

# IsfMonitor 用户手册

**Product Name :** IsfMonitor

**Product Version :** V1.3

**Release Date :** 2023.5

**Contact :** @李艳青 (liyanqing1987@163.com)

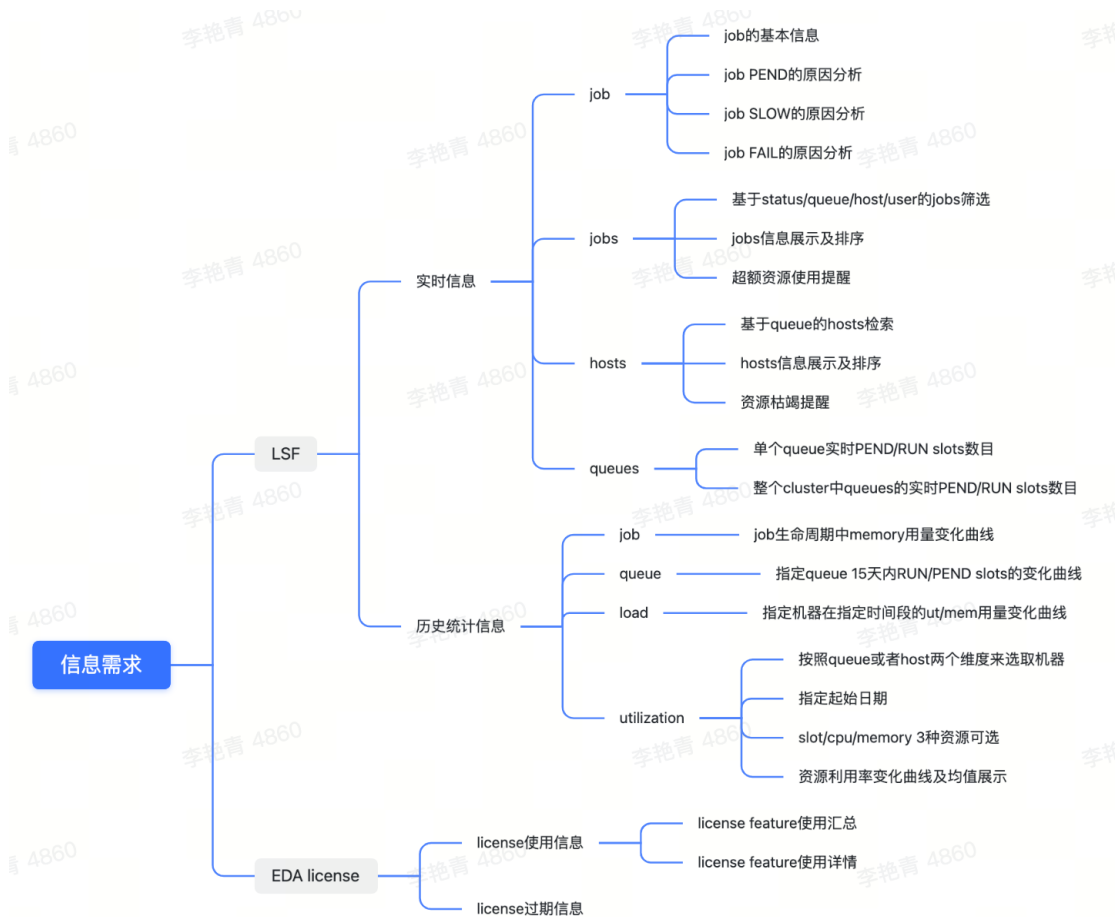
## 目录

一、简介	3
二、环境依赖	4
2.1 操作系统依赖	4
2.2 python 版本依赖	4
2.3 集群管理工具	4
三、工具安装及配置	5
3.1 工具下载	5
3.2 工具安装	6
3.3 工具配置	7
四、工具使用	9
4.1 数据采集 <b>bsample</b>	9
4.2 数据展示 <b>bmonitor</b>	10
4.2.1 工具载入	10
4.2.2 帮助信息	10
4.2.3 JOB 页	11
4.2.4 JOBS 页	14
4.2.5 HOSTS 页	15
4.2.6 QUEUES 页	16
4.2.7 LOAD 页	17
4.2.8 UTILIZATION 页	17
4.2.9 LICENSE 页	18
4.2.10 其它功能	21
五、辅助工具	23
5.1 seedb	23
5.2 patch	24
六、技术支持	27
附录	28
附 1. 变更历史	28

# 一、简介

IsfMonitor 是一款用于 LSF/openlava 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，可以满足集成电路设计用户对于 LSF/license 的绝大部分信息查询和常规问题解决需求。

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：



IsfMonitor 就是为如上信息需求提供一站式解决方案。

## 二、环境依赖

### 2.1 操作系统依赖

IsfMonitor 的开发和测试操作系统为 **CentOS Linux release 7.9.2009 (Core)**，这也是 IC 设计常用的操作系统版本之一。

centos6/centos7/centos8，及对应的 redhat 版本应该都可以运行，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运行。

建议在 centos7.9 操作系统下使用。

### 2.2 python 版本依赖

IsfMonitor 基于 python 开发，其开发和测试的 python 版本为 **python3.8.8**，推荐使用 **Anaconda3-2021.05** 以解决库依赖问题。

不同版本的 python 可能会有 python 库版本问题，按照系统要求安装对应版本的 python 库即可解决。

### 2.3 集群管理工具

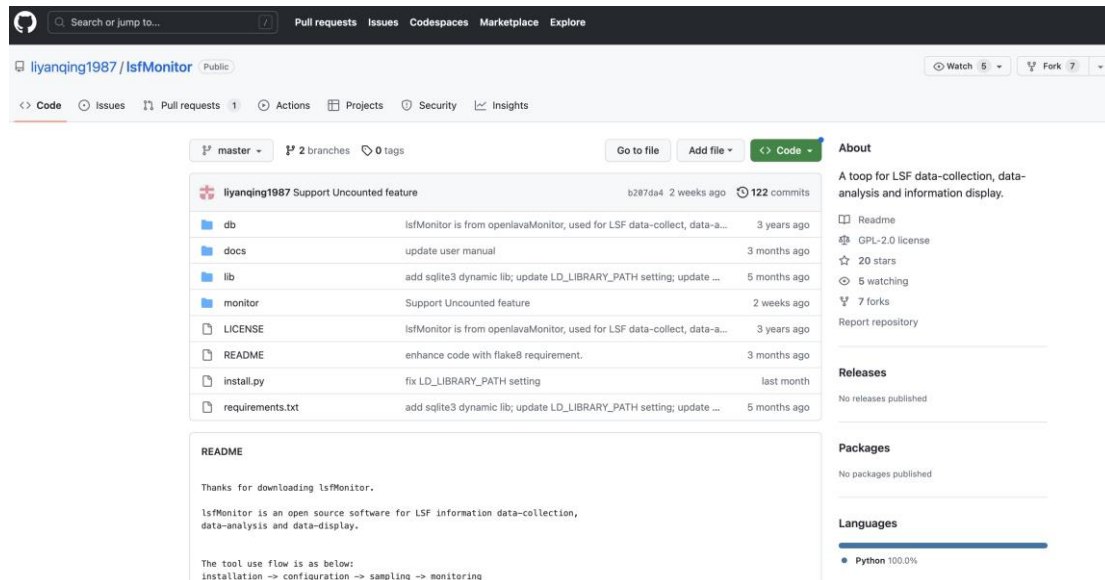
IsfMonitor 依赖 LSF/Oenlava 集群管理系统，暂不支持其它集群管理系统。

