

licenseMonitor 用户手册

Product Name : licenseMonitor

Product Version : V1.3.1

Release Date : 2024.03.06

Contact : [@李艳青](#) (liyanqing.1987@bytedance.com)

[@张静文](#) (zhangjingwen.silvia@bytedance.com)

目录

一、简介.....	4
1.1 主要用途	4
1.2 工作原理	5
二、环境依赖.....	8
2.1 操作系统依赖	8
2.2 PYTHON 版本依赖	8
2.3 使用环境依赖	8
三、工具安装及配置.....	9
3.1 工具下载	9
3.2 工具安装	10
3.3 工具配置	13
3.3.1 config/config.py	14
3.3.2 config/LM_LICENSE_FILE	17
3.3.3 config/project	17
3.3.4 config/utilization	20
3.3.5 config/cost	21
3.3.6 config/others	22
四、工具使用.....	24
4.1 数据采集 LICENSE_SAMPLE	24
4.1.1 帮助信息	24
4.1.2 采样示例	24
4.1.3 数据库	26
4.2 数据展示 LICENSE_MONITOR	28
4.2.1 工具载入	28
4.2.2 帮助信息	29
4.2.3 菜单栏	29
4.2.4 SERVER 页	31
4.2.5 FEATURE 页	31
4.2.6 EXPIRES 页	34
4.2.7 USAGE 页	35
4.2.8 CURVE 页	37
4.2.9 UTILIZATION 页	40
4.2.10 COST 页	45
4.2.11 其它功能	48
五、辅助工具.....	50
5.1 GEN_LM_LICENSE_FILE	50
5.2 GET_PRODUCT_FEATURE_RELATIONSHIP	51
5.3 CONFIG_PRODUCT_FEATURE_RELATIONSHIP	53
5.4 UPDATE_PRODUCT_FEATURE_RELATIONSHIP	55
5.5 VIEW_PRODUCT_FEATURE_REALATIONSHIP	57
5.6 UPDATE_PROJECT_EXECUTE_HOST_WITH_LSF	59
5.7 GET_LICENSE_LOG	60
5.8 COLLECT_FEATURE_RECORD_FROME_LICENSE_LOG	62
5.9 SHOW_LICENSE_LOG_INFO	63
5.10 SEEDB	65

5.11 PATCH	67
六、最佳实践.....	69
6.1 CONFIG/LM_LICENSE_FILE 自动化更新	69
6.2 CONFIG/PROJECT/PROJECT_EXECUTE_HOST 自动化更新	69
6.3 CONFIG/OTHERS/PRODUCT_FEATURE.YAML 自动化更新	70
6.4 CONFIG/OTHERS/LICENSE_LOG.YAML 自动生成	71
6.5 CONFIG/OTHERS/FEATURE_RECORD_ON_LICENSE_LOG.YAML 自动生成	71
七、常见问题及解决.....	73
7.1 图形显示问题	73
7.2 LICENSE 信息没有或者不全	73
八、技术支持.....	75
附录.....	76
附 1. 变更历史	76

一、简介

EDA, electronic design automation, 即电子设计自动化, 是指利用计算机辅助设计软件, 来完成超大规模集成电路芯片的功能设计、综合、验证、物理设计等流程的设计方式。EDA 位于集成电路产业的最上游, 其成本在集成电路设计总成本中占有较高的比重, 全流程的 EDA 工具价格常常达到千万甚至数亿, 出于高效利用和节省成本的考虑, 需要实时了解 EDA 工具的 license 使用情况, 并精准地获取其使用率信息和项目使用状况。

licenseMonitor 工具是一款 ByteDance 开源的 EDA license 信息检索和分析系统, 对 IC 设计降本增效具有重要的意义。

1.1 主要用途

IC 工程师是 EDA license 的主要用户, 其主要的 EDA license 信息获取需求有如下几种:

1. 某个 license feature 是否存在。
2. 某个 license feature 总数量是多少。
3. 某个 license feature 当前被哪些人用了。
4. 某个 license feature 的过期日期是哪天。

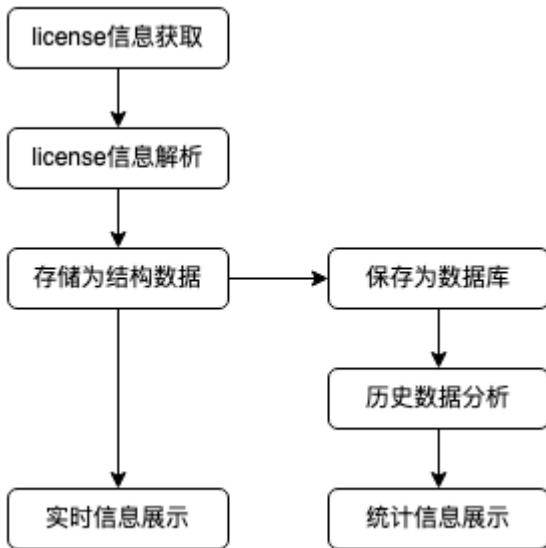
对 IC 环境管理员而言, 还有一些额外的 EDA license 实时信息需求。

