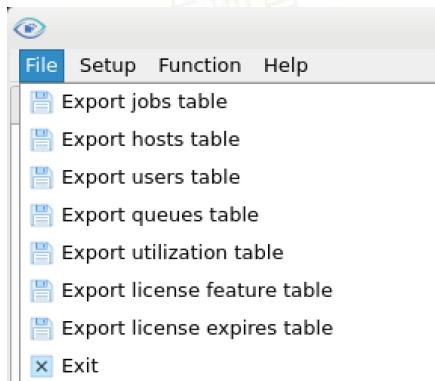
**--disable\_license:** 启动的时候不执行 license 信息获取步骤， 以加快 GUI 打开速度。

**--dark\_mode:** 启用暗黑主题模式。

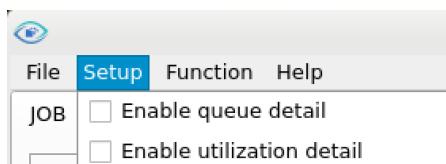
### 4.2.3 菜单栏

bmonitor 菜单栏包含 File, Setup, Function, Help 四部分。

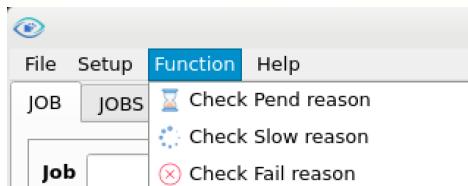
**File:** 包含 Export \* table 功能和 Exit 功能。



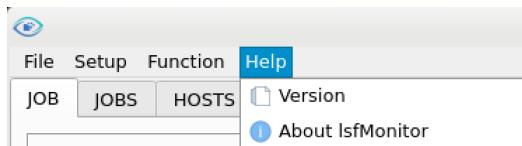
**Setup:** 包含“Enable queue detail”和“Enable utilization detail”两个复选框。



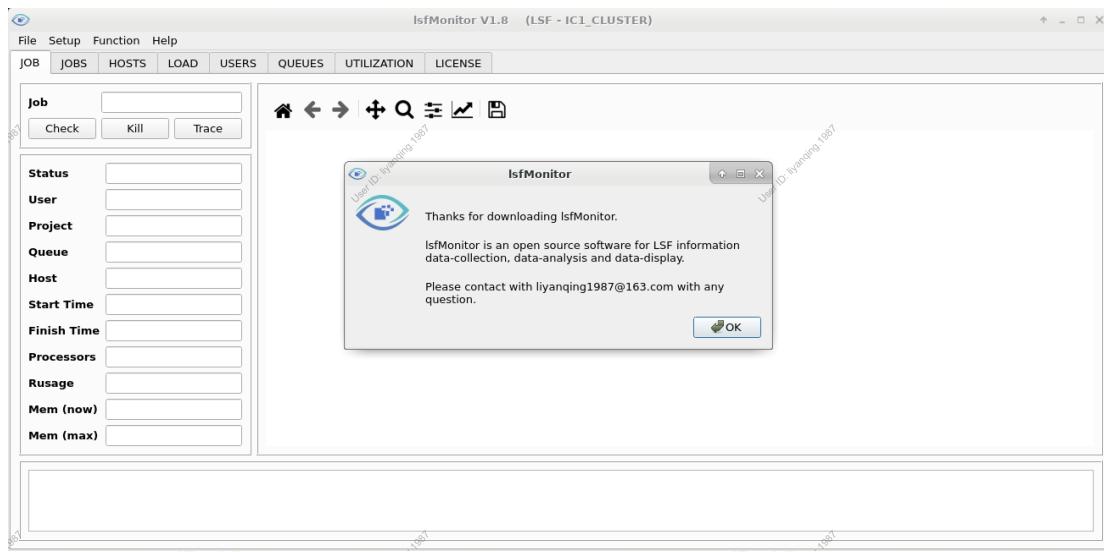
**Function:** 包含“Check Pend reason”、“Check Slow reason”和“Check Fail reason”三个功能。



**Help:** 包含“Version”和“About lsfMonitor”两个信息项。



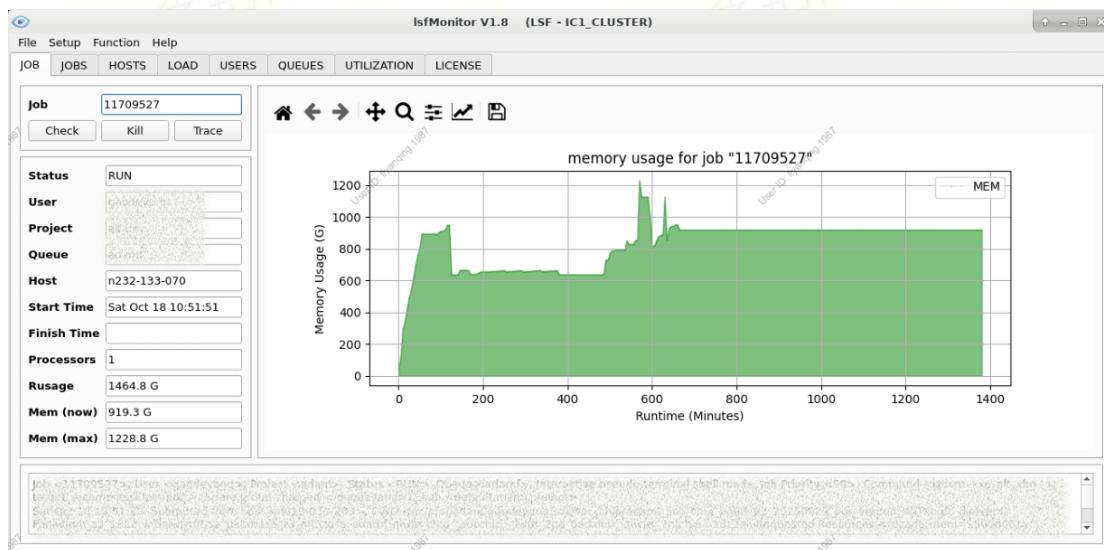
Help 菜单中的“About lsfMonitor”内容如下。



#### 4.2.4 JOB 页

JOB 页主要用于查看指定 job 的详细信息，以及 job 内存用量的历史曲线。

在 Job 框输入 jobid，点击 Check 按钮，可以查看指定 job 的详细信息（来源于 bjobs -UF <JOBID>）和 job 的内存用量曲线。



说明：

- 右边栏显示 job 生命周期内的 memory 用量曲线，如未显示，可能是没有启动周期性的采样，或者采样了但是 job 的 runtime 太短未采到。
- 下侧显示 job 的详细信息。（通过 bjobs -UF <jobid>获取）

- 点击“Kill”按钮，在有权限的情况下会 kill 掉当前 job，
- 点击“Trace”按钮，会追踪当前 job pend/slow/fail 的原因（仅能 trace 自己的 job）。
- 在有 job 信息的情况下，在 Host 框中敲击回车，会跳转到 LOAD 页，并显示第一台 server 的负载变化情况。

## 4.2.5 JOBS 页

JOBS 页主要用于批量查看 jobs 的关键信息。

	Job	User	Status	Queue	Host	Started	Project	Slot	Rusage (G)	Mem (G)	Command
1	<b>1199</b>		RUN	interactive	n249-075-247	2025-09-21 14:32:43		1	2		vscode
2	<b>145046</b>		RUN	interactive	n249-075-247	2025-09-22 10:14:17		1	2		vscode
3	<b>146232</b>		RUN	interactive	n249-075-247	2025-09-22 10:23:00		1	2		vscode ./
4	<b>146828</b>		RUN	interactive	n249-075-247	2025-09-22 10:28:43		1	2		vscode
5	<b>168984</b>		RUN	interactive	n249-075-247	2025-09-22 11:09:59		1	2		vscode
6	<b>279239</b>		RUN	interactive	n249-075-247	2025-09-22 14:55:56		1	2		vscode
7	<b>322929</b>		RUN	interactive	n249-075-247	2025-09-22 16:12:40		2	2		vscode --wait
8	<b>643677</b>		RUN	interactive	n249-075-247	2025-09-23 10:46:16		1	2		vscode
9	<b>706269</b>		RUN	interactive	n249-075-247	2025-09-23 14:27:44		1	2		vscode
10	<b>2233126</b>		RUN	interactive	n249-075-247	2025-09-25 17:39:56		1	2		vscode --extensions-dir /data/usr/...
11	<b>2763277</b>		RUN	interactive	n249-075-247	2025-09-26 10:41:15		1	2		vscode .
12	<b>6037692</b>		RUN	interactive	n249-075-247	2025-10-09 12:03:10		1	2	0	vscode
13	<b>6144810</b>		RUN	interactive	n249-075-247	2025-10-09 14:41:23		1	2	0.6	vscode ./
14	<b>7508492</b>		RUN	interactive	n249-075-247	2025-10-11 16:11:18		1	2	0.4	vscode .
15	<b>8351120</b>		RUN	interactive	n249-075-247	2025-10-13 12:53:50		1	2	2.7	vscode --no-sandbox