LSF 9.\*/10.\*的版本支持较好，Openlava 几个版本间 output message 有一定差异，并不是所有版本都能够很好支持。

## 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>。

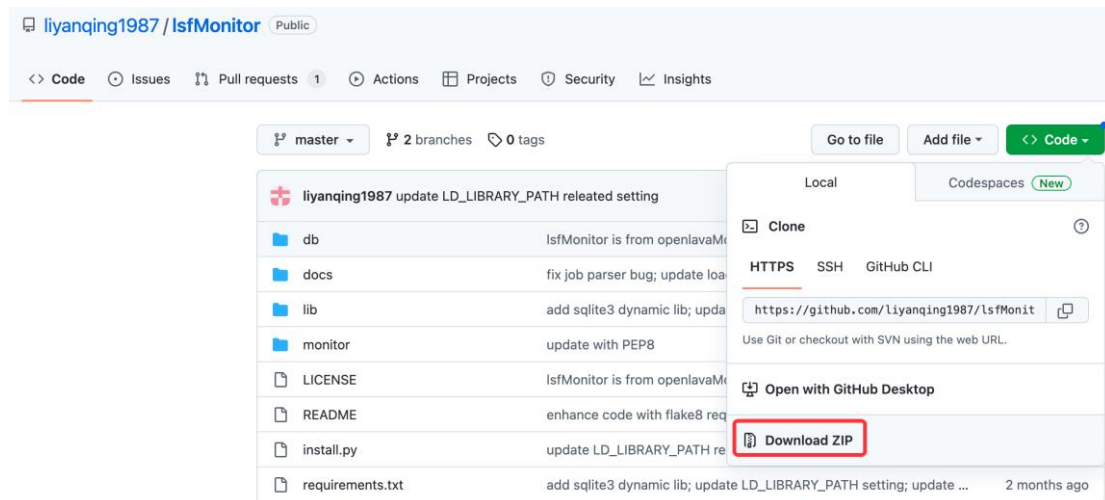


可以采用“git clone <https://github.com/liyanqing1987/lsfMonitor>.git”的方式拉取源代码。

#### Bash

```
[liyanqing@cmp1 test]$ git clone
https://github.com/liyanqing1987/lsfMonitor.git
Cloning into 'lsfMonitor'...
remote: Enumerating objects: 709, done.
remote: Counting objects: 100% (281/281), done.
remote: Compressing objects: 100% (171/171), done.
remote: Total 709 (delta 177), reused 208 (delta 109), pack-reused
428
Receiving objects: 100% (709/709), 908.67 KiB | 594.00 KiB/s,
done.
Resolving deltas: 100% (442/442), done.
```

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



如果国内的用户不方便访问 github，也可以通过 [https://gitee.com/liyanqing1987/ic\\_flow\\_platform](https://gitee.com/liyanqing1987/ic_flow_platform) 获取 lsfMonitor 的源代码。

## 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。

```
Bash
[liyanqing.1987@n212-206-207 tools]$ cd lsfMonitor/
[liyanqing.1987@n212-206-207 lsfMonitor]$ ls
db docs install.py lib LICENSE monitor README
requirements.txt
```

确认 python 版本正确，并基于安装包中的 requirements.txt 安装 python 依赖库。

```
Bash
[root@ic-admin1 licenseMonitor]# pip3 install -r requirements.txt
Looking in indexes: https://bytedpypi.byted.org/simple/
Requirement already satisfied: pexpect==4.8.0 in
/ic/software/tools/python3/3.8.8/lib/python3.8/site-packages (from
-r requirements.txt (line 1)) (4.8.0)
Requirement already satisfied: ptyprocess>=0.5 in
/ic/software/tools/python3/3.8.8/lib/python3.8/site-packages (from
pexpect==4.8.0->-r requirements.txt (line 1)) (0.7.0)
```

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

```
Bash
[root@ic-admin1 lsfMonitor]# python3 install.py
>>> Check python version.
      Required python version : (3, 5)
      Current  python version : (3, 8)

>>> Generate script
"/ic/data/usr/liyanqing.1987/test/lsfMonitor/monitor/bin/bmonitor"
.
>>> Generate script
"/ic/data/usr/liyanqing.1987/test/lsfMonitor/monitor/bin/sample".
>>> Generate script
"/ic/data/usr/liyanqing.1987/test/lsfMonitor/monitor/tools/check_i
ssue_reason".
>>> Generate script
"/ic/data/usr/liyanqing.1987/test/lsfMonitor/monitor/tools/patch".
>>> Generate script
"/ic/data/usr/liyanqing.1987/test/lsfMonitor/monitor/tools/process
_tracer".
>>> Generate script
"/ic/data/usr/liyanqing.1987/test/lsfMonitor/monitor/tools/seedb".
>>> Generate script
"/ic/data/usr/liyanqing.1987/test/lsfMonitor/monitor/tools/show_li
cense_feature_usage".
>>> Generate config file
"/ic/data/usr/liyanqing.1987/test/lsfMonitor/monitor/conf/config.p
y".

Done, Please enjoy it.
```

### 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，大多数都需要重新配置。

```
Bash
# Specify the database directory.
```

```
db_path = "/ic/data/usr/liyanqing.1987/tools/lsfMonitor/db"

# Specify lmstat path, example
"/eda/synopsys/scl/2021.03/linux64/bin/lmstat".
lmstat_path = ""

# Specify lmstat bsub command, example "bsub -q normal -Is".
lmstat_bsub_command = ""
```

**db\_path:** lsfMonitor 需要保存部分 LSF 数据，保存为 sqlite 类型的文本数据库，db\_path 用于指定数据库的顶层路径。

**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 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。

一个使用的 demo 配置如下。

```
Bash
# Specify the database directory.
db_path = "/ic/software/cad_data/it/lsfMonitor/db"

# Specify lmstat path, example
"/eda/synopsys/scl/2021.03/linux64/bin/lmstat".
lmstat_path =
"/ic/software/synopsys/scl/2021.03/linux64/bin/lmstat"

# Specify lmstat bsub command, example "bsub -q normal -Is".
lmstat_bsub_command = "bsub -q normal -Is"
```



## 四、工具使用

lsfMonitor 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 lsfMonitor 安装路径下的 `monitor/bin` 子目录中。

### 4.1 数据采集 `bsample`

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

```
Bash
[liyanqing.1987@n212-206-207 lsfMonitor]$ monitor/bin/bsample -h
usage: bsample.py [-h] [-j] [-q] [-H] [-l] [-u] [-U]

optional arguments:
  -h, --help            show this help message and exit
  -j, --job              Sample running job info 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 info with command "busers".
  -U, --utilization     Sample utilization (slot/cpu/mem) info with
command "lsload/bhosts/lshosts".
```

**--help:** 打印帮助信息。

**--job:** 采集 job 信息并存储。

**--queue:** 采集 queue 信息并存储。

**--host:** 采集 host 信息并存储。

**--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。

我们推荐用 `crontab` 来定时采样（Jenkins 类似），推荐 `job/queue/load` 采样间隔为 5 分钟，推荐 `utilization` 采样间隔为 1 小时。下面是一个示例。

```
Bash
# For lsfMonitor
*/5 * * * *
/ic/software/cad_tools/it/lsfMonitor/monitor/bin/bsample -j
*/5 * * * *
/ic/software/cad_tools/it/lsfMonitor/monitor/bin/bsample -q
*/5 * * * *
/ic/software/cad_tools/it/lsfMonitor/monitor/bin/bsample -l
0 * * * * /ic/software/cad_tools/it/lsfMonitor/monitor/bin/bsample
-U
```

特别注意：

V1.3 版本开始，bsample 采样数据库的数据格式有变化，跟之前版本 bsample 生成的数据库不通用，所以只能清理数据库后重新安装使用，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。

- 直接引用 bmonitor 绝对路径。
- 采用 modules 管理和加载环境，引用 bmonitor 脚本名即可。
- 将 bmonitor 连接到 LSF 的 bsub 脚本路径中，引用 bmonitor 脚本名即可。

推荐第三种方式，下面是具体效果。

```
Bash
[liyanqing.1987@n212-206-207 ~]$ which bmonitor
/ic/software/tools/lsf/10.1/linux2.6-glibc2.3-x86_64/bin/bmonitor
```