1. EDA license server 及 vendor daemon 的状态。(故障警示)
2. 全量 license feature 的数量信息。(总量, 使用量)
3. 全量 license feature 的过期信息。(未过期, 即将过期, 已经过期)
4. 全量 license feature 的使用信息。(谁, 什么时间开始, 使用了几个)
5. EDA license 的使用量历史数据。(总量/用量的周期变化和峰值情况)
6. EDA license 的使用率历史数据。(利用率分析, 下次采购需要增/减采购量)
7. 不同 project 在 EDA license 上的使用情况。(适用于项目费用分摊)

licenseMonitor 就用于满足如上这些实时信息获取和历史信息统计需求。

1.2 工作原理

licenseMonitor 的工作流程如下图所示。



绝大多数 EDA vendor 都是使用 FlexNet 的 lmgrd 来管理 EDA license server，所以可以通过 FlexNet 的工具 lmstat 获取 EDA license 信息，样式如下。

```
License server status: PORT@HOSTNAME
License file(s) on HOSTNAME: LICENSE_FILE:

HOSTNAME: license server UP (MASTER) v11.16.4

Vendor daemon status (on HOSTNAME):

  VENDOR_DAEMON: UP v11.16.4
Feature usage info:

  Users of FEATURE1: (Total of 1 license issued; Total of 0
  licenses in use)
  Users of FEATURE2: (Total of 8 licenses issued; Total of 5
  licenses in use)

  "FEATURE2" v2021.06, vendor: VENDOR_DAEMON, expiry: 28-sep-2022
  vendor_string: UHD:PERM
  floating license

  USER1 EXECUTE_HOST /dev/pts/1 14165.SUBMIT_HOST (v2018.06)
  (HOSTNAME/PORT 482), start Sat 6/18 11:35, 4 licenses
```

USER2 EXECUTE_HOST /dev/pts/2 52926.SUBMIT_HOST (v2018.06)
(HOSTNAME/PORT 2845), start Mon 6/20 15:31

... .

Feature	Version	#licenses	Vendor
Expires			
FEATURE1	1.0	1	
VENDOR_DAEMON	28-Sep-2022		
FEATURE2	2021.06	2	
VENDOR_DAEMON	28-sep-2022		
...			

解析 license 信息，将其保存为 python 的字典，数据结构如下。

```
'expires': '',  
},]  
},  
},  
}  
}
```

最后，将这些结构化的数据直接展示，即为实时数据；将这些结构化的数据存储分析后展示，即为统计数据。

二、环境依赖

2.1 操作系统依赖

licenseMonitor 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

licenseMonitor 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 使用环境依赖

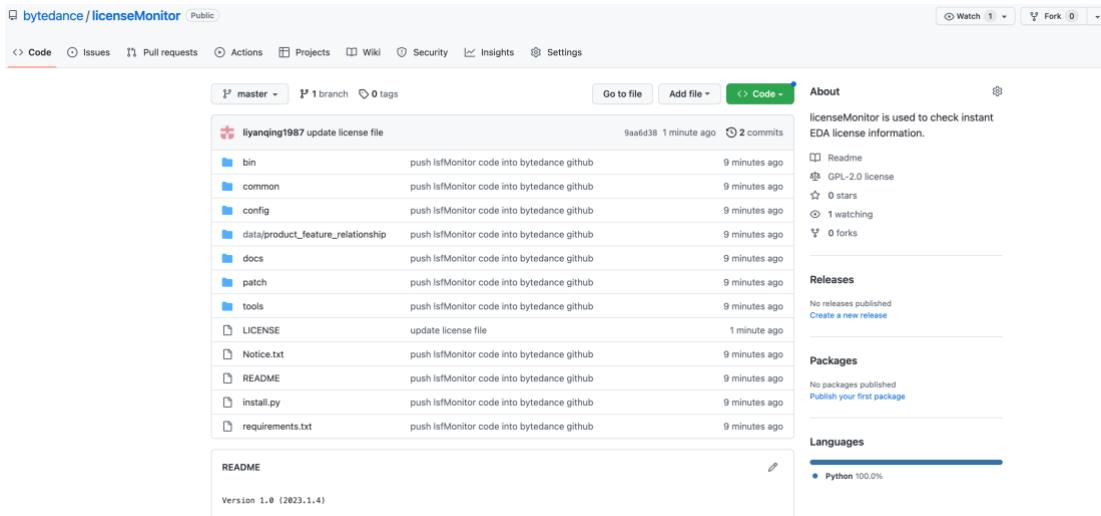
使用 licenseMonitor, 环境需要满足如下依赖:

- 可以访问 license server。如果本机 (login server) 跟 license server 之间网络不通, 那么需要使用 bsub 的方式将 lmstat 丢到 computing server 上去执行。
- 如果是非管理员用户, 需要在环境中配置有正确的 LM_LICENSE_FILE 设置, lmstat 依赖 LM_LICENSE_FILE 的设置获取 license 信息。

三、工具安装及配置

3.1 工具下载

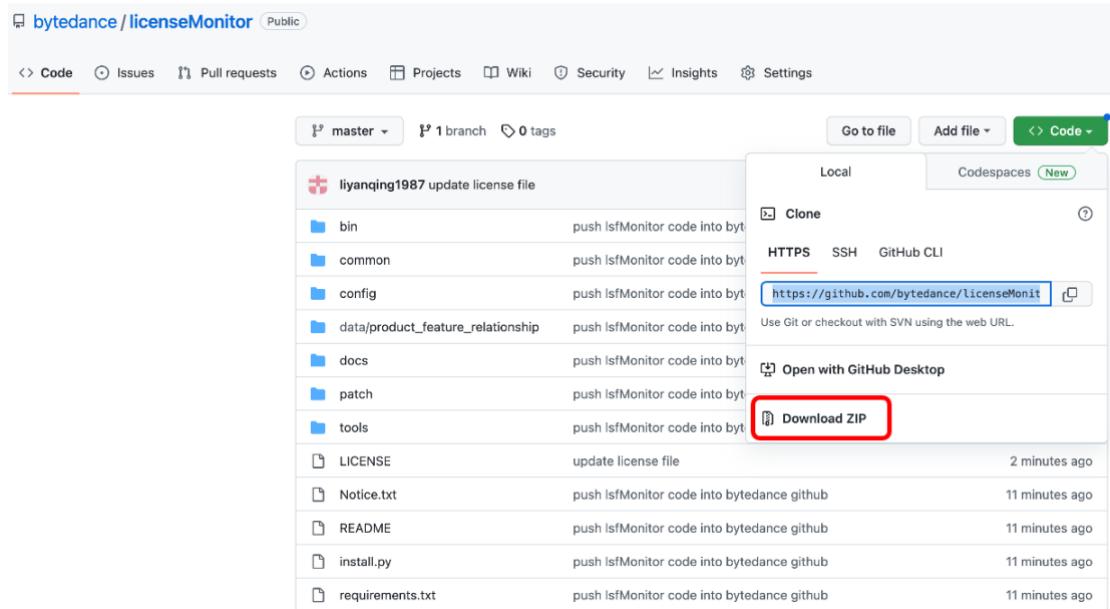
licenseMonitor 的 github 路径位于 <https://github.com/bytedance/licenseMonitor>。