说明：

- 点击任意列标题，可以排序列内容。
- 如果 job Rusage 没有设，或者 Rusage 小于 Mem (job 的实际内存用量) 的值，Mem 值背景色会变红。
- 点击 Job 列内容，可以直接跳转到 JOB 页，并展示 job 的信息。（粗体字均可点击跳转，后同）
- 点击 Status 列内容：
  - 如果 Status 为 RUN， bmonitor 会调用工具“Check Issue Reason”来查看 job SLOW 的原因。
  - 如果 Status 为 PEND， bmonitor 会调用工具“Check Issue Reason”来查看 job PEND 的原因。
  - 如果 Status 为 EXIT， bmonitor 会调用工具“Check Issue Reason”来查看 job FAIL 的原因。

## 4.2.6 HOSTS 页

HOSTS 页主要用于查看 hosts 的静态和动态信息。

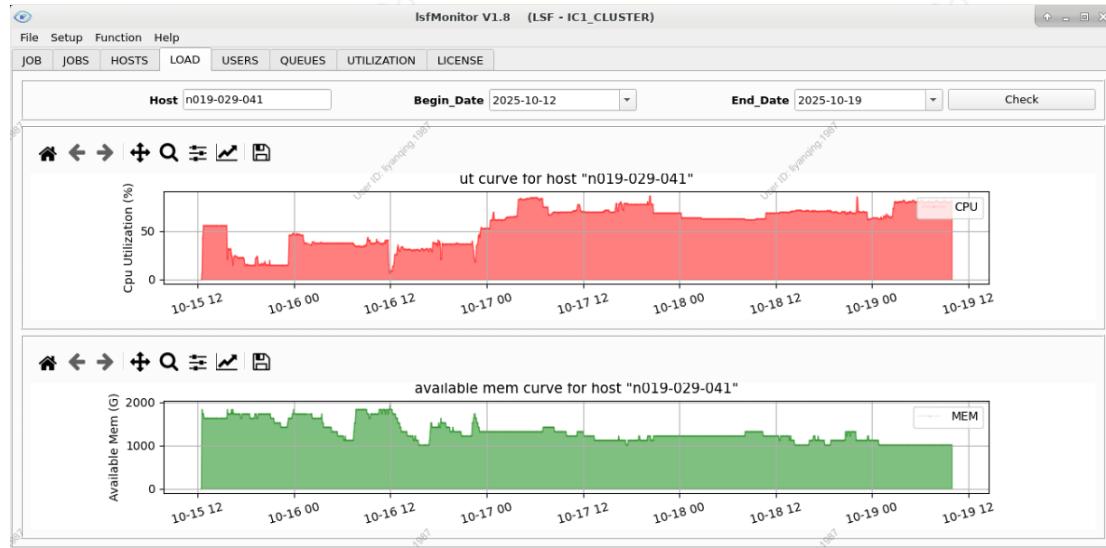
Host	Status	Queue	MAX	Njobs	Ut (%)	MaxMem (G)	aMem (G)	saMem (G)	MaxSwp (G)	Swp (G)	Tmp (G)
1 ic-lsf-main-m1	ok	closed_LIM	32	0	3	123	109	109	0	0	459
2 ic-lsf-main-m2	ok	closed_LIM	32	0	0	123	121	121	0	0	462
3 ic-lsf-main-m3	closed_Busy	closed_LIM	32	0	17	123	48	48	0	0	376
4 n019-028-021	ok	closed_LIM	120	112	24	992	443	338	127	127	1780
5 n019-028-024	ok	closed_LIM	57	21	2	958	933	314	127	127	1799
6 n019-028-025	ok	closed_LIM	57	20	2	958	932	334	127	127	1799
7 n019-028-026	ok	closed_LIM	57	21	2	958	927	314	127	127	1799
8 n019-028-027	ok	closed_LIM	57	20	2	958	933	324	127	127	1799
9 n019-028-028	ok	closed_LIM	57	21	5	958	910	313	127	127	1799
10 n019-028-029	ok	closed_LIM	57	22	2	958	903	284	127	127	1799
11 n019-028-030	ok	closed_LIM	57	20	3	958	887	323	127	127	1799
12 n019-028-031	ok	closed_LIM	57	21	1	958	929	193	127	127	1799
13 n019-028-032	ok	closed_LIM	57	20	2	958	924	224	127	127	1799
14 n019-028-033	ok	closed_LIM	57	20	1	958	934	243	127	127	1799
15 n019-028-036	ok	closed_LIM	96	9	0	744	728	491	127	127	1799

说明：

- 点击任意列标题，可以排序列内容。
- 如果 host 的 Status 异常 (unavail/unreach/closed\_LIM)，Status 状态背景色会变红。
- 如果 host 的 Ut 使用率超过 90%，Ut 值背景色会变红。
- 如果 host 的 aMem 或者 saMem 不足 MaxMem 的 10%，对应值的背景色会变红。
- 如果 host 的 tmp 可用量变为 0，Tmp 值背景色会变红。
- “aMem”是指系统的 available memory，即系统上实际的可用内存。
- “saMem”是指 scheduling available memory，即调度器所判断的可用内存，是 aMem 减去 rusage\_mem 之后的内存值。
- MaxMem  $\geq$  aMem  $\geq$  saMem
- 左击 Host 列内容，可以跳转到 LOAD 页，展示指定 host 的 cpu 和 memory 历史用量曲线。
- 右击 Host 列内容，可以弹出“Open”和“Close”两个选项，分别用于 hopen 和 hclose 当前 host，生效的前提是当前用户具有 LSF 管理员权限。
- 点击 Njobs 列内容，可以跳转到 JOBS 页，展示指定 host 上所有的 RUN jobs。

## 4.2.7 LOAD 页

LOAD 页主要用于查看 host 的 ut 和 memory 负载信息。



说明：

- Host 文本框支持模糊匹配，也可以从 HOSTS 页面点击 hostname 跳转过来。

## 4.2.8 USERS 页

USERS 页主要用于查看用户 job 相关的统计信息。

The table displays user statistics across various columns:

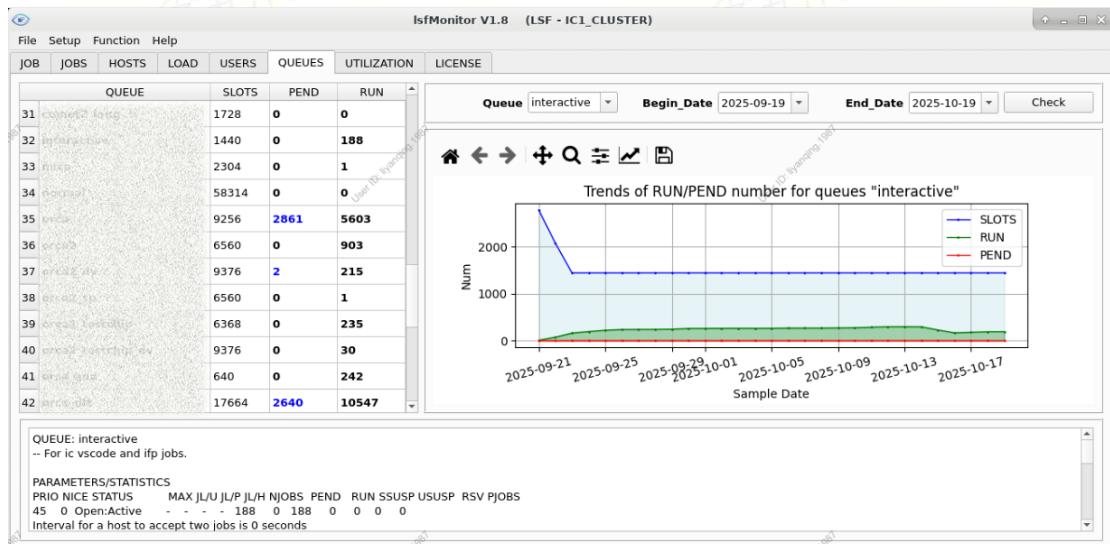
User	Job_Num	Pass_Rate (%)	Total_Rusage_Mem (G)	Avg_Rusage_Mem (G)	Total_Max_Mem (G)	Avg_Max_Mem (G)	Total_Mem_Waste (G)	Avg_Mem_Waste (G)
1 ALL	407927	96.0	4803375.0	11.8	857714.5	2.1	3945660.5	9.7
2	5	100.0	112.8	22.6	17.5	3.5	95.3	19.1
3	3510	82.9	43374.7	12.4	10295.2	2.9	33079.5	9.4
4	1	100.0	2.0	2.0	5.1	5.1	-3.1	-3.1
5	7	100.0	614.7	87.8	744.0	106.3	-129.3	-18.5
6	10	80.0	431.5	43.1	1.2	0.1	430.3	43.0
7	1	100.0	0.7	0.7	8.4	8.4	-7.7	-7.7
8	29	93.1	844.8	29.1	132.3	4.6	712.5	24.6
9	2	0.0	3.4	1.7	3.4	1.7	-0.0	-0.0
10	17	94.1	83.0	4.9	61.7	3.6	21.3	1.3
11	2	50.0	5.7	2.9	1.4	0.7	4.3	2.2
12	28	96.4	48.6	1.7	13.1	0.5	35.5	1.3
13	8	100.0	9.3	1.2	28.4	3.5	-19.1	-2.4
14	9	100.0	1113.3	123.7	24.4	2.7	1088.9	121.0