### 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，不过这些参数都可以在 bmonitor 启动后设置。

Bash

```
[liyanqing.1987@n212-206-207 lsfMonitor]$ monitor/bin/monitor -h
usage: bmonitor.py [-h] [-j JOBID] [-u USER] [-f FEATURE] [-t
{JOB,JOBS,HOSTS,QUEUES,LOAD,LICENSE}] [-dl]
```

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
                           Specify license feature which you want to
                           see on "LICENSE" tab.
-t {JOB,JOBS,HOSTS,QUEUES,LOAD,LICENSE}, --tab
{JOB,JOBS,HOSTS,QUEUES,LOAD,LICENSE}
                           Specify current tab, default is "JOB" tab.
-dl, --disable_license    Disable license check function.
```

**--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。

**--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, LICENSE}:** 指定页面，会将 bmonitor 打开到指定 GUI 页面。

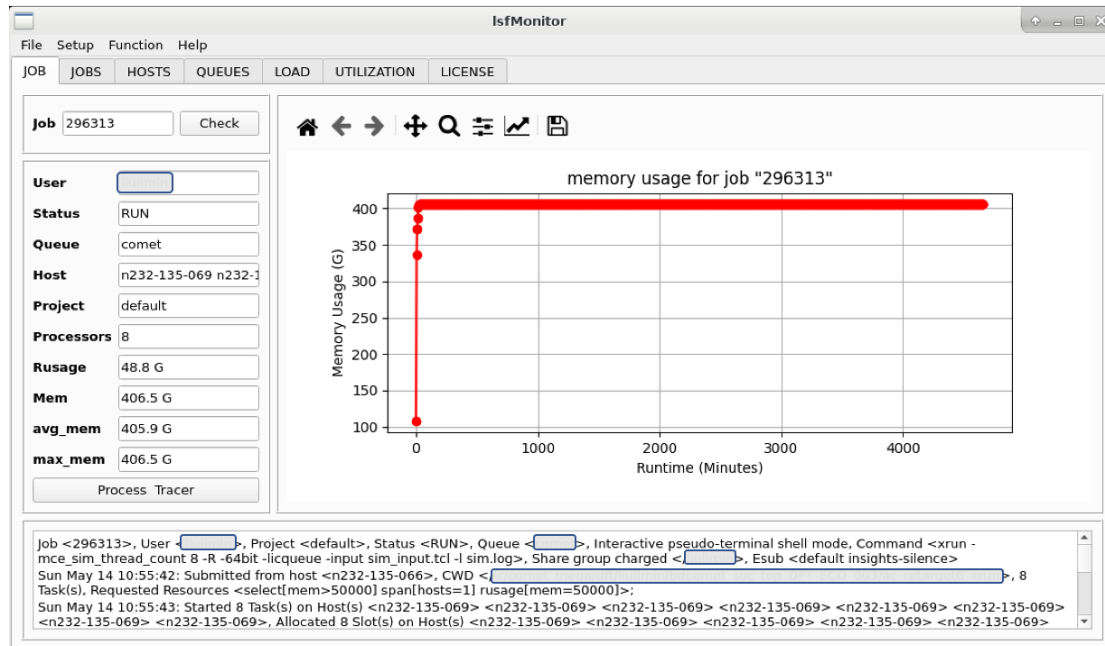
**--disable\_license:** 启动的时候不执行 license 信息获取步骤，以加快 GUI 打开速度。

### 4.2.3 JOB 页

lsfMonitor 的 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。

- 获取 job 详细信息和内存用量曲线

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 bjob -UF <JOBID>）和 job 的内存用量曲线。



通过 job 的详细信息，可以帮助用户 debug 如下常见问题。

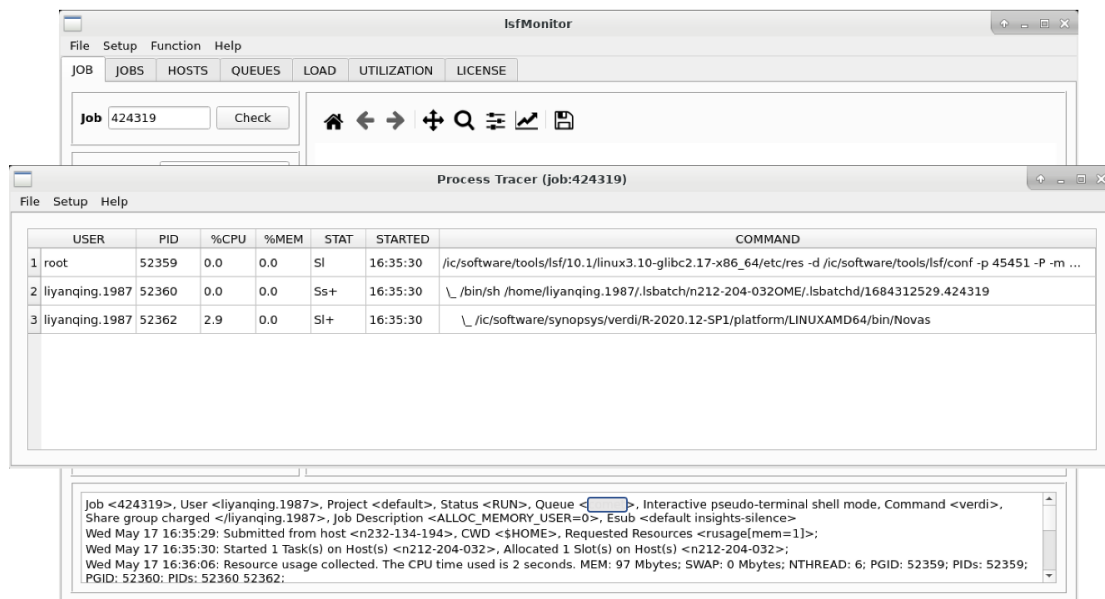
**job PEND 原因：**如果 job 是 PEND 状态，可以从详细信息中的"PENDING REASONS:"看到 job PEND 的原因。

**job FAIL 原因：**如果 job 是 EXIT 状态，可以从退出码判断 job 异常退出是系统原因还是工具原因（退出码小于 128 为工具原因，大于 128 为系统原因）。

**job SLOW 原因：**如果 job 运行缓慢，可以通过进程追踪的方式判断 job 卡在哪。

- 追踪 job 进程 (trace job slow issue)

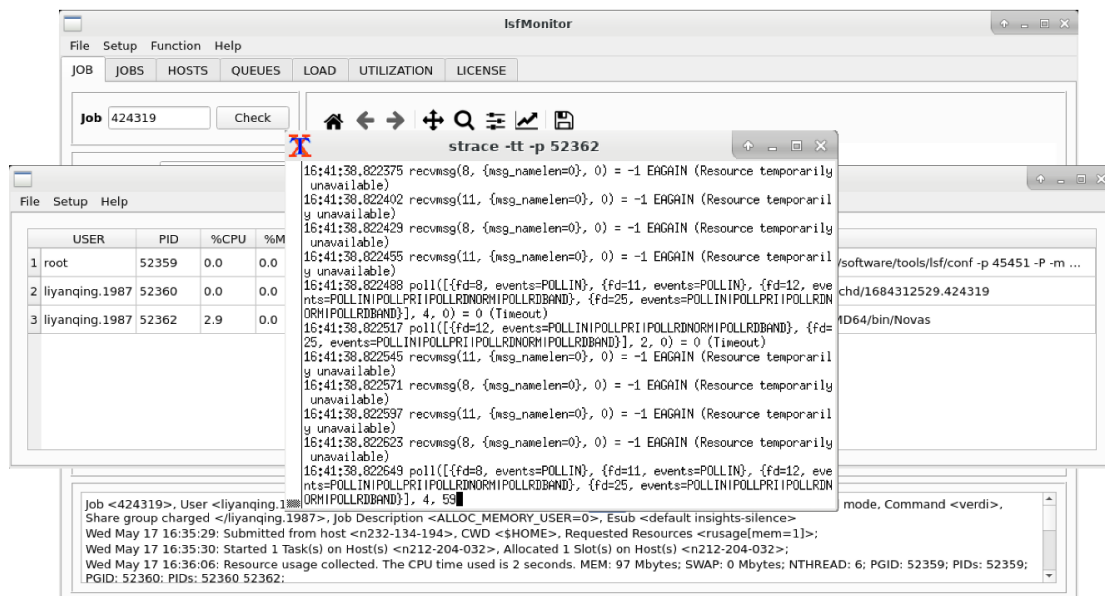
如果感觉 job 运行过于缓慢，可以点击 JOB 页的“Process Tracer”按钮追踪 job 的进程状态。



此时是以进程树的方式展现 job 相关的所有进程，其中核心进程是下面 EDA 工具的进程。

如果核心进程是 R（run）的状态，那么状态是正常的。

如果所有进程都是 S（sleep）的状态，那么任务可能遇到异常，可以点击 process\_tracer 工具的 PID 列对应的 pid 查看相应进程的工作状态。比如此时点击“PID”列的最后一项 52362，可以看到这个 Novas 进程的系统交互细节。



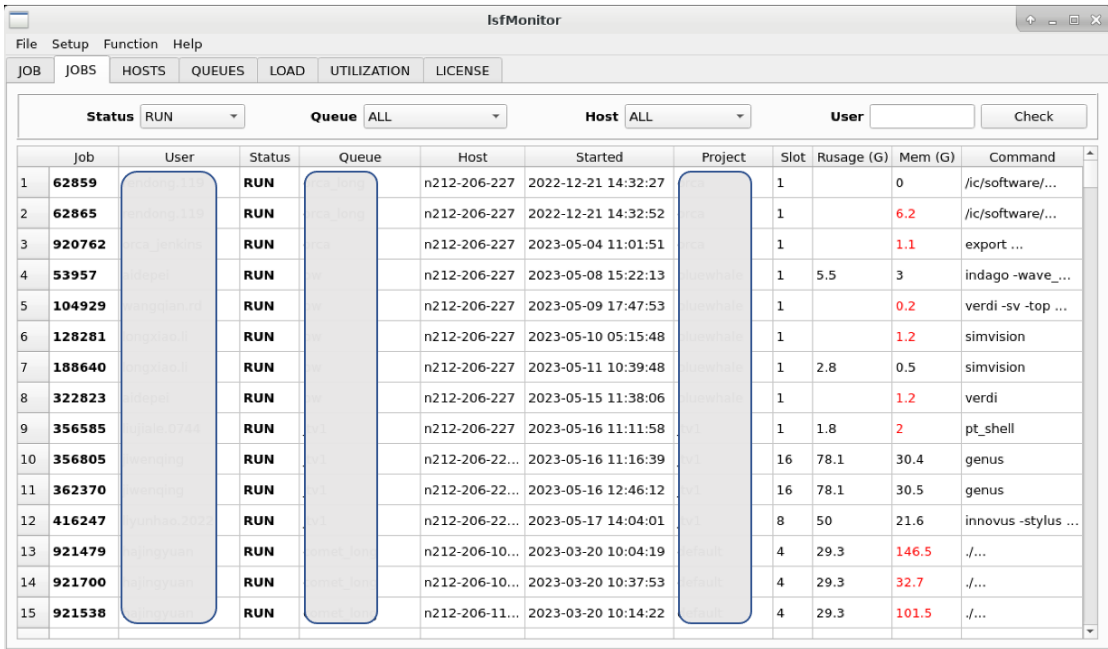
进程的系统交互情况可以帮助用户判断 job 运行缓慢的原因，绝大多数情况下，EDA 工具运行极端缓慢都是由于工具本身问题导致的。

## 4.2.4 JOBS 页

IsfMonitor 的 JOBS 页，主要用于批量查看 jobs 的信息。

可以通过 Status/Queue/Host 下来菜单来筛选 job。

也可以通过 User 输入框输入用户名，然后点击 Check 按钮来按照 user 筛选 job。