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

```
[liyanqing@personal_pc test]$ git clone  
https://github.com/bytedance/licenseMonitor.git  
Cloning into 'licenseMonitor'...  
remote: Enumerating objects: 26, done.  
remote: Counting objects: 100% (26/26), done.  
remote: Compressing objects: 100% (19/19), done.  
remote: Total 26 (delta 2), reused 26 (delta 2), pack-reused 0  
Unpacking objects: 100% (26/26), done.
```

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



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 licenseMonitor 的环境依赖关系。

安装包下的文件和目录如下。

```
[root@ic-admin2 tools]# cd licenseMonitor
[root@ic-admin2 licenseMonitor]# ls
bin  common  config  data  db  docs  install.py  lib  LICENSE
Notice.txt  README.md  requirements.txt  tools
```

确认 python 版本正确。

```
[root@ic-admin2 licenseMonitor]# python3 --version
Python 3.8.8
```

基于安装包中的 requirements.txt 安装 python 依赖库。(python 库安装一般需要使用 root 账号操作)

```
[root@ic-admin2 licenseMonitor]# pip3 install -r requirements.txt
Looking in indexes: https://bytedpypi/byted.org/simple/
Collecting matplotlib==3.7.2 (from -r requirements.txt (line 1))
    Downloading
https://bytedpypi/byted.org/packages/pypi/matplotlib/matplotlib-
3.7.2-cp38-cp38-manylinux_2_12_x86_64.manylinux2010_x86_64.whl
(9.2 MB)
    ━━━━━━━━━━━━━━━━ 9.2/9.2 MB 105.3
MB/s eta 0:00:00
Collecting PyQt5==5.15.9 (from -r requirements.txt (line 2))
    Downloading
https://bytedpypi/byted.org/packages/pypi/pyqt5/PyQt5-5.15.9-cp37-
abi3-manylinux_2_17_x86_64.whl (8.4 MB)
    ━━━━━━━━━━━━━━━━ 8.4/8.4 MB 154.2
MB/s eta 0:00:00
...

```

在安装目录下，使用命令“python3 install.py”安装 licenseMonitor。(公共软件安装一般需要使用 root 账号操作)

```
[root@ic-admin2 licenseMonitor]# python3 install.py
>>> Check python version.
    Required python version : (3, 8)
    Current  python version : (3, 8)

>>> Generate script
"/ic/data/usr/liyanqing.1987/tools/licenseMonitor/bin/license_mon
itor".
>>> Generate script
"/ic/data/usr/liyanqing.1987/tools/licenseMonitor/bin/license_samp
le".
>>> Generate script
"/ic/data/usr/liyanqing.1987/tools/licenseMonitor/tools/collect_fe
ature_record_from_license_log".
>>> Generate script
"/ic/data/usr/liyanqing.1987/tools/licenseMonitor/tools/config_pro
duct_feature_relationship".
>>> Generate script
"/ic/data/usr/liyanqing.1987/tools/licenseMonitor/tools/gen_LM_LIC
ENSE_FILE".
```

```
>>> Generate script
"/ic/data/usr/liyanqing.1987/tools/licenseMonitor/tools/get_license_log".
>>> Generate script
"/ic/data/usr/liyanqing.1987/tools/licenseMonitor/tools/get_product_feature_relationship".
>>> Generate script
"/ic/data/usr/liyanqing.1987/tools/licenseMonitor/tools/patch".
>>> Generate script
"/ic/data/usr/liyanqing.1987/tools/licenseMonitor/tools/seedb".
>>> Generate script
"/ic/data/usr/liyanqing.1987/tools/licenseMonitor/tools/show_license_log_info".
>>> Generate script
"/ic/data/usr/liyanqing.1987/tools/licenseMonitor/tools/update_product_feature_relationship".
>>> Generate script
"/ic/data/usr/liyanqing.1987/tools/licenseMonitor/tools/update_project_execute_host_with_lsf".
>>> Generate script
"/ic/data/usr/liyanqing.1987/tools/licenseMonitor/tools/view_product_feature_relationship".
>>> Generate config file
"/ic/data/usr/liyanqing.1987/tools/licenseMonitor/config/config.py".
>>> Generate LM_LICENSE_FILE configuration file
"/ic/data/usr/liyanqing.1987/tools/licenseMonitor/config/LM_LICENSE_FILE".
>>> Generate product-feature relationship file
"/ic/data/usr/liyanqing.1987/tools/licenseMonitor/config/others/product_feature.yaml".
    *Warning*: config file
"/ic/data/usr/liyanqing.1987/tools/licenseMonitor/config/others/product_feature.yaml" already exists, will not update it.
>>> Generate project list file
"/ic/data/usr/liyanqing.1987/tools/licenseMonitor/config/project/project_list".
>>> Generate project-submit_host relationship file
"/ic/data/usr/liyanqing.1987/tools/licenseMonitor/config/project/project_submit_host".
>>> Generate project-execute_host relationship file
"/ic/data/usr/liyanqing.1987/tools/licenseMonitor/config/project/project_execute_host".
>>> Generate project-user relationship file
```

```
"/ic/data/usr/liyanqing.1987/tools/licenseMonitor/config/project/project_user".
>>> Generate utilization white feature file
"/ic/data/usr/liyanqing.1987/tools/licenseMonitor/config/utilization/utilization_white_feature".
>>> Generate utilization black feature file
"/ic/data/usr/liyanqing.1987/tools/licenseMonitor/config/utilization/utilization_black_feature".
>>> Generate utilization white product file
"/ic/data/usr/liyanqing.1987/tools/licenseMonitor/config/utilization/utilization_white_product".
>>> Generate utilization black product file
"/ic/data/usr/liyanqing.1987/tools/licenseMonitor/config/utilization/utilization_black_product".
>>> Generate cost white feature file
"/ic/data/usr/liyanqing.1987/tools/licenseMonitor/config/cost/cost_white_feature".
>>> Generate cost black feature file
"/ic/data/usr/liyanqing.1987/tools/licenseMonitor/config/cost/cost_black_feature".
>>> Generate cost white product file
"/ic/data/usr/liyanqing.1987/tools/licenseMonitor/config/cost/cost_white_product".
>>> Generate cost black product file
"/ic/data/usr/liyanqing.1987/tools/licenseMonitor/config/cost/cost_black_product".
```