说明：

- 点击任意列标题，可以排序列内容。
- Job\_Num 是在指定时间段内用户所有 DONE/EXIT 任务的总量。
- Pass\_Rate 是在指定时间段内用户 DONE 任务数量/job\_Num 数量 的比值。
- Total\_Mem\_Waste 是在指定时间段内 Total\_Rusage\_Mem - Total\_Max\_Mem 的值，标识用户申请了但未使用而造成的内存浪费的量。

#### 4.2.9 QUEUES 页

QUEUES 页主要用于查看所有 queue 的实时和历史信息。

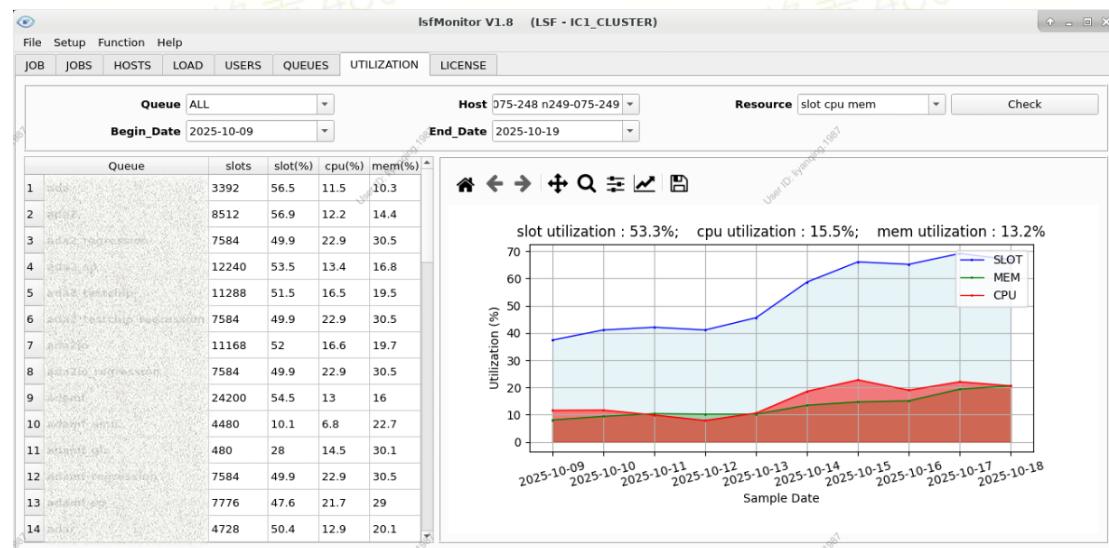


说明：

- 如果 queue 中 PEND 的 job 数目不为 0，数字会被红标。
- 点击 QUEUE 列内容，可以展示对应 queue 的详细信息和 queue 中 SLOTS/RUN/PEND 数据的变化曲线。
- 点击 PEND 列内容，可以跳转到 JOBS 页，展示指定 queue 上所有的 PEND jobs。(点击 RUN 列数字亦然)
- 在 Queue 下拉菜单中选择多个 queue，点击 Check 按钮，可以把多个 queues 的 SLOTS/PEND/RUN 信息累加展示。(对于共享队列，累加的 SLOTS 不具备参考意义)
- 如果想看更细致的数据，可以在菜单栏 Setup 中选中“Enable queue detail”选项，注意此时采样周期被默认重置为最近一周。

## 4.2.10 UTILIZATION 页

UTILIZATION 页主要用来查看 slot/cpu/memory 等资源的使用率统计信息，信息周期默认为最近一个月。



说明：

- 点击任意列标题，可以排序列内容。
- 点击 QUEUE 列的内容，可以展示 queue 的 slot/cpu/mem 使用率的变化曲线。
- 如果想看更细致的数据，可以在菜单栏 Setup 中选中“Enable utilization detail”选项，注意此时采样周期被默认重置为最近一周。
- 左侧和右侧的 utilization 统计值有可能会有所差别，尤其是在队列机器有变更（增加/减少）的情况，这是因为左侧结果是按照“sum(服务器利用率)/len(服务器数目)”计算出来的，右侧结果是按照“sum(整体按天汇聚利用率)/天数”计算出来的，理论上左侧表格中的使用率信息更准确一些。

## 4.2.11 LICENSE 页

LICENSE 页主要用于查看 EDA license 的使用情况。

启动 lsfMonitor 前，需要保证当前 terminal 中 LM\_LICENSE\_FILE 环境变量配置正确，bmonitor 将环境变量 LM\_LICENSE\_FILE 作为唯一的 license server 设定来源。

IsfMonitor V1.8 (LSF - IC1\_CLUSTER)

Feature Information					Expires Information				
Server	Vendor	Feature	Issued	In_Use	License Server	Feature	Num	Expires	
1	5280@ic-lic02	cdslmd	LicFileVersion	1	0	1	LicFileVersion	1	permanent(...)
2	5280@ic-lic02	cdslmd	Use_Server_Options	1	0	2	Use_Server_Options	1	permanent(...)
3	5280@ic-lic02	cdslmd	Affirma_sim_analysis_env	5	0	3	Affirma_sim_analysis_env	1	14-jul-2121
4	5280@ic-lic02	cdslmd	Allegro_Viewer_Plus	1	0	4	Affirma_sim_analysis_env	2	23-apr-2121
5	5280@ic-lic02	cdslmd	Allegro_performance	1	0	5	Affirma_sim_analysis_env	1	23-mar-2122
6	5280@ic-lic02	cdslmd	Allegro_studio	1	0	6	Affirma_sim_analysis_env	1	25-jul-2120
7	5280@ic-lic02	cdslmd	Base_Verilog_Lib	1	0	7	Allegro_Viewer_Plus	1	27-sep-2119
8	5280@ic-lic02	cdslmd	COSLITE_ACCESS	1	0	8	Allegro_performance	1	27-sep-2119
9	5280@ic-lic02	cdslmd	Concept_HDL_studio	1	0	9	Allegro_studio	1	27-sep-2119
10	5280@ic-lic02	cdslmd	Extended_Verilog_Lib	1	0	10	Base_Verilog_Lib	1	27-sep-2119
11	5280@ic-lic02	cdslmd	Integrated_Metrics_Center	5	0	11	COSLITE_ACCESS	1	23-mar-2122
12	5280@ic-lic02	cdslmd	PCB_design_studio	1	0	12	Concept_HDL_studio	1	27-sep-2119
13	5280@ic-lic02	cdslmd	Palladium_XP_HDL_Checking	1	0	13	Extended_Verilog_Lib	1	27-sep-2119
14	5280@ic-lic02	cdslmd	Palladium_Z1_Compile	2	0	14	Integrated_Metrics_Center	1	14-jul-2121