The screenshot shows the IsfMonitor application window with the 'JOBS' tab selected. The interface includes filters for Status (set to 'RUN'), Queue (set to 'ALL'), Host (set to 'ALL'), and a User input field with a 'Check' button. Below the filters is a table listing 15 jobs. The table columns are Job, User, Status, Queue, Host, Started, Project, Slot, Rusage (G), Mem (G), and Command. The 'Rusage' and 'Mem' columns show values in red for jobs 13, 14, and 15, indicating high memory usage.

Job	User	Status	Queue	Host	Started	Project	Slot	Rusage (G)	Mem (G)	Command
1 62859	zhangqiang_134	RUN	ic_long	n212-206-227	2022-12-21 14:32:27	ic	1		0	/ic/software/...
2 62865	zhangqiang_134	RUN	ic_long	n212-206-227	2022-12-21 14:32:52	ic	1	6.2		/ic/software/...
3 920762	zhangqiang_134	RUN	ic	n212-206-227	2023-05-04 11:01:51	ic	1	1.1		export ...
4 53957	zhangqiang_134	RUN	ic	n212-206-227	2023-05-08 15:22:13	ic	1	5.5	3	indago -wave_...
5 104929	zhangqiang_134	RUN	ic	n212-206-227	2023-05-09 17:47:53	ic	1		0.2	verdi -sv -top ...
6 128281	zhangqiang_134	RUN	ic	n212-206-227	2023-05-10 05:15:48	ic	1		1.2	simvision
7 188640	zhangqiang_134	RUN	ic	n212-206-227	2023-05-11 10:39:48	ic	1	2.8	0.5	simvision
8 322823	zhangqiang_134	RUN	ic	n212-206-227	2023-05-15 11:38:06	ic	1		1.2	verdi
9 356585	zhangqiang_134	RUN	ic	n212-206-227	2023-05-16 11:11:58	ic	1	1.8	2	pt_shell
10 356805	zhangqiang_134	RUN	ic	n212-206-227	2023-05-16 11:16:39	ic	16	78.1	30.4	genus
11 362370	zhangqiang_134	RUN	ic	n212-206-227	2023-05-16 12:46:12	ic	16	78.1	30.5	genus
12 416247	zhangqiang_134	RUN	ic	n212-206-227	2023-05-17 14:04:01	ic	8	50	21.6	innovus -stylus ...
13 921479	zhangqiang_134	RUN	ic	n212-206-10...	2023-03-20 10:04:19	ic	4	29.3	146.5	./...
14 921700	zhangqiang_134	RUN	ic	n212-206-10...	2023-03-20 10:37:53	ic	4	29.3	32.7	./...
15 921538	zhangqiang_134	RUN	ic	n212-206-11...	2023-03-20 10:14:22	ic	4	29.3	101.5	./...

- jobs 基本信息

可以按照 Status / Queue / Host / User 来过滤和筛选 jobs。

默认 jobs 的信息来源于命令 `bjobs -UF -r -u all`，部分展示参数意义如下：

Host	Started	Processor	Slot	Rusage	Mem
job 执行的机器	job 开始 RUN 的时间	job 申请的 cpu 核数	job 申请的 slots	job 申请的 memory	job 实际使用的 memory

- jobs 特殊操作/显示

点击 Job 列的 jobid，可以直接跳转到 JOB 页，并展示选中 job 的信息。

Mem 列，点击标题，可以让 job 按照 memory 用量排序。

如果 Rusage（job 的内存申请量）没有设，或者 Rusage 小于 Mem（job 的实际内存用量）的值，Mem 值会显示红色，对应的 job 设置都是不规范的，需要修正。

如果 Status 列为 PEND，点击 PEND，可以通过工具“check\_issue\_reason”来查看 job PEND 的原因。

## 4.2.5 HOSTS 页

IsfMonitor 的 JOBS 页，主要用于查看所有 hosts 的信息。