注意此处的 Warning 信息，“product_feature.yaml”文件无法生成，是因为我们已经预置了一个包含 cadence/synopsys/mentor 等部分 vendor product-feature 映射关系的文件，当后期你还可以更新或替换这个文件。

3.3 工具配置

licenseMonitor 的主要配置文件位于 config 目录下。

```
[root@ic-admin2 licenseMonitor]# cd config/
[root@ic-admin2 config]# ls
config.py  cost  __init__.py  LM_LICENSE_FILE  others  project
utilization
[root@ic-admin2 config]# ls *
config.py  __init__.py  LM_LICENSE_FILE
```

```
cost:  
cost_black_feature  cost_black_product  cost_white_feature  
cost_white_product  
  
others:  
feature_record_on_license_log.yaml  license_log.yaml  
product_feature.yaml  
  
project:  
project_execute_host  project_list  project_submit_host  
project_user  
  
utilization:  
utilization_black_feature  utilization_black_product  
utilization_white_feature  utilization_white_product
```

config.py: 主配置文件，按照实际的环境和需求来配置。

LM_LICENSE_FILE: 配置 license server 列表，用于生成“LM_LICENSE_FILE”的环境变量。

project/: 和项目 (project) 相关的配置，用于 COST 页计费分摊。

utilization/: 和利用率 (utilization) 相关的配置，用于 UTILIZATION 页。

cost/: 和计费 (COST) 相关的配置，用于 COST 页。

others/: 其它的一些配置，包括 product/feature 映射关系、license log 路径等配置。

如果觉得本章节的内容较难理解，可以读完第四章以后再回来读工具配置的方法。

3.3.1 config/config.py

安装目录下主要的配置文件为 config/config.py，用于配置工具的基本设置。

参数均设有默认值，但是部分参数还需要按照用户实际情况重新配置。

```
# Specify EDA license administrators.  
administrators = "ALL"  
  
# Specify lmstat path, example
```

```
"/eda/synopsys/scl/2021.03/linux64/bin/lmstat".
lmstat_path =
"/ic/data/usr/liyanqing.1987/tools/licenseMonitor/tools/lmstat"

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

# Specify the database directory where to save sample data.
db_path = "/ic/data/usr/liyanqing.1987/tools/licenseMonitor/db"

# Specify which are the primary factors when getting project
information.
# It could be one or several items between
"user/execute_host/submit_host".
project_primary_factors = "user execute_host submit_host"

# Max record number when searching license log.
max_record_num = 1000

# The time interval to fresh license information automatically,
unit is "second", default is 300 seconds.
fresh_interval = 300
```

administrators： 指定 licenseMonitor 管理员， 默认为“ALL”， 可以指定多个用户，中间用空格隔开。

licenseMonitor 管理员有两个特权，一是可以访问 licenseMonitor 配置好的全量的 LM_LICENSE_FILE 设置（普通用户则只能从当前环境中读取 LM_LICENSE_FILE 环境变量），二是可以访问 CURVE/UTILIZATION/COST 页面（普通用户无法访问）。

lmstat_path： 指定 lmstat 工具的路径， 默认为 licenseMonitor 自带的 lmstat。

lmstat 工具用于获取 EDA license 的信息， licenseMonitor 自带的为 v11.16.4.0 版本，如果存在兼容性问题，可以替换为 EDA 工具目录中的 lmstat。

lmstat_bsub_command： 执行 lmstat 命令时前置的 bsub 命令， 默认为“”， 即本机运行，不 bsub 出去。

因为系统配置限制，部分服务器（比如 login server）上可能并不能执行 lmstat 命令，只能把 lmstat 采用 bsub 的方式丢到 computing server 上运行，此时需要在此处配置 bsub 的具体指令。

如果执行机器不能执行 lmstat 的情况下，不配置 lmstat_bsub_command 会导致 licenseMonitor 的主界面上信息缺失或者空白。

db_path：指定数据库路径， 默认在 licenseMonitor 的安装路径下，但是建议指定到专门的数据路径，并且请确保此路径存在，且执行采样的账号具有写权限。

project_primary_factors：用于 COST 页的费用分摊功能，EDA license 是哪个项目使用，需要依赖一些关键因素来区分，区分的默认顺序是“user execute_host submit_host”。