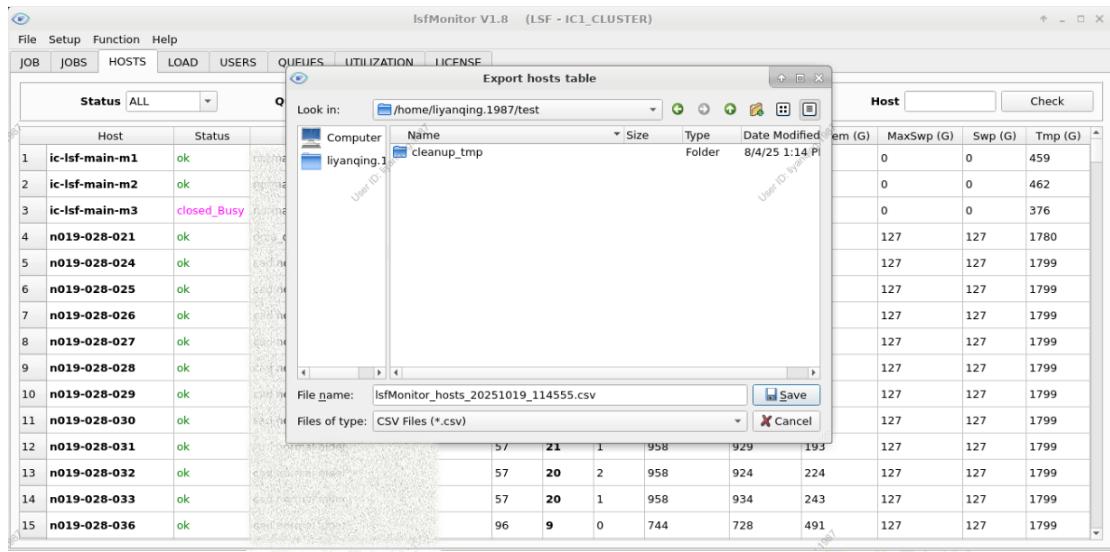
说明：

- 点击任意列标题，可以排序列内容。
- 其中 START\_TIME 启动时间在 3 天以前的，日期会标红。
- 右侧 Expires Information 表格中“Expires”列的内容，如果已过期，显示为灰色字体；如果两周内过期，显示为红色字体；如果未过期，显示为黑色字体。
- 左侧 Feature Information 表格中“In\_Use”列的内容，如果非零，左击可以弹出 license feature 的使用详情。
- 如果 conf/config.py 中 license\_administrator 配置不是“all”，且当前用户不在其中，则 LICENSE 页面不可见。
- 如果 conf/config.py 中 excluded\_license\_servers 配置了指定的 license server(s)，则指定 license server(s) 的信息会在 LICENSE 页面中被过滤掉。

## 4.2.12 其它功能介绍

### 4.2.12.1 保存表格

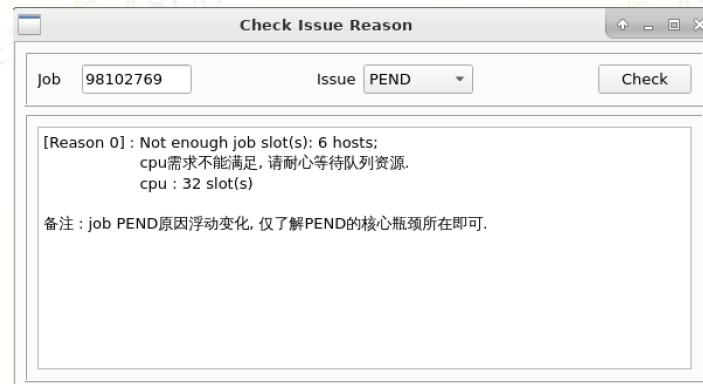
通过菜单栏 File -> Export \* table，可以将 bmonitor 中的任意表格保存为 csv 文件。



#### 4.2.12.2 检查 job PEND 的原因

以下两种方式可以触发“Check Issue Reason”工具来查找指定 job PEND 的原因：

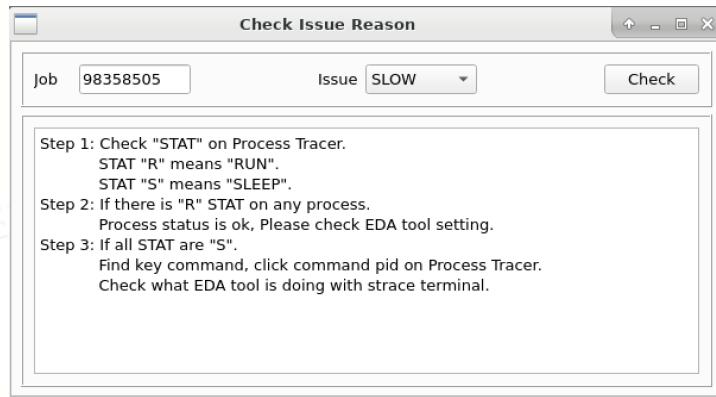
- 菜单栏 Function -> Check Pend reason。
- 点击 JOBS 页 Status 列的 PEND 单元格。



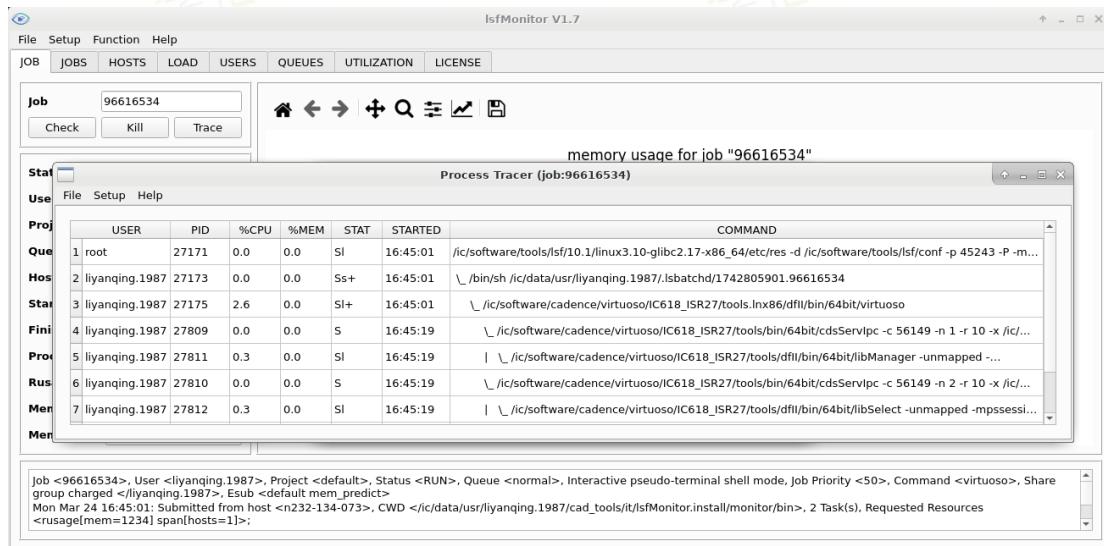
#### 4.2.12.3 检查 job 运行缓慢的原因

以下三种方式可以触发“Check Issue Reason”工具来查找指定 job 运行缓慢的原因：

- 菜单栏 Function -> Check Slow reason。
- JOB 页 job 的 Status 为 RUN 时, 点击“Trace”按钮。
- 点击 JOBS 页 Status 列的 RUN 单元格。



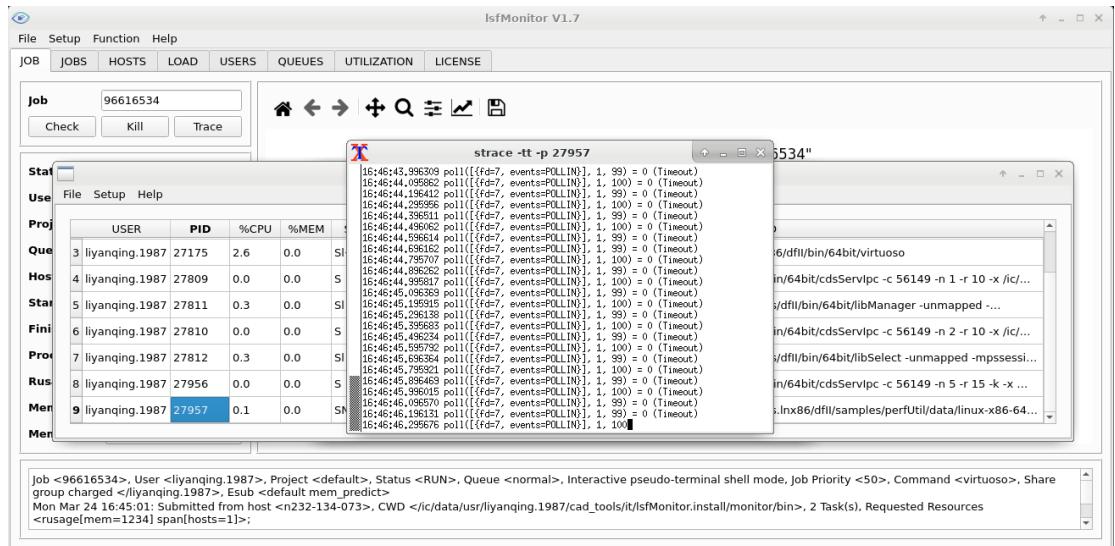
同时会弹出当前 job 的进程树信息。(一般普通用户只能 trace 自己的 job 进程树)  
(这个截图比较麻烦，直接偷懒用旧的截图了)