可以按照 Queue 来筛选 hosts。

IsfMonitor

File

Setup

Function

Help

JOB

JOBS

HOSTS

QUEUES

LOAD

UTILIZATION

LICENSE

Queue

ALL

	Host	Status	Queue	Ncpus	MAX	Njobs	Ut (%)	MaxMem (G)	Mem (G)	MaxSwp (G)	Swp (G)	Tmp (G)
6	n212-204-013	ok	...	64	64	11	13	1945	1843	127	110	1581
7	n212-204-014	ok	...	64	64	11	0	1945	1843	127	122	1591
8	n212-204-015	ok	...	64	64	17	0	1945	1740	127	126	1607
9	n212-204-016	ok	...	64	64	12	0	1945	1740	127	127	1608
10	n212-204-017	ok	...	64	64	28	13	1945	1740	127	127	1598
11	n212-204-018	ok	...	64	64	23	5	1945	1740	127	127	1600
12	n212-204-019	ok	...	64	64	20	0	1945	1638	127	125	1592
13	n212-204-020	ok	...	64	64	22	13	1945	1536	127	127	1592
14	n212-204-021	ok	...	64	64	25	13	1945	1638	127	127	1586
15	n212-204-032	ok	...	64	64	24	33	1945	959	127	124	1569
16	n212-204-033	closed_Busy	...	64	64	44	27	1945	93	127	124	1608
17	n212-204-035	ok	...	64	64	56	21	1945	190	127	127	1608
18	n212-204-036	closed_Busy	...	64	64	56	24	1945	73	127	93	1608
19	n212-204-037	ok	...	64	64	25	2	1945	1228	127	124	1596
20	n212-204-038	closed_Adm	...	64	64	4	2	1945	1740	127	125	1608

- hosts 基本信息

默认 hosts 的信息来源于命令 bhosts（host 的 job 信息）、lshosts（host 的静态资源信息）和 lslload（host 的动态资源信息），部分展示参数意义如下：

Queue	Ncpus	MAX	Njobs	Ut	Maxmem	Mem	Tmp
host 所属的队列	host 的 cpu 物理核数	host 接收 job 的最大数目	host 上 job 的数目	host 的 cpu 使用率	host 的内存大小	host 的内存可用量	host 的 /tmp 空间可用量

- hosts 特殊操作/显示

点击 Host 列的服务器名，可以跳转到 LOAD 页，展示指定 host 的 cpu 和

memory 历史用量曲线。

点击 Njobs 列的数字，可以跳转到 JOBS 也，展示指定 host 上所有的 RUN/PEND jobs。

数字内容的列，都可以通过点击列标题进行排序。

如果 host 的状态异常 (unavail/unreach/closed\_LIM)，Status 状态会显示为红色。

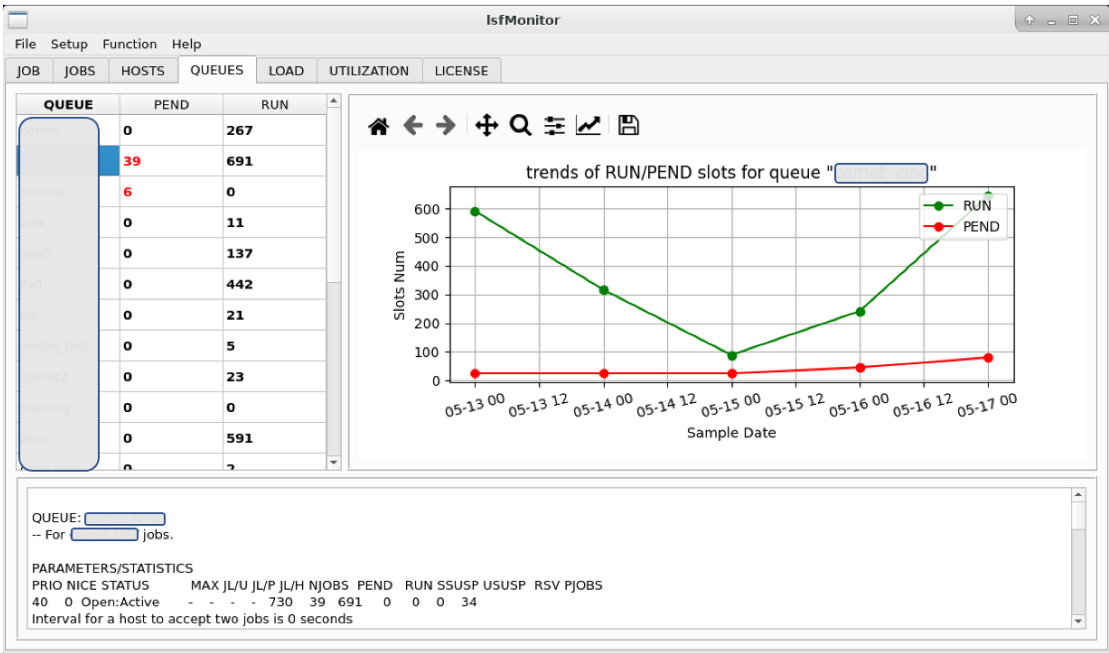
如果 host 的 Ut 使用率超过 90%，Ut 值会显示为红色，表示 cpu 过载。

如果 host 的 memory 用量超过 90%，Mem 值会显示为红色，标识 memory 过载。

如果 host 的 tmp 可用量变为 0，Tmp 值会显示为红色，标识/tmp 过载。

### 4.2.6 QUEUES 页

IsfMonitor 的 QUEUES 页，主要用于查看所有 queue 的信息。



如果 queue 中 PEND 的 job 数目不为 0，数字会被红标。

点击 QUEUE 列的队列名，可以展示 queue 的详细信息和 queue 中 PEND/JOB 数据的历史变化曲线。

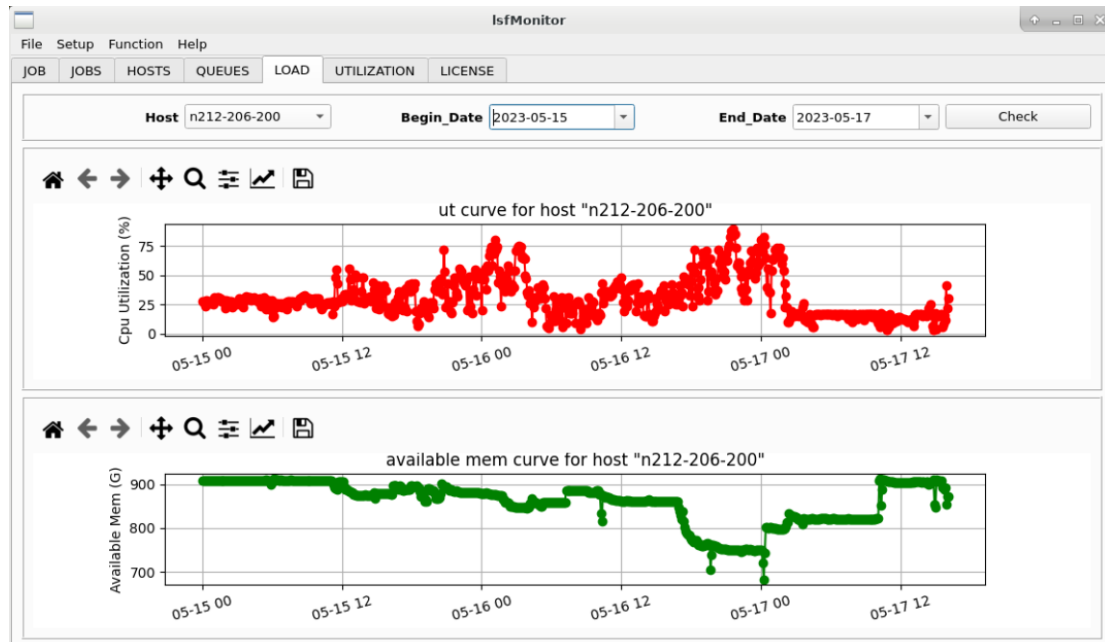
其中 queue 的 PEND/RUN 的历史数据默认展示 15 天，每天的值都是一个统计均值。



## 4.2.7 LOAD 页

IsfMonitor 的 LOAD 页，主要用于查看 host 的负载信息。

通过 Host 下拉菜单选中服务器，通过 Begin\_Date 和 End\_Date 选中日期范围，点击 Check 按钮，即可查看服务器的 ut/available\_mem 的变化曲线。

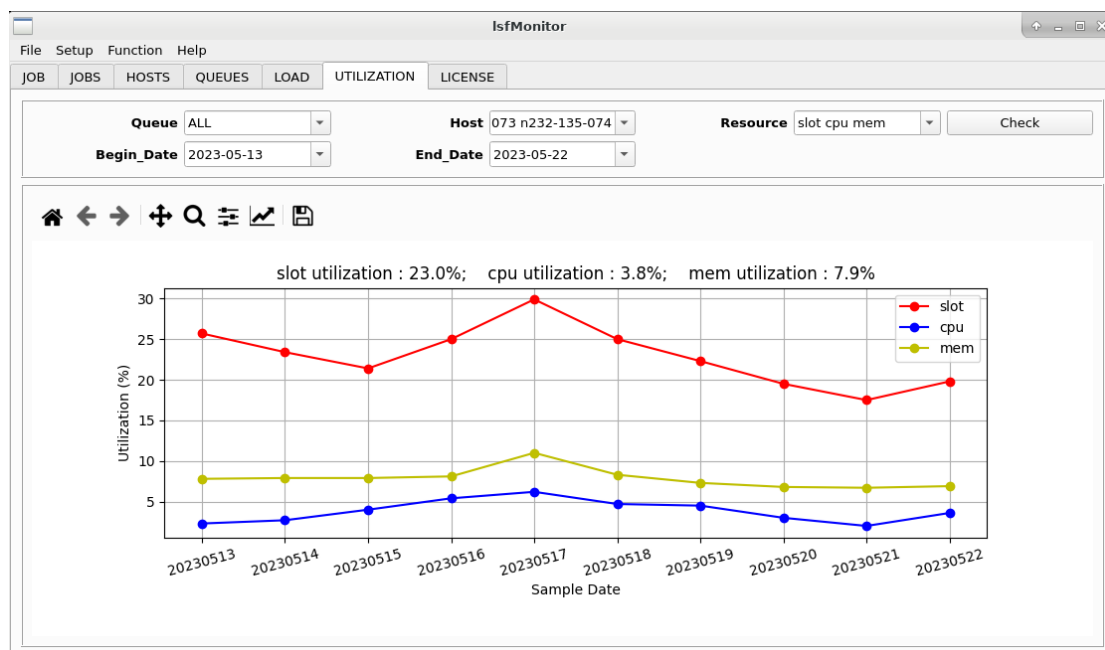