按照默认的项目首要因素，先按照 feature 的 user 判断这个 feature 是哪个 project 使用的，如果不能判断，再依次按照 execute_host 和 submit_host 来判断，仍然不能确定 project 的，则一律归入“others”项目。

max_record_num：用于 license log 信息检索功能，指定检索时显示的最大记录，默认值为 1000。

fresh_interval：licenseMonitor 被动更新 EDA license 信息的时间间隔下限，默认为 300 (s)，即 5 分钟。

fresh_interval 是控制上次获取 EDA license 信息后，多长时间内不要重复获取这个信息。fresh_interval 的值不要设置的太低，以防止频繁连接 license server 导致程序卡顿。

下面是一个根据实际环境修改过的配置。

```
# Specify EDA license administrators.
administrators = "root liyanqing.1987"

# Specify lmstat path, example
"/eda/synopsys/scl/2021.03/linux64/bin/lmstat".
lmstat_path =
"/ic/data/usr/liyanqing.1987/tools/licenseMonitor/tools/lmstat"

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

# Specify the database directory where to save sample data.
db_path = "/ic/data/usr/liyanqing.1987/database/licenseMonitor/db"

# Specify which are the primary factors when getting project
information.
# It could be one or several items between
"user/execute_host/submit_host".
project_primary_factors = "user execute_host submit_host"

# Max record number when searching license log.
```

```
max_record_num = 1000

# The time interval to fresh license information automatically,
unit is "second", default is 300 seconds.
fresh_interval = 300
```

和默认的设置相比，修改了如下项目。

- 将“root”和“liyanqing.1987”增加为管理员。
- 设置了 lmstatbsub_command 为“bsub -q normal -ls”，因为在测试机器上并不能执行 lmstat (iptables 禁止访问)，且用户可以使用 LSF 的 normal 队列，故将 lmstat 命令 bsub 到 normal 队列中执行。
- 将 db_path 从 software 目录修改到了 database 目录。

3.3.2 config/LM_LICENSE_FILE

lmstat 工具用于获取 EDA license 的信息，它依赖于 LM_LICENSE_FILE 这一环境变量获取 license server 的信息。普通用户主要关注自己在用的 license server 上的信息，所以执行 license_monitor 的时候会去用户的环境变量中获取 LM_LICENSE_FILE 的设置，而管理员则更关注全量的 license server 信息，所以依赖于 config/LM_LICENSE_FILE 这个文件来配置环境全量的 license server 信息。

LM_LICENSE_FILE 的配置格式如下，每行一个 license server 设置，写成 <port>@<hostname> 的样式，多个 license server 则写成多行。

```
1055@ic-lic03
1717@ic-lic03
27020@ic-lic03
```

如果用户采用 module files 来管理 EDA 环境，那么可以采用 tools/gen_LM_LICENSE_FILE 这一工具来自动生成 config/LM_LICENSE_FILE 文件，具体使用方法请参照第五章。

3.3.3 config/project

product 目录用来存放项目相关的信息，用于 COST 页的费用分摊功能。

其中 project_submit_host/project_execute_host/project_user 分别用于从 EDA

license 的 submit_host/execute_host/user 的维度来区分其项目归属，所以需要澄清几个基本概念。下面是用 lmstat 工具抓取的一条 EDA license feature 使用信息。

```
liyanqing.1987 n232-132-013 n232-134-067:2874308_1 (v2022.04) (ic-  
lic03/1717 1324), start Sun 6/25 17:16
```

其中：

liyanqing.1987 : user, 用户名。

n232-132-013 : execute_host, 执行 EDA 工具的机器。

n232-124-067 : submit_host, 这是一台 login server, 并不能执行 EDA 工具, 任务通过 bsub 的方式丢到 computing server n232-132-013 上运行。

判断这次使用产生的费用应该分给哪个项目的时候，我们根据 config.py 中的如下设置来依次判断。

```
project_primary_factors = "user  execute_host  submit_host"
```

首先根据 config/project/project_user 判断 user “liyanqing.1987”属于哪个项目；如果不能判断，那么再根据 config/project/project_execute_host 判断 execute host “n232-132-013”属于哪个项目；如果还是不能判断，那么再根据 config/project/project_submit_host 判断 submit host “n232-124-067”属于哪个项目。如果最终都不能判断，则这笔费用被归入“others”，或者不参与费用分摊。

3.3.3.1 config/project/project_list

project_list 文件用于指明参加费用分摊的项目有哪些。

下面是一个配置示例。

```
proj_a  
proj_b  
proj_c  
proj_d
```

每行仅填写一个项目，项目名中不能有空格，多个项目填写多行。

3.3.3.2 config/project_user

project_user 用于基于 EDA license 的 user 来判断项目信息，配置方法跟 project_submit_host 一致。

下面是一个配置示例。

```
zhangsan : proj_a(0.3) proj_b(0.7)
lisi : proj_c
wangwu : proj_d
...
```

以下面这行为例。

```
zhangsan : proj_a(0.3) proj_b(0.7)
```

意为用户 zhangsan 同时归属于 proj_a (占比 30%) 和 proj_b (占比 70%)，user 为 zhangsan 的 license feature 使用所产生的费用，由 proj_a 承担 30%，由 proj_b 承担 70%。

3.3.3.3 config/project/project_execute_host

project_submit_host 用于基于 EDA license 的 execute host 来判断项目信息，配置方法跟 project_submit_host 一致。

下面是一个配置示例。

```
host1 : proj_a(0.333) proj_b(0.333) proj_c(0.334)
host2 : proj_b(0.5) proj_c(0.5)
host3 : proj_d
...
```

以下面这行为例。

```
host1 : proj_a(0.333) proj_b(0.333) proj_c(0.334)
```

意为 host1 同时归属于 proj_a (占比 33.3%)、proj_b (占比 33.3%) 和 proj_c (占比 33.4%)，execute host 为 host1 的 license feature 使用所产生的费用，由 proj_a 承担 33.3%，由 proj_b 承担 33.3%，由 proj_c 承担 33.4%。