正常情况下应该至少有一个核心进程保持“R”的运行状态，如果所有的进程均为“S”的睡眠状态则可能意味着异常等待。

点击 PID 号，可以查看指定进程的系统交互信息，以方便判断进程运行缓慢的原因。

(这个截图比较麻烦，直接偷懒用旧的截图了)



#### 4.2.12.4 检查 job 非正常退出的原因

以下两种方式可以触发“Check Issue Reason”工具来查找指定 job 非正常退出的原因：

- 菜单栏 Function -> Check Pend reason。
- 点击 JOBS 页 Status 列的 EXIT 单元格。



#### 4.2.12.5 查看 EDA license feature 的使用详情

在 LICENSE 页，如果左侧“Feature Information”的“In\_Use”列的值不为零，那么点击“In\_Use”值可以展示当前 feature 的使用信息。

IsfMonitor V1.8 (LSF - IC1\_CLUSTER)

Server		USER	SUBMIT_HOST	EXECUTE_HOST	NUM	VERSION	START_TIME	JOB	Num	Expires
1	5280@ic-lic10		n019-028-201	n249-075-239	3	v22.100	Thu 10/16 2:06	9384027	1	permanent(...)
2	5280@ic-lic10		n019-028-197	n232-135-137	2	v22.100	Thu 10/16 2:06	10199145	1	permanent(...)
3	5280@ic-lic10		n019-028-197	n232-132-204	2	v22.100	Thu 10/16 2:06	10219327	1	14-jul-2121
4	5280@ic-lic10		n019-030-196	n232-129-023	3	v22.100	Thu 10/16 2:06	9212408	2	23-apr-2121
5	5280@ic-lic10		n019-028-196	n019-029-084	3	v22.100	Thu 10/16 2:06	10181369	1	23-mar-2122
6	5280@ic-lic10		n019-030-195	n232-132-077	3	v22.100	Thu 10/16 2:06	9956607	1	25-jul-2120
7	5280@ic-lic10		n019-028-194	n232-132-086	1	v22.100	Thu 10/16 2:06	*	1	27-sep-2119
8	5280@ic-lic10		n019-028-194	n232-132-025	3	v22.100	Thu 10/16 2:06	9737926	1	27-sep-2119
9	5280@ic-lic4		n019-028-194	n232-133-141	3	v22.100	Thu 10/16 2:06	8565251	1	27-sep-2119
10	5280@ic-lic10		n249-073-246	n232-135-138	1	v22.100	Thu 10/16 2:06	*	1	27-sep-2119
11	5280@ic-lic10		n249-073-242	n249-075-182	7	v23.100	Thu 10/16 2:06		1	23-mar-2122
12	5280@ic-lic02								1	27-sep-2119
13	5280@ic-lic10	cdslmd	VIP_AXI_ACE_5	490 89	13	5280@ic-lic02		Extended_Verilog_Lib	1	27-sep-2119
14	5280@ic-lic4	cdslmd	Palladium_Z3_Domain	288 56	14	5280@ic-lic02		Integrated_Metrics_Center	1	14-jul-2121

每一行为一个 check-in 记录，JOB 列记录着当前 check-in 行为对应的 job，如果 job 单元格为空说明 job 完成但 license 没释放，job 单元格为“\*”说明找到当前 check-in 找到多个可能对应的 job，无法确切判断。

## 五、辅助工具

lsmMonitor 自带一些工具，大多数是其功能组件，也有部分是可以单独使用的工具。

```
[root@ic-admin01 lsfMonitor]# ls monitor/tools/
akill      check_issue_reason    lmstat      patch
process_tracer   seedb      show_license_feature_usage
akill.py  check_issue_reason.py message.py  patch.py
process_tracer.py seedb.py  show_license_feature_usage.py
```

**akill**: 独立工具，是 bkill 的增强型功能。

**check\_issue\_reason**: 组件，图形界面工具，查看 job PEND/FAIL/SLOW 的原因。

**lmsta**: 独立工具，用于检索 EDA license 信息。

**message.py**: 组件，图形界面工具，显示指定信息。

**patch**: 独立工具，用于更新工具安装包。

**process\_tracer**: 组件，图形界面工具，追踪指定 process 或 jobid 的进程树。

**seedb**: 独立工具，查看 sqlite3 数据库内容，lsmMonitor 即使用 sqlite3 保存相关数据。

**show\_license\_feature\_usage**: 组件，图形界面工具，查看 feature 的使用情况。

### 5.1 akill

akill 是 bkill 的增强型工具，位于安装目录下的 monitor/tools/akill，可以根据 jobid/job\_name/command/submit\_time/execute\_host/queue/user 等维度来便捷地 kill job，其帮助信息如下。

```
[root@ic-admin01 lsfMonitor]# monitor/tools/akill -h
usage: akill.py [-h] [-j JOBID [JOBID ...]] [-J JOB_NAME
[JOB_NAME ...]] [-c COMMAND [COMMAND ...]] [-s SUBMIT_TIME
[SUBMIT_TIME ...]] [-m EXECUTE_HOST [EXECUTE_HOST ...]]
[-q QUEUE [QUEUE ...]] [-u USER [USER ...]]

optional arguments:
  -h, --help            show this help message and exit
  -j JOBID [JOBID ...], --jobid JOBID [JOBID ...]
                        kill specified job(s) based on jobid(s),
                        support fuzzy matching, also support jobid range like "10200-
                        10450".
```

```
-J JOB_NAME [JOB_NAME ...], --job_name JOB_NAME [JOB_NAME ...]
          kill specified job(s) based on
job_name(s), support fuzzy matching.
-c COMMAND [COMMAND ...], --command COMMAND [COMMAND ...]
          kill specified job(s) based on command(s),
support fuzzy matching.
-s SUBMIT_TIME [SUBMIT_TIME ...], --submit_time SUBMIT_TIME
[SUBMIT_TIME ...]
          kill specified job(s) based on
submit_time(s), support fuzzy matching.
-m EXECUTE_HOST [EXECUTE_HOST ...], --execute_host EXECUTE_HOST
[EXECUTE_HOST ...]
          kill specified job(s) based on execute
host(s).
-q QUEUE [QUEUE ...], --queue QUEUE [QUEUE ...]
          kill specified job(s) based on queue(s).
-u USER [USER ...], --user USER [USER ...]
          kill specified job(s) based on user(s).
```

akill 更详细的用法, 请参照 lsfMonitor 安装目录下的 docs/akill\_user\_manual.pdf。

## 5.2 patch

patch 是帮助 lsfMonitor 打补丁的工具, 其帮助信息如下。

```
[root@ic-admin01 lsfMonitor]# monitor/tools/patch -h
usage: patch.py [-h] [-p PATCH_PATH]

optional arguments:
  -h, --help            show this help message and exit
  -p PATCH_PATH, --patch_path PATCH_PATH
                        Specify patch path (new install package
path).
```

**--patch\_path:** 指定补丁包（也就是新的安装包）路径。

一般而言, lsfMonitor 的版本变更, 主要是新增 python 脚本, 或者是现有 python 脚本内容变更, 针对这种变更的安装包, patch 可以自动将新的安装包变更更新到当前安装包。

下面是一个示例。

```
[root@ic-admin01 lsfMonitor]# monitor/tools/patch -p  
/ic/data/usr/liyanqing.1987/test/lsfMonitor-master  
Install Path : /ic/data/usr/liyanqing.1987/lsfMonitor  
Patch path : /ic/data/usr/liyanqing.1987/lsfMonitor-master  
  
*Warning*: current install path name is "lsfMonitor", but patch  
path name is "lsfMonitor-master".  
Do you want to continue? (y|n) y  
  
> Copying python file "/ic/data/usr/liyanqing.1987/lsfMonitor-  
master/monitor/bin/bsample.py" into  
"/ic/data/usr/liyanqing.1987/test/lsfMonitor/monitor/bin/bsample.p  
y".  
> Copying python file "/ic/data/usr/liyanqing.1987/lsfMonitor-  
master/monitor/common/common_license.py" into  
"/ic/data/usr/liyanqing.1987/test/lsfMonitor/monitor/common/common  
_license.py".  
> Copying python file "/ic/data/usr/liyanqing.1987/lsfMonitor-  
master/monitor/tools/seedb.py" into  
"/ic/data/usr/liyanqing.1987/test/lsfMonitor/monitor/tools/seedb.p  
y".  