其中 ut 和 mem 的曲线均可以通过放大镜按钮放大指定部位，也可以通过 HOME 按钮恢复原状。

## 4.2.8 UTILIZATION 页

IsfMonitor 的 UTILIZATION 页，主要用来查看 slot/cpu/memory 等资源的使用率统计信息。

可以通过 Queue 来选择服务器，也可以直接通过 Host 来选择服务器，通过 Resource 来多选要统计的资源项，Begin\_Date 和 End\_Date 用来统计时间段。选好后，点击 Check。



每种类型的资源，会在标题栏处显示所选日期的利用率均值，也会在折线图中显示所选日期的使用率变化曲线。

## 4.2.9 LICENSE 页

IsfMonitor 的 LICENSE 页，主要用于查看 EDA license 的使用情况。

启动 IsfMonitor 前，需要保证当前 terminal 已经配置 LM\_LICENSE\_FILE 等 license 相关的环境变量。

IsfMonitor

File Setup Function Help

JOB

JOBS

HOSTS

QUEUES

LOAD

UTILIZATION

LICENSE

Show

ALL

Server

ALL

Vendor

ALL

Feature

Filter

Feature Information

Expires Information

	Server	Vendor	Feature	Issued	In Use		License Server	Feature	Num	Expires
1	27020@ic-synopsys...	snpslmd	SSS	1	0	1	27020@ic-synopsys...	SSS	1	08-Oct-2023
2	27020@ic-synopsys...	snpslmd	Debug_HAPS-100	8	0	2	27020@ic-synopsys...	Debug_HAPS-100	8	08-oct-2023
3	27020@ic-synopsys...	snpslmd	HAPS-100-AMBA...	8	0	3	27020@ic-synopsys...	HAPS-100-AMBA-XTOR...	8	08-oct-2023
4	27020@ic-synopsys...	snpslmd	HAPS-100-AMBA...	8	0	4	27020@ic-synopsys...	HAPS-100-AMBA-XTOR...	8	08-oct-2023
5	27020@ic-synopsys...	snpslmd	ProtoCompiler100	16	9	5	27020@ic-synopsys...	ProtoCompiler100	16	08-oct-2023
6	27020@ic-synopsys...	snpslmd	ProtoCompiler100-RT	8	0	6	27020@ic-synopsys...	ProtoCompiler100-RT	8	08-oct-2023
7	27020@ic-synopsys...	snpslmd	ProtoCompilerMP	8	0	7	27020@ic-synopsys...	ProtoCompilerMP	8	08-oct-2023
8	27020@ic-synopsys...	snpslmd	VCSCompiler_Net	8	0	8	27020@ic-synopsys...	VCSCompiler_Net	8	08-oct-2023
9	27020@ic-synopsys...	snpslmd	VHDL-Elaborator	8	0	9	27020@ic-synopsys...	VHDL-Elaborator	8	08-oct-2023
10	27020@ic-synopsys...	snpslmd	VT_Assertions	8	0	10	27020@ic-synopsys...	VT_Assertions	8	08-oct-2023
11	27020@ic-synopsys...	snpslmd	VT_Coverage	8	0	11	27020@ic-synopsys...	VT_Coverage	8	08-oct-2023
12	27020@ic-synopsys...	snpslmd	VT_Testbench	8	0	12	27020@ic-synopsys...	VT_Testbench	8	08-oct-2023
13	27020@ic-synopsys...	snpslmd	VT_VCS_Xprop	8	0	13	27020@ic-synopsys...	VT_VCS_Xprop	8	08-oct-2023
14	27020@ic-synopsys...	snpslmd	mvdngen	8	0	14	27020@ic-synopsys...	mvdngen	8	08-oct-2023

- **license 过滤**

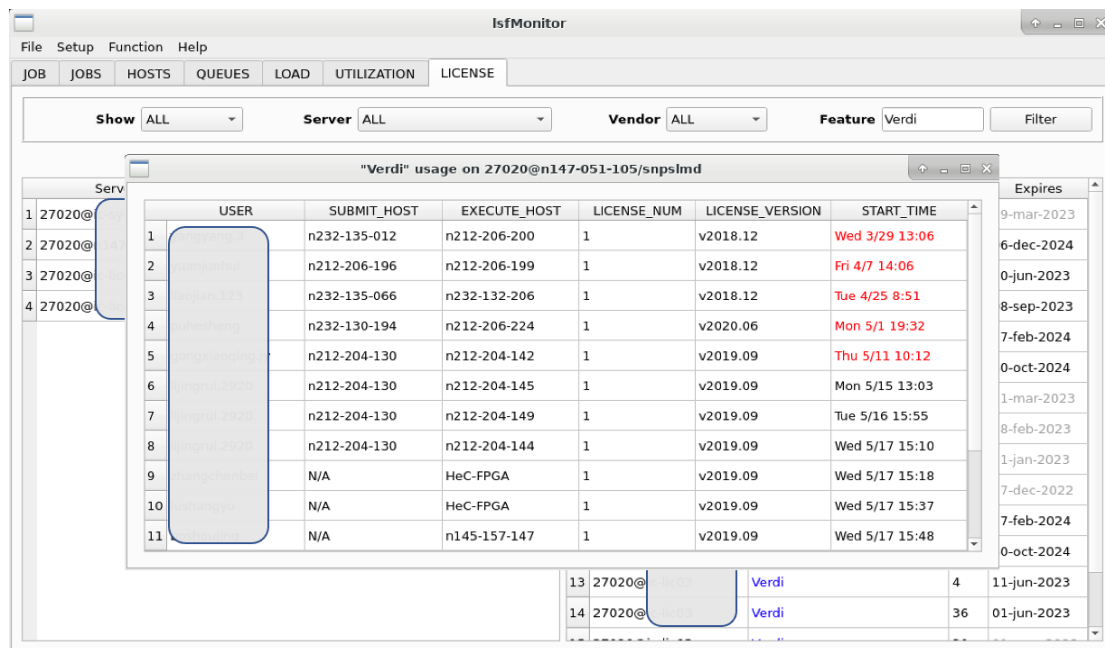
默认显示所有 license 的信息，但是也可以按照“Show”，“Server”，“Vendor”，和“Feature”来过滤 License 信息。