3.3.3.4 config/project/project_submit_host

project_submit_host 用于基于 EDA license 的 submit host 来判断项目信息。

一个 submit host 可以属于一个 project, 也可以属于多个 project, 如果属于多个 project, 必须为每一个 project 设置一个分摊比例, 且总的分摊系数必须为 1。

下面是一个配置示例。

```
host1 : proj_a(0.3) proj_b(0.7)
host2 : proj_c
host3 : proj_a(0.333) proj_b(0.333) proj_d(0.334)
```

以下面这行为例。

```
host1 : proj_a(0.3) proj_b(0.7)
```

意为 host1 同时归属于 proj_a (占比 30%) 和 proj_b (占比 70%), submit host 为 host1 的 license feature 使用所产生的费用, 由 proj_a 承担 30%, 由 proj_b 承担 70%。

3.3.4 config/utilization

utilization 目录用来存放 UTILIZATION 页面相关的配置, 当前主要有 4 种类型的配置。

- utilization_black_feature : 指定 feature 的黑名单, 支持正则匹配, 黑名单中的 feature 将会在统计 utilization 数据的时候删掉。
- utilization_black_product : 指定 product 的黑名单, 支持正则匹配, 在查看 product 维度的 utilization 信息时, 黑名单中的 product 将会被删掉。
- utilization_white_feature : 指定 feature 的白名单, 支持正则匹配, 仅有白名单中的 feature 会用来统计 utilization 数据。
- utilization_wihte_product : 指定 product 的白名单, 支持正则匹配, 在查看 product 维度的 utilization 信息时, 仅有白名单中的 product 会用来统计 utilization 数据。

3.3.4.1 config/utilization/utilization_black_feature

每行填写一个 feature (或者 feature 的正则匹配), 多个 feature 填写多行。

下面是一个配置示例。

VCSMXRunTime_Net
VC-RDC-BASE

3.3.4.2 config/utilization/utilization_black_product

每行填写一个 product (或者 product 的正则匹配), 多个 product 填写多行。

下面是一个配置示例。

```
DWC .*
VC VIP .*
```

3.3.4.3 config/utilization/utilization_white_feature

格式同 3.3.5.1 utilization_black_feature。

3.3.4.4 config/utilization/utilization_white_product

格式同 3.3.5.2 utiliaztion_black_product

3.3.5 config/cost

cost 目录用来存放 COST 页面相关的配置, 当前主要有 4 种类型的配置。

- cost_black_feature : 指定 feature 的黑名单, 支持正则匹配, 黑名单中的 feature 将会在统计 cost 数据的时候删掉。
- cost_black_product : 指定 product 的黑名单, 支持正则匹配, 在查看 product 维度的 cost 信息时, 黑名单中的 product 将会被删掉。
- cost_white_feature : 指定 feature 的白名单, 支持正则匹配, 仅有白名单中的 feature 会用来统计 cost 数据。
- cost_wihte_product : 指定 product 的白名单, 支持正则匹配, 在查看 product 维度的 cost 信息时, 仅有白名单中的 product 会用来统计 cost 数据。

3.3.5.1 config/cost/cost_black_feature

格式同 3.3.5.1 utilization_black_feature

3.3.5.2 config/cost/cost_black_product

格式同 3.3.5.2 utilization_black_product

3.3.5.3 config/cost/cost_white_feature

格式同 3.3.5.3 utilization_white_feature

3.3.5.4 config/cost/cost_white_product

格式同 3.3.5.4 utilization_white_product

3.3.6 config/others

config/others 目录用来存放其它的配置文件。

3.3.6.1 config/others/product_feature.yaml

product_feature.yaml 用于 UTILIZATION/COST 页面的 product 信息维度展示。

EDA license 管理和信息统计的最小维度是 feature，但是实际上公司采购的时候是按照 product 的维度来采购的，一般每个 product 会包含一个或几个 feature，这样在做使用率分析和费用分摊统计的时候，不同用户就会面临 feature-product 视角的 gap，product_feature.yaml 这个映射关系文件就是将相关数据在 product 和 feature 两个维度做转换。

product_feature.yaml 用于配置每个 vendor 的 product 和 feature 的映射关系，样式如下。

```
ansyslmd:  
  SC_WG:  
    - Ansys SeaScape Worker Group 256 Pack  
  anshpc_pack:  
    - Ansys Electronics Desktop (HPCPack)  
    ...
```

将 yaml 转换为 python 的 dict 结构后，可以看到其数据结构样式为：

```
{<vendor>: {<feature>: [<product1>, <product2>, ...]}}
```

product_feature.yaml 可以借助 tools/get_product_feature_relationship 和 tools/config_product_feature_relationship 两个脚本生成和配置，详情请参照第五章。

3.3.6.2 config/others/license_log.yaml

license_log.yaml 用于配置每个 license server 及其 license log 的对应关系，其样式如下。

```
1055@ic-lic03: /ic/data/CAD/lic/logs/ansys_ic-lic03.log  
1717@ic-lic03: /ic/data/CAD/lic/mentor/ic-lic03/saltd.log  
27020@ic-lic03: /ic/data/CAD/lic/logs/synopsys_ic-lic03.log
```

这个配置文件可以由 tools/get_license_log 生成，详情请参照第五章。

3.3.6.3 config/others/feature_record_on_license_log.yaml

feature_record_on_license_log.yaml 主要用于 UTILIZATION/COST 页面在开启“Enable Utilization Log Search”/“Enable Cost Log Search”情况下的高级检索功能。

feature_record_on_license_log.yaml 用来保存所有所有 license feature 在 license log 中 grep 到的记录，样式如下。

```
caldesignrev:  
mgcld:  
  1717@ic-lic03:  
    - host: n232-132-146  
      user: zhangsan  
    - host: n232-132-209  
      user: lishi
```

feature_record_on_license_log.yaml 可以由 tools/collect_feature_record_from_license_log 生成，详情请参照第五章。