```

## 5.3 seedb

seedb 是查看 sqlite3 文本数据库内容的工具，其帮助信息如下：

```
[root@ic-admin01 lsfMonitor]# monitor/tools/seedb -h  
usage: seedb.py [-h] -d DATABASE [-t TABLES [TABLES ...]] [-k KEYS  
[KEYS ...]] [-n NUMBER]  
  
optional arguments:  
-h, --help            show this help message and exit
```

```
-d DATABASE, --database DATABASE
                                Required argument, specify the database
file.

-t TABLES [TABLES ...], --tables TABLES [TABLES ...]
                                Specify the tables you want to review,
make sure the tables exist.

-k KEYS [KEYS ...], --keys KEYS [KEYS ...]
                                Specify the table keys you want to review,
make sure the table keys exist.

-n NUMBER, --number NUMBER
                                How many lines you want to see.
```

**--database**: 指定 sqlite3 数据库文件。

**--tables**: 指定想查看的数据库 tables。

**--keys**: 指定想查看的 table 关键词。

**--number**: 指定查看的行数，如果内容很多，可以截取有限的内容查看。

示例一，查看 load.db 数据库中的表。

```
[root@ic-admin01 lsfMonitor]# monitor/tools/seedb -d
/ic/software/cad_data/it/lsfMonitor/db/monitor/load.db
DB_FILE : /ic/software/cad_data/it/lsfMonitor/db/monitor/load.db
TABLES  :
=====
load_ic-hpc-mon02
load_ic-lsfmaster1
load_ic-lsfmaster2
...
=====
```

示例二，查看 load.db 数据库中指定的表 (load\_n212-206-211) 的内容。

```
[root@ic-admin01 lsfMonitor]# monitor/tools/seedb -d
/ic/software/cad_data/it/lsfMonitor/db/monitor/load.db -t
load_n212-206-211
DB_FILE : /ic/software/cad_data/it/lsfMonitor/db/monitor/load.db
TABLE   : load_n212-206-211
=====
sample_second          sample_time          ut
```

tmp	swp	mem
---	---	---
---	---	---
1683984602	20230513_213002	0%
1671G	252.8G	773G
1683984902	20230513_213502	0%
1671G	252.8G	809G
...		
=====		

示例三，查看 load.db 数据库中指定的表 (load\_n212-206-211) 的指定列 (mem) 的内容。

```
[root@ic-admin01 lsfMonitor]# monitor/tools/seedb -d
/ic/software/cad_data/it/lsfMonitor/db/monitor/load.db -t
load_n212-206-211 -k mem
DB_FILE : /ic/software/cad_data/it/lsfMonitor/db/monitor/load.db
TABLE   : load_n212-206-211
=====
mem
-----
773G
809G
845G
886G
903G
...
=====
```

示例四，查看 load.db 数据库中指定的表 (load\_n212-206-211) 的指定列 (mem) 的内容，只看前三行。

```
[root@ic-admin01 lsfMonitor]# monitor/tools/seedb -d
/ic/software/cad_data/it/lsfMonitor/db/monitor/load.db -t
load_n212-206-211 -k mem -n 3
DB_FILE : /ic/software/cad_data/it/lsfMonitor/db/monitor/load.db
TABLE   : load_n212-206-211
=====
mem
-----
773G
```

```
809G  
845G  
=====
```

## 5.4 memPrediction

memPrediction 是 lsfMonitor 内置的一个 LSF 任务 memory 资源分析和自动预设工具，可以用来分析 LSF cluster 中用户 memory/cpu 使用状况，也可以通过 LSF esub 机制自动为未预设内存的 job 自动设置 rusage mem 的值。

lsfMonitor 安装时会自动安装 memPrediction，但是不会配置和启用，如果需要启用，具体使用方法请参照 lsfMonitor 安装目录下的 memPrediction 工具用户文档 docs/memPrediction\_user\_manual.pdf。

## 六、lsfMonitor 常见问题及解决

### 6.1 图形显示问题

问题描述：

安装后 bmonitor 不显示图形界面，或者图形界面显示不全、显示效果异常。

问题原因：

- \* 使用的 python 版本并非 3.12.12 或者兼容版本。
- \* python 库安装不全。

解决方案：

使用推荐的 python3.12.12，按照 3.2 章节的方法安装 requirements.txt 的 python 依赖库。

### 6.2 JOBS 页信息缺失

问题描述：

JOBS 页部分信息缺失。

问题原因：

- \* 使用的非兼容版本的 openlava。
- \* 使用的版本过老的 LSF（比如 9.1.2 或者更老的版本）。

解决方案：

使用推荐版本的 LSF/volclava/openlava。

### 6.3 LICENSE 页信息缺失

问题描述：

LICENSE 页不显示有效的 license 信息。

问题原因：

- \* bmonitor 启动的 terminal 没有配置环境变量 LM\_LICNSE\_FILE。
- \* monitor/conf/config.py 中 lmstat\_bsub\_command 变量配置错误。（常见）

解决方案：

\* 确认 bmonitor 启动的 terminal 中已经配置好正确的环境变量 LM\_LICENSE\_FILE。

\* 如果当前机器允许执行 EDA 工具（可运行 lmstat），那么将 monitor/conf/config.py 中 lmstat\_bsub\_command 变量配置为空，否则设置合适的 bsub 命令。

## 6.4 HOSTS 页和 LOAD 页中的 mem 值为什么不一致

问题描述：

对同一台 server，就当前时刻而言，在 HOSTS 页上看到的 Mem 值和在 LOAD 页中看到的“available mem”值不一致，HOSTS 中看到的值往往偏小。

问题原因：

HOSTS 页中的 Mem 值跟 LOAD 页中的 mem 值信息来源不一样，作用也不一样。

- HOSTS 页，mem 信息来源于“bhosts -l”，显示的是这台机器上实际可以被 reserve 的 mem，已经把被用的和被 reserve 的都排除在外了，所以值会偏小。（用途是判断机器还能否接受 job）
- LOAD 页，mem 的信息来源是“lsload”，显示的是这台机器上真实的剩余 mem 信息，不考虑 rusage 的 mem。（用途是判断机器真实的 mem 用量，判断是否会发生 OOM）

## 七、技术支持

本工具为开源工具，由开源社区维护，可以提供如下类型的技术支持：

- 部署和使用技术指导。
- 接收 bug 反馈并修复。
- 接收功能修改建议。(需审核和排期)

获取技术支持的方式包括：

- 通过 Contact 邮箱联系开发者。
- 添加作者微信 “liyanqing\_1987”，注明“真实姓名/公司/lstMonitor”，由作者拉入技术支持群。



# 附录

## 附 1. 变更历史

日期	版本	变更描述	备注
2017	1.0	发布第一个版本 openlavaMonitor, 仅支持 openlava。	
2020	1.1	更名 lsfMonitor, 增加对 LSF 的支持。	
2022	1.2	增加 LICENSE 信息的采集和展示。	
2023.05	1.3	<ul style="list-style-type: none"><li>增加 UTILIZATION 信息的采集和展示。</li><li>增加 patch 工具。</li><li>优化数据库数据格式。</li></ul>	数据库格式跟旧版本不兼容, 需重新安装。
2023.06	1.3.1	<ul style="list-style-type: none"><li>优化 utilization 数据采样和取用方式, 采样时汇聚。</li><li>优化 EDA license 信息采样操作, 通过 multi-process 方式并行大幅缩短采样时间。</li></ul>	
2023.06	1.3.2	<ul style="list-style-type: none"><li>HOSTS 页增加按照 Status/MAX/MaxMem/Host 选择 hosts 的过滤功能。</li><li>LICENSE 页增加按照 User 选择 license feature 的过滤功能。</li></ul>	
2023.09	1.3.3	<ul style="list-style-type: none"><li>show_license_feature_usage 页面显示 feature 跟 job 的关联关系。</li><li>增加工具 akill, 帮助更快捷地 kill job。</li><li>Check 按钮自动触发 LSF/LICENSE 信息的重新采集, 保证刷新功能。</li></ul>	