一般来说，最常用的方法是直接检索希望查看的 license feature 名字。（支持模糊检索）

比如搜索一下 verdi 相关的 license feature 使用信息，已知其核心 feature 名为“Verdi”，在 Feature 框输入，点击“Filter”按钮过滤。

Feature Information					Expires Information			
Server	Vendor	Feature	Issued	In_Use	License Server	Feature	Num	Expires
1 27020@synopsys.com	snpslmd	Verdi	6	5	1 27020@synopsys.com	Verdi	2	19-mar-2023
2 27020@147-051-100	snpslmd	Verdi	14	15	2 27020@synopsys.com	Verdi	6	06-dec-2024
3 27020@147-051-100	snpslmd	Verdi	22	22	3 27020@147-051-100	Verdi	12	10-jun-2023
4 27020@147-051-100	snpslmd	Verdi	77	39	4 27020@147-051-100	Verdi	2	08-sep-2023
					5 27020@147-051-100	Verdi	12	27-feb-2024
					6 27020@147-051-100	Verdi	10	30-oct-2024
					7 27020@147-051-100	Verdi	5	31-mar-2023
					8 27020@147-051-100	Verdi	5	28-feb-2023
					9 27020@147-051-100	Verdi	5	21-jan-2023
					10 27020@147-051-100	Verdi	5	17-dec-2022
					11 27020@147-051-100	Verdi	12	27-feb-2024
					12 27020@147-051-100	Verdi	25	30-oct-2024
					13 27020@147-051-100	Verdi	4	11-jun-2023
					14 27020@147-051-100	Verdi	36	01-jun-2023

点击“In\_use”列的数字（如果数字非零），可以查看具体谁使用了这些 license。



如果某个 license feature 的 START\_TIME 早于 3 天前，会标红。

- **license 信息**

LICENSE 页会展示“Feature Information”和“Expires Information”两类信息。

其中“Feature Information”指 license 的使用信息，包括如下内容：

Server	Vendor	Feature	Issued	In_use
license server 名	Vendor daemon 名	license feature 名	一般认为是 license file 中 license 的总数，包括过期的部分。	在使用的 liense feature 数目。

“Expires Information”值 license 的过期信息，包括如下内容：

License Server	Feature	Num	Expires
license server 名	license feature 名。 因为 license file 可以追加内容，所以同一 feature 可能在同一 license	License feature 数目	过期日期 过期会标灰 两周内过期会

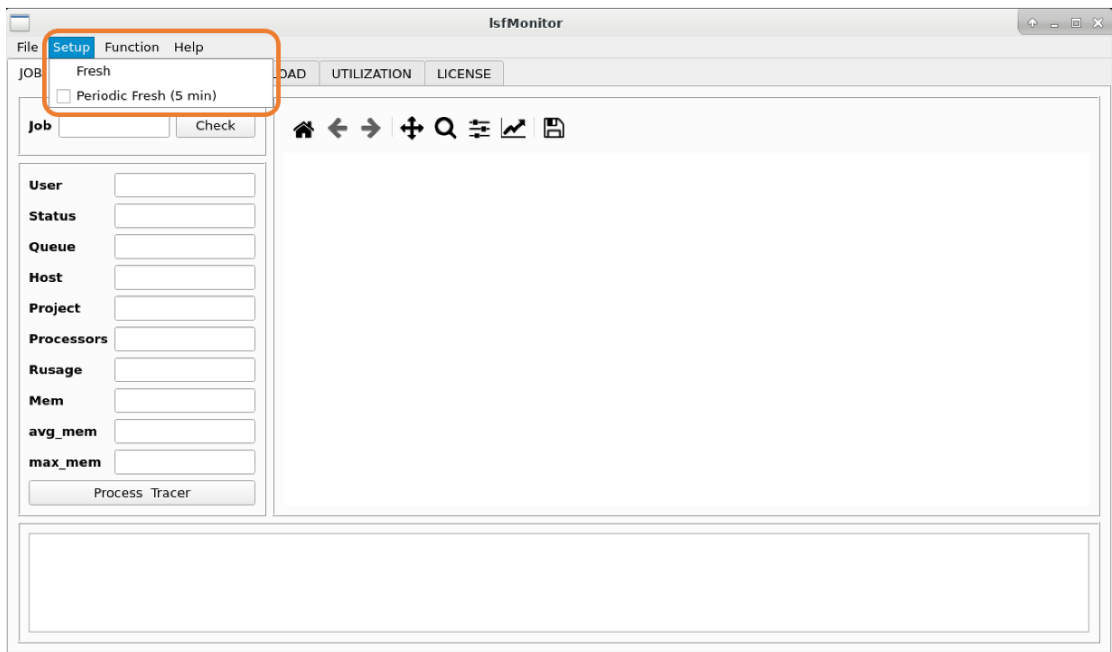
	server 中出现多次。		标红
--	---------------	--	----

### 4.2.10 其它功能

- 信息刷新功能

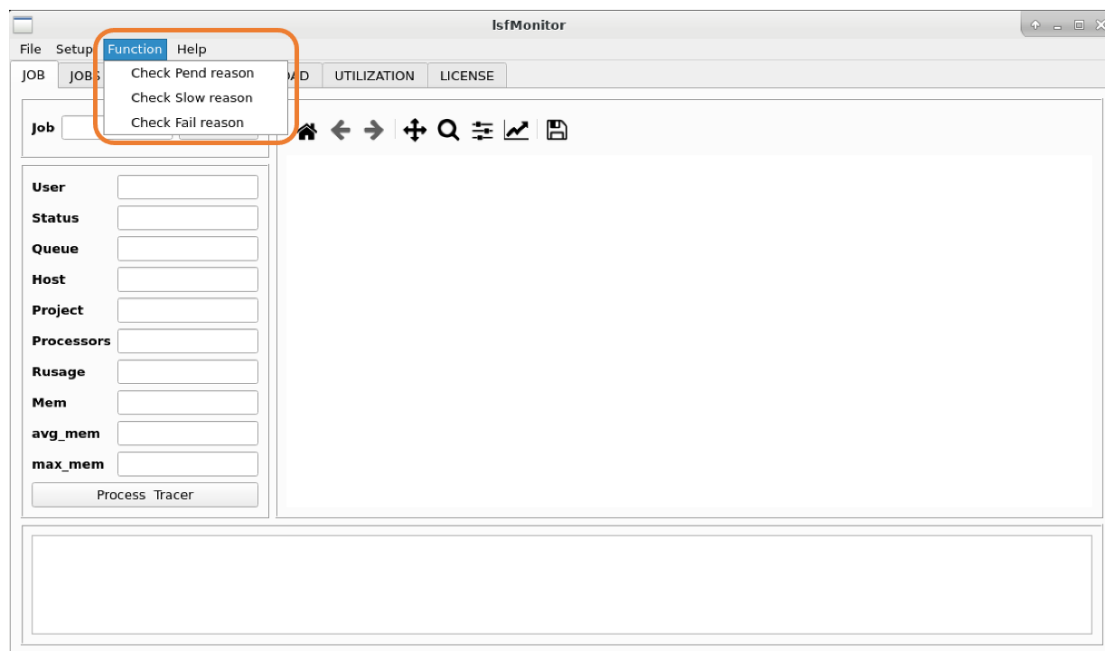
IsfMonitor 默认展示静态信息，也就是说，在 GUI 界面启动时会抓取瞬时的 LSF/Openlava/License 信息，后续如果用户没有特殊操作，相关信息不会更新。

如果用户希望一次性或者规律性更新 GUI 界面上的信息，可以通过菜单栏 Setup 的 Fresh 或者 Periodic Fresh 功能来刷新界面信息。

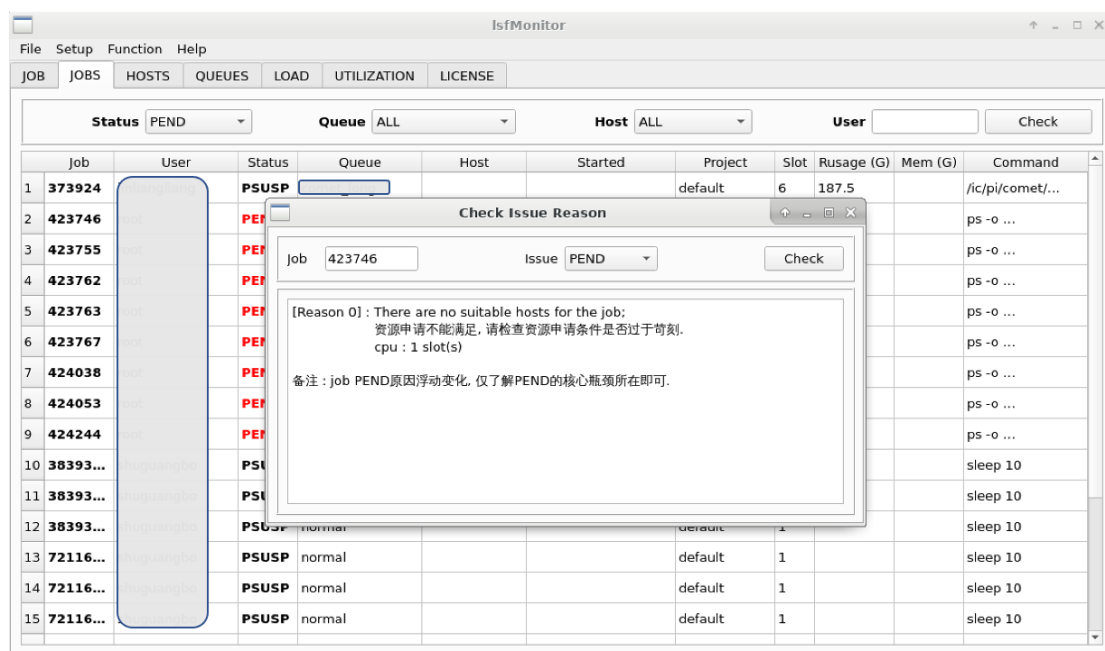


- 异常 job 问题检查

菜单栏 Function 中包含 job pend/slow/fail 原因查询功能。



在弹出的 Check Issue Reason 工具中，输入 jobid，点击“Check”按钮，如果显示相关问题的检测结果。



## 五、辅助工具

lsfMonitor 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。

Bash