它的生成依赖于 config/others/license_log.yaml 的存在，所以需要先生成 license_log.yaml，再生成 feature_record_on_license_log.yaml。

四、工具使用

4.1 数据采集 license_sample

4.1.1 帮助信息

license_sample 用于采集 utilization 信息和 usage 信息，其使用帮助如下。

```
[root@ic-admin2 licenseMonitor]# bin/license_sample -h
usage: license_sample.py [-h] [-u] [-U]

optional arguments:
  -h, --help            show this help message and exit
  -u, --usage           Sample license feature usage info.
  -U, --utilization    Sample license feature utilization info.
```

--help: 打印帮助信息。

--usage: 采集 EDA license feature 的使用记录。

--utilization: 采集 EDA license feature 的使用率信息，比如某个 license server 上有某个 vendor 的 feature A，feature A 的数目为 100，采样时使用了 87 个，瞬时使用率即为 87%。因为 cost 费用分摊功能需要依赖 utilization 的数据库，所以如果需要使用 COST 页的费用分摊功能，那么采集 usage 信息的时候，务必同步采集 utilization 信息。

4.1.2 采样示例

手工执行脚本 license_sample 进行采样。

```
[root@ic-admin2 licenseMonitor]# bin/license_sample -u -U
>>> Sampling license usage information ...
>>> Detect project setting ...
      Create directory
"/ic/data/usr/liyanqing.1987/tools/licenseMonitor/db/project_setting".
      Create directory
"/ic/data/usr/liyanqing.1987/tools/licenseMonitor/db/project_setting/20231219182105".
      Copy
```

```
"/ic/data/usr/liyanqing.1987/tools/licenseMonitor/config/project/project_list" into directory
"/ic/data/usr/liyanqing.1987/tools/licenseMonitor/db/project_setting/20231219182105".
    Copy
"/ic/data/usr/liyanqing.1987/tools/licenseMonitor/config/project/project_execute_host" into directory
"/ic/data/usr/liyanqing.1987/tools/licenseMonitor/db/project_setting/20231219182105".
    Copy
"/ic/data/usr/liyanqing.1987/tools/licenseMonitor/config/project/project_submit_host" into directory
"/ic/data/usr/liyanqing.1987/tools/licenseMonitor/db/project_setting/20231219182105".
    Copy
"/ic/data/usr/liyanqing.1987/tools/licenseMonitor/config/project/project_user" into directory
"/ic/data/usr/liyanqing.1987/tools/licenseMonitor/db/project_setting/20231219182105".
>>> Sampling usage info ...
    Create directory
"/ic/data/usr/liyanqing.1987/tools/licenseMonitor/db/license_server/1055@ic-lic03/ansyslmd".
>>> Sampling utilization info ...
    Sampling utilization info for "1055@ic-lic03/ansyslmd/alinks_gui" ...
...
>>> Counting utilization (day average) info ...
    Sampling usage info for "1717@ic-lic03/mgcld/caldfmproperty" ...
    Sampling usage info for "1717@ic-lic03/mgcld/calibreadp" ...
...
    Counting utilization (day average) info for "27020@ic-lic03/snpslmd/verilog" ...
    Counting utilization (day average) info for "27020@ic-lic03/snpslmd/vhdl" ...
```

注意，license_sample 采用多线程采样，所以输出信息顺序有些混乱。

license_sample 依次干了如下几件事情。

- 检查 config/project 下的配置文件，如果没有存储在<db_path>/project_settings 下，或者存储过但最新的设置有变更，则重新拷贝一份过来（目录带时间戳）。

- 采集 license feature 的 usage 信息并存储。
- 根据 license feature 的 usage 信息计算 utilization 信息并存储。
- 按天汇聚 utilization 信息并存储。

我们推荐用 crontab 来定时采样（Jenkins 类似），推荐采样间隔为 5 分钟。下面是一个示例。（crontab -e）

```
[root@ic-admin2 licenseMonitor]# crontab -l
SHELL=/bin/bash
PATH=/ic/software/tools/anaconda/Anaconda3-
2021.05/bin:/ic/software/cad_tools/bin:/usr/local/bin:/bin:/usr/bi
n:/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 licenseMonitor, sample every 5 minutes.
*/5 * * * *
/ic/data/usr/liyanqing.1987/tools/licenseMonitor/bin/license_sampl
e -u -U
```

4.1.3 数据库

第一次采样后，生成的数据库目录结构如下。

```
[root@ic-admin2 ~]# cd
/ic/data/usr/liyanqing.1987/database/licenseMonitor/db/
[root@ic-admin2 db]# tree
.
├── license_server
│   ├── 1055@ic-lic03
│   │   └── ansyslmd
│   │       ├── usage.db
│   │       ├── utilization_day.db
```

```
    └── utilization.db
  └── 1717@ic-lic03
    ├── mgcld
    │   ├── usage.db
    │   ├── utilization_day.db
    │   └── utilization.db
    └── saltd
      ├── usage.db
      ├── utilization_day.db
      └── utilization.db
  └── 27020@ic-lic03
    └── snpslmd
      ├── usage.db
      ├── utilization_day.db
      └── utilization.db
└── project_setting
  └── 20231219182633
    ├── project_execute_host
    ├── project_list
    ├── project_submit_host
    └── project_user
```