2023.11	1.4	<ul style="list-style-type: none"> <li>所有的单选下拉框均改为复选下拉框，支持多选。</li> <li>QUEUES 及 UTILIZATION 页面，支持选择更细力度的数据曲线。</li> <li>QUEUES 页面支持选择数据曲线日期范围。</li> <li>UTILIZATION 页面增加 queues 的 utilization 统计汇总信息。</li> </ul>	queue.db 数据格式跟旧版本不兼容，需要删掉旧的 db 文件。
2023.12	1.4.1	<ul style="list-style-type: none"> <li>增加工具 logo 和菜单栏项目图标。</li> <li>支持所有的表格导出。</li> <li>曲线图显示效果优化。</li> </ul>	
2024.03	1.4.2	<ul style="list-style-type: none"> <li>增加对多 LSF/openlava cluster 的支持。</li> </ul>	db_path 会自动做一次切换。
2024.06	1.5	<ul style="list-style-type: none"> <li>增加图形界面尺寸自适应。</li> <li>增加文本自动补全功能。</li> <li>在 JOB 和 JOBS 页增加 kill job 功能。</li> <li>在 JOBS 页和 HOSTS 页增加右键菜单。</li> <li>在 Check Fail Reason 功能中增加退出码和退出原因的明确对应。</li> <li>将表格输出格式由 Excel 切换为 csv 文件。 (避免 Excel 65536 行的限制)</li> <li>增加任务内存分析和预设工具 memPrediction。</li> </ul>	
2024.08	1.5.1	<ul style="list-style-type: none"> <li>bsample 增加 DONE/EXIT job 信息采样功能，保存为 json 文件。</li> <li>memPrediction 增加 bsample 采样 json 作为 training 数据源。</li> <li>修复 job 信息采样中 MEM 信息获取的 bug。</li> </ul>	db_path 中的 job 目录切换为 job_mem 目录，job 目录改存 job 历史信息。

2024.09	1.6	<ul style="list-style-type: none"> <li>bsample 中修改 user 采样信息内容。</li> <li>bmonitor 新增 USER 页，展示用户 job 的汇总信息。</li> <li>bmonitor 增加 dark_mode，面板可以选择亮色系或者暗色系。</li> <li>支持 ByteDance 开源的 LSF 兼容调度器 volclava。</li> </ul>	db_path 中新增 user 目录。
2025.03	1.7	<ul style="list-style-type: none"> <li>优化 bsample，加快采样速度。</li> <li>增加对 queue 信息中“HOSTS: &lt;host_group&gt;+&lt;num&gt;”写法的支持。</li> <li>HOSTS 页 Mem 项拆分为 aMem（系统可用内存）和 saMem（调度可用内存）。</li> <li>conf/config.py 中增加参数 “excluded_license_servers”，用于排除指定的 license servers。</li> </ul>	
2025.10	1.8	<ul style="list-style-type: none"> <li>标题栏增加 scheduler - cluster 信息的展示。</li> <li>LOAD 页面，Host 下拉单选框改为支持模糊匹配的文本框。</li> <li>QUEUES 页面，增加 Queue 复选框，可以汇聚多 queue 的 SLOTS/PEND/RUN 历史用量信息。</li> <li>复选框增加模糊匹配功能。</li> <li>config.py 中增加 license_administrators 参数，仅指定的管理员可以看到 LICENSE 页。</li> </ul>	
2026.01	2.0	<ul style="list-style-type: none"> <li>python 支持版本从 3.8.8 切换到 3.12.12。</li> <li>为工具启动和退出增加 log 记录。</li> <li>根据 AI 建议做了部分代码 enhancement。</li> </ul>	支持 python 版本切换，旧版本工具建议重新安装

备注：小的 hotfix 不计入变更历史，bugfix 会实时 checkin 到 github 上。

## 附 2. LSF 任务 exit code 含义

- '-9': 通常表示作业被强制终止
- '-8': 通常表示资源严重不足
- '-7': 通常表示无效的主机名或节点
- '-6': 通常表示作业被系统强制终止
- '-5': 通常表示作业被用户终止或中断
- '-4': 通常表示无效的队列名称
- '-3': 通常表示无效的作业 ID
- '-2': 通常表示内部错误
- '-1': 通常表示系统级错误
- '0': 成功完成并正常退出
- '1': 通常表示一般性的警告性错误
- '2': 通常表示一般性的错误
- '3': 通常表示一般性的致命错误
- '4': 通常表示初始化失败
- '5': 通常表示输入/输出错误
- '6': 通常表示无效的参数
- '7': 通常表示无法分配足够的内存
- '8': 通常表示目录不存在或无法访问
- '9': 通常表示文件不存在或无法访问
- '10': 通常表示不支持的功能或操作
- '11': 通常表示文件已存在
- '12': 通常表示超时错误
- '13': 通常表示权限被拒绝
- '14': 通常表示资源不足
- '15': 通常表示信号中断
- '16': 通常表示管道损坏或被关闭
- '17': 通常表示死锁错误
- '18': 通常表示磁盘已满

- '19': 通常表示读取错误
- '20': 通常表示写入错误
- '21': 通常表示连接错误
- '22': 通常表示无效或损坏的数据
- '23': 通常表示操作被中止
- '24': 通常表示过多打开的文件
- '25': 通常表示操作不可行
- '26': 通常表示无效的文件系统操作
- '27': 通常表示文件太大, 超出限制
- '28': 通常表示信号已被阻塞
- '29': 通常表示管道已满或破裂
- '30': 通常表示无效的进程
- '31': 通常表示在操作中断时出现错误
- '32': 通常表示某个命令的语法错误
- '33': 通常表示触发某个限制或配额
- '34': 通常表示某个服务或进程的异常终止
- '35': 通常表示输入或输出的格式错误
- '36': 通常表示某个进程或作业已经在运行
- '37': 通常表示某个进程或作业已经停止或结束
- '38': 通常表示操作被取消或中断
- '39': 通常表示需要更高的权限来执行操作
- '40': 通常表示某个连接或会话已经关闭
- '41': 通常表示接收到无效或损坏的数据
- '42': 通常表示某个网络连接失败
- '43': 通常表示某个网络服务不可用
- '44': 通常表示某个数据库操作失败
- '45': 通常表示某个文件或目录已经损坏
- '46': 通常表示某个操作已超时
- '47': 通常表示安全验证失败

- '48': 通常表示发生未知的错误
- '49': 通常表示某个任务或处理已中断
- '50': 通常表示某个进程或作业达到资源限制
- '51': 通常表示故障引起的不可恢复的错误
- '52': 通常表示某个进程被非法访问或操作
- '53': 通常表示某个操作被有效的权限限制
- '54': 通常表示某个系统服务或组件不可用
- '55': 通常表示某个数据损坏或丢失
- '56': 通常表示某个网络连接已经超时
- '57': 通常表示某个数据库连接失败
- '58': 通常表示某个脚本或程序操作失败
- '59': 通常表示某个任务被取消或终止
- '60': 通常表示某个进程或作业已过期
- '61': 通常表示某个配置错误导致无法正常执行
- '62': 通常表示某个日志文件错误
- '63': 通常表示某个加密或解密操作失败
- '64': 通常表示某个进程或作业运行时间过长
- '65': 通常表示与某个硬件设备的通信失败
- '66': 通常表示某个数据库查询操作失败
- '67': 通常表示某个网络协议错误
- '68': 通常表示某个文件系统操作失败
- '69': 通常表示某个环境变量未设置或无效
- '70': 通常表示某个进程或作业的使用量超过限制
- '71': 通常表示某个数据库连接超时
- '72': 通常表示某个配置文件错误
- '73': 通常表示某个依赖项未满足
- '74': 通常表示某个加密或解密密钥无效
- '75': 通常表示某个消息传递失败
- '76': 通常表示某个网络通信错误

- '77': 通常表示某个文件或目录已损坏
- '78': 通常表示某个版本错误
- '79': 通常表示某个资源不可用
- '80': 通常表示某个任务超时
- '81': 通常表示某个链接无效或过期
- '82': 通常表示某个权限被拒绝
- '83': 通常表示某个输入无效或错误
- '84': 通常表示某个输出无效或错误
- '85': 通常表示某个连接被重置
- '86': 通常表示某个数据库事务失败
- '87': 通常表示某个网络服务超负荷
- '88': 通常表示某个文件被锁定
- '89': 通常表示某个配置项缺失或无效
- '90': 通常表示某个配置文件丢失或损坏
- '91': 通常表示某个请求被限制或阻止
- '92': 通常表示某个协议错误
- '93': 通常表示某个验证失败
- '94': 通常表示某个数据传输错误
- '95': 通常表示某个脚本或程序非法操作
- '96': 通常表示某个任务被其他任务阻塞
- '97': 通常表示某个命令已过时或不再支持
- '98': 通常表示某个外部资源不可达
- '99': 通常表示某个时间戳无效或过期
- '100': 通常表示某个进程或作业已经终止
- '101': 通常表示某个协议切换失败
- '102': 通常表示某个网络连接已被废弃
- '103': 通常表示某个配置文件格式错误
- '104': 通常表示某个请求被重定向
- '105': 通常表示某个数据库操作异常