```
[liyanqing.1987@n212-206-207 lsfMonitor]$ ls monitor/tools/  
check_issue_reason  check_issue_reason.py  message.py  patch  
patch.py  process_tracer  process_tracer.py  seedb  seedb.py  
show_license_feature_usage  show_license_feature_usage.py
```

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

message.py: 组件，图形界面工具，显示指定信息。

patch.py: 组件，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，lsfMonitor 即使用 sqlite3 保存相关数据。

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

### 5.1 seedb

seedb 是查看 sqlite3 文本数据库内容的工具，其帮助信息如下：

Bash

```
[liyanqing.1987@n212-206-207 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。

下面是一个使用示例，查看 load.db 数据库中，查看 load\_n212-206-211 表中 mem 列的前五行数据。

```
Bash
[liyanqing.1987@n212-206-207 lsfMonitor]$ monitor/tools/seedb -d
/ic/software/cad_data/it/lsfMonitor/db/monitor/load.db -t
load_n212-206-211 -k mem -n 5
DB FILE : /ic/software/cad_data/it/lsfMonitor/db/monitor/load.db
TABLE : load_n212-206-211
=====
mem
----
911.9G
910G
908.1G
906G
904.6G
=====
```

## 5.2 patch

patch 是帮助 lsfMonitor 打补丁的工具，其帮助信息如下。

```
TypeScript
[liyanqing.1987@n212-206-207 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。

```
TypeScript
[liyanqing.1987@n212-206-207 lsfMonitor]$ monitor/tools/patch -p
/ic/data/usr/liyanqing.1987/test/lsfMonitor-master
Install Path : /ic/data/usr/liyanqing.1987/test/lsfMonitor
Patch path : /ic/data/usr/liyanqing.1987/test/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/test/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/test/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/test/lsfMonitor-
master/monitor/tools/seedb.py" into
"/ic/data/usr/liyanqing.1987/test/lsfMonitor/monitor/tools/seedb.p
y".
> Copying python file
"/ic/data/usr/liyanqing.1987/test/lsfMonitor-
master/monitor/tools/patch.py" into
"/ic/data/usr/liyanqing.1987/test/lsfMonitor/monitor/tools/patch.p
y".
> Copying python file
"/ic/data/usr/liyanqing.1987/test/lsfMonitor-
```

```
master/monitor/tools/tool_1.py" into  
"/ic/data/usr/liyanqing.1987/test/lsfMonitor/monitor/tools/tool_1.  
py".
```

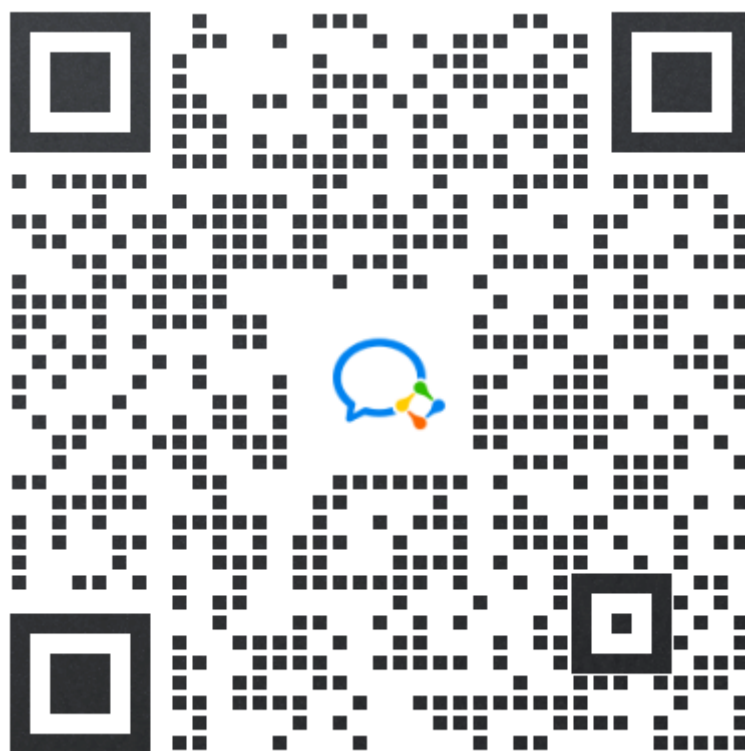
## 六、技术支持

本工具为开源工具，无商业支持，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。
- bug 反馈及修复。
- 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。
- 加入官方技术支持群。



# 附录

## 附 1. 变更历史

日期	版本	变更描述	主要源代码变更	备注
2017	1.0	发布第一个版本 openlavaMonitor，仅支持 openlava。		
2020	1.1	更名 IsfMonitor，增加对 LSF 的支持。		
2022	1.2	增加 LICENSE 信息的采集 和展示。		
2023.5	1.3	增加 UTILIZATION 信息的 采集和展示。  增加 patch 工具。  优化数据库数据格式。	bsample.py bmonitor.py common_pyqt5.py patch.py （新增）	数据库格式跟 旧版本不兼 容，需要清空 旧数据库后重 新采样。