- '106': 通常表示某个加密或解密错误
- '107': 通常表示某个脚本或程序运行环境错误
- '108': 通常表示某个资源限制被超出
- '109': 通常表示某个输入输出错误
- '110': 通常表示某个网络连接超载
- '111': 通常表示某个进程或作业被阻塞
- '112': 通常表示某个命令执行失败
- '113': 通常表示某个服务不可达
- '114': 通常表示某个证书无效或过期
- '115': 通常表示某个文件系统错误
- '116': 通常表示某个资源被释放或移除
- '117': 通常表示某个地址被禁止访问
- '118': 通常表示某个任务被挂起或暂停
- '119': 通常表示某个操作系统错误
- '120': 通常表示某个协议版本错误
- '121': 通常表示某个文件或目录不存在
- '122': 通常表示某个资源临时不可用
- '123': 通常表示某个命令语法错误
- '124': 通常表示某个日志记录失败
- '125': 通常表示某个编码或解码错误
- '126': 通常表示某个执行权限不足
- '127': 通常表示某个命令未找到
- '128': 通常表示某个无效的退出状态
- '129': 通常表示某个库依赖项错误
- '130': 通常表示某个进程或作业被中断
- '131': 通常表示某个信号捕获失败
- '132': 通常表示某个文件被修改
- '133': 通常表示某个连接被拒绝
- '134': 通常表示某个堆栈溢出

- '135': 通常表示某个资源超时
- '136': 通常表示某个内存分配错误
- '137': 通常表示某个进程或作业因超出资源限制而被杀死
- '138': 通常表示某个信号超出范围
- '139': 通常表示某个分段错误
- '140': 通常表示某个程序或进程收到了致命信号
- '141': 通常表示某个时钟错误
- '142': 通常表示某个文件格式无效
- '143': 通常表示某个程序被终止或中断
- '144': 通常表示某个信号被阻止
- '145': 通常表示某个操作被终止
- '146': 通常表示某个目录切换失败
- '147': 通常表示某个管道错误
- '148': 通常表示某个孤儿进程（没有父进程）
- '149': 通常表示某个挂起的进程或作业
- '150': 通常表示某个资源耗尽
- '151': 通常表示某个镜像损坏或无效
- '152': 通常表示某个文件被锁定
- '153': 通常表示某个内存映射错误
- '154': 通常表示某个信号处理失败
- '155': 通常表示某个网络操作失败
- '156': 通常表示某个设备驱动错误
- '157': 通常表示某个套接字连接错误
- '158': 通常表示某个链接超时
- '159': 通常表示某个文件描述符无效
- '160': 通常表示某个插件或扩展错误
- '161': 通常表示某个远程主机不可达
- '162': 通常表示某个文件读取错误
- '163': 通常表示某个文件写入错误

- '164': 通常表示某个数据包损坏
- '165': 通常表示某个数据库连接错误
- '166': 通常表示某个超出容量限制
- '167': 通常表示某个死锁状态
- '168': 通常表示某个证书验证失败
- '169': 通常表示某个系统时钟漂移
- '170': 通常表示某个正在进行的操作被取消
- '171': 通常表示某个权限不允许操作
- '172': 通常表示某个目标不可到达
- '173': 通常表示某个文件系统不支持操作
- '174': 通常表示某个系统服务异常
- '175': 通常表示某个网络地址不可用
- '176': 通常表示某个资源已经存在
- '177': 通常表示某个数据内容被篡改
- '178': 通常表示某个系统崩溃或故障
- '179': 通常表示某个进程或作业超时
- '180': 通常表示某个用户操作中止
- '181': 通常表示某个文件被加密
- '182': 通常表示某个库文件损坏
- '183': 通常表示某个账户权限不足
- '184': 通常表示某个资源被占用
- '185': 通常表示某个数据格式错误
- '186': 通常表示某个网络连接已关闭
- '187': 通常表示某个缺少依赖项
- '188': 通常表示某个系统日志错误
- '189': 通常表示某个命令行参数错误
- '190': 通常表示某个配置文件缺失或损坏
- '191': 通常表示某个文件权限错误
- '192': 通常表示某个进程或作业已经在运行

- '193': 通常表示某个设备或服务不可用
- '194': 通常表示某个数据验证错误
- '195': 通常表示某个协议操作失败
- '196': 通常表示某个系统初始化错误
- '197': 通常表示某个备份或恢复错误
- '198': 通常表示某个连接被重置
- '199': 通常表示某个程序逻辑错误
- '200': 通常表示某个成功的状态, 表示程序或进程成功终止
- '201': 通常表示某个一个请求被响应成功
- '202': 通常表示某个异步操作已经启动结果可能在以后的时间返回
- '203': 通常表示某个接受已接收的请求但未执行
- '204': 通常表示某个请求成功执行, 但没有返回实体内容
- '205': 通常表示某个请求成功执行后, 需要发送新的请求才能获取更新的页面
- '206': 通常表示某个请求成功执行, 但只返回了部分内容
- '207': 通常表示某个多状态响应, 用于表示多个操作的结果
- '208': 通常表示某个请求执行的结果已经包含在响应消息中
- '209': 通常表示某个请求成功执行, 但返回的信息已经被代理修改
- '210': 通常表示某个请求成功执行, 但返回的资源已经移动
- '211': 通常表示某个请求成功执行, 但返回的内容已更改
- '212': 通常表示某个请求成功执行, 但需要额外的身份验证
- '213': 通常表示某个请求成功执行, 但返回的数据部分已失效
- '214': 通常表示某个请求成功执行, 但没有满足请求的结果
- '215': 通常表示某个请求成功执行, 但需要进一步处理
- '216': 通常表示某个请求成功执行, 但存在范围不匹配的内容
- '217': 通常表示某个请求成功执行, 但返回的数据类型不受支持
- '218': 通常表示某个请求成功执行, 但返回的数据已经已知
- '219': 通常表示某个请求成功执行, 但存在内部冲突
- '220': 通常表示某个请求成功执行, 但进程或操作处于暂停状态
- '221': 通常表示某个请求成功执行, 但返回的内容未更改

'222': 通常表示某个请求成功执行, 但返回的内容已被重新排序

'223': 通常表示某个请求成功执行, 但返回的内容未被重置

'224': 通常表示某个请求成功执行, 但存在版本冲突

'225': 通常表示某个请求成功执行, 但存在服务器过载

'226': 通常表示某个请求成功执行, 但返回的内容是通过协商缓存生成的

'227': 通常表示某个请求成功执行, 但源数据存在问题

'228': 通常表示某个请求成功执行, 但数据解析失败

'229': 通常表示某个请求成功执行, 但存在内容更新冲突

'230': 通常表示某个请求成功执行, 但返回的内容包含警告信息

'231': 通常表示某个请求成功执行, 但返回的内容需要用户进一步处理

'232': 通常表示某个请求成功执行, 但返回结果需要进一步验证

'233': 通常表示某个请求成功执行, 但返回的内容需要管理员干预

'234': 通常表示某个请求成功执行, 但存在资源限制

'235': 通常表示某个请求成功执行, 但返回的内容包含敏感信息

'236': 通常表示某个请求成功执行, 但返回的内容包含过期信息

'237': 通常表示某个请求成功执行, 但返回的内容存在冲突

'238': 通常表示某个请求成功执行, 但返回的内容被转换

'239': 通常表示某个请求成功执行, 但服务端要求重新认证

'240': 通常表示某个请求成功执行, 但需要进行重定向

'241': 通常表示某个请求成功执行, 但存在过多的重定向

'242': 通常表示某个请求成功执行, 但返回的内容需要进行安全验证

'243': 通常表示某个请求成功执行, 但返回的内容需要进行数据转换

'244': 通常表示某个请求成功执行, 但存在连接超时

'245': 通常表示某个请求成功执行, 但返回的内容已过期

'246': 通常表示某个请求成功执行, 但返回的内容需要进行语义解析

'247': 通常表示某个请求成功执行, 但返回的内容需要进行格式转换

'248': 通常表示某个请求成功执行, 但返回的内容需要进行编码转换

'249': 通常表示某个请求成功执行, 但返回的内容需要进行内容裁剪

'250': 通常表示某个请求成功执行, 但返回的内容需要进行内容合并

'251': 通常表示某个请求成功执行, 但返回的内容需要进行内容过滤

'252': 通常表示某个请求成功执行, 但返回的内容需要进行内容排序

'253': 通常表示某个请求成功执行, 但返回的内容需要进行内容分割

'254': 通常表示某个请求成功执行, 但返回的内容需要进行内容聚合

'255': 通常表示命令因不明原因执行失败