10 directories, 16 files

其数据如下：

- License_server: 存放 usage 和 utilization 信息。
 - 1055@ic-lic03: 存放具体 license_server 上的信息。
 - ansyslmd: 存放具体 vendor_daemon 上的信息。
 - usage.db: 存放 license feature 使用的细节信息。
 - utilization.db: 存放 license feature 使用率信息。
 - utilization_day.db: 存放 utilization.db 按天汇聚后的信息。
- project_setting: 存放 config/project 的数据，会保存每一个变更版本。
 - 20231219182633: 将每一次的 project 配置目录保存为时间戳的形式（每次有变更即保存），便于按时间段统计计费分摊情况。
 - project_list: 存放项目列表信息。
 - project_submit_host: 存放每个 submit_host 的项目归属情况。
 - project_execute_host: 存放每个 execute_host 的项目归属情况。

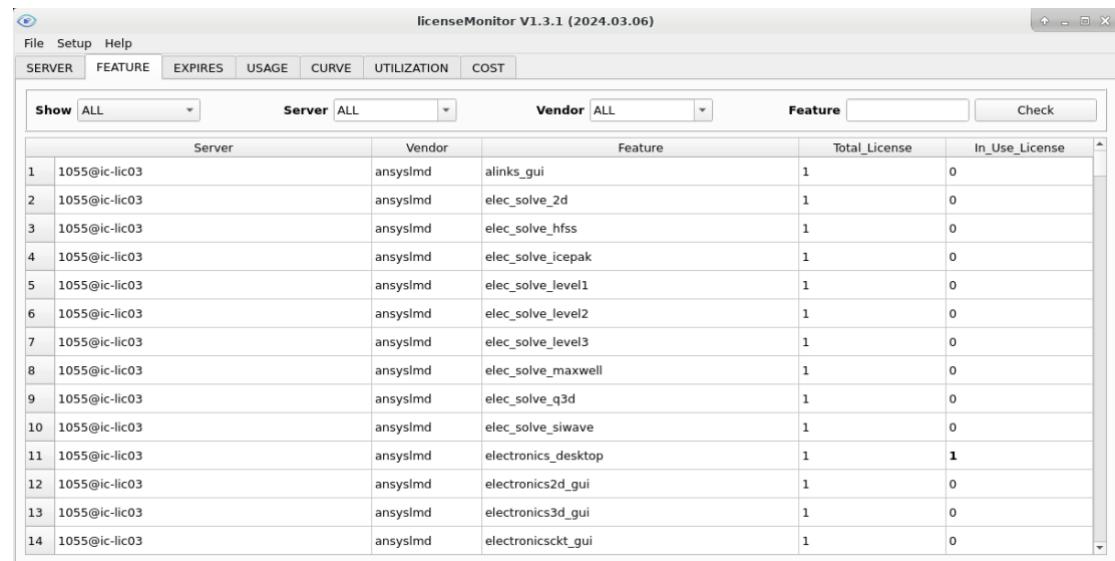
- project_user: 存放每个 user 的项目归属情况。

4.2 数据展示 license_monitor

4.2.1 工具载入

licenseMonitor 的主程序是 license_monitor，位于 licenseMonitor 安装目录下的 bin/license_monitor，安装后可以直接引用。如果使用环境中配置了 modules，则可以通过 module load 的方式引用 license_monitor。

```
[liyanqing.1987@n232-135-067 licenseMonitor]$ bin/license_monitor
* [2024-03-08 18:42:54] Loading license info, please wait a
moment ...
```



The screenshot shows the licenseMonitor application window. The title bar reads "licenseMonitor V1.3.1 (2024.03.06)". The menu bar includes File, Setup, Help. Below the menu is a toolbar with tabs: SERVER, FEATURE, EXPIRES, USAGE, CURVE, UTILIZATION, COST. There are also buttons for Show (set to ALL), Server (set to ALL), Vendor (set to ALL), Feature (empty), and Check. The main area is a table with the following columns: Server, Vendor, Feature, Total_License, and In_Use_License. The table contains 14 rows of data, each corresponding to a server entry from 1 to 14. The "In_Use_License" column shows values 0 or 1.

	Server	Vendor	Feature	Total_License	In_Use_License
1	1055@ic-lic03	ansyslmd	alinks_gui	1	0
2	1055@ic-lic03	ansyslmd	elec_solve_2d	1	0
3	1055@ic-lic03	ansyslmd	elec_solve_hfss	1	0
4	1055@ic-lic03	ansyslmd	elec_solve_icepak	1	0
5	1055@ic-lic03	ansyslmd	elec_solve_level1	1	0
6	1055@ic-lic03	ansyslmd	elec_solve_level2	1	0
7	1055@ic-lic03	ansyslmd	elec_solve_level3	1	0
8	1055@ic-lic03	ansyslmd	elec_solve_maxwell	1	0
9	1055@ic-lic03	ansyslmd	elec_solve_q3d	1	0
10	1055@ic-lic03	ansyslmd	elec_solve_siwave	1	0
11	1055@ic-lic03	ansyslmd	electronics_desktop	1	1
12	1055@ic-lic03	ansyslmd	electronics2d_gui	1	0
13	1055@ic-lic03	ansyslmd	electronics3d_gui	1	0
14	1055@ic-lic03	ansyslmd	electronicsckt_gui	1	0

license_monitor 在启动的过程中，需要读取当前 license_server 的情况，这需要一定的时间，尤其是在 license_server 较多的情况下。

4.2.2 帮助信息

license_monitor 的帮助信息如下。

```
[liyanqing.1987@n232-135-067 licenseMonitor]$ bin/license_monitor
-h
usage: license_monitor.py [-h] [-f FEATURE] [-u USER] [-t
{SERVER,FEATURE,EXPIRES,USAGE,UTILIZATION,COST}]

optional arguments:
-h, --help            show this help message and exit
-f FEATURE, --feature FEATURE
                      Specify license feature which you want to
see on "FEATURE/EXPIRES/USAGE/UTILIZATION/COST" tab.
-u USER, --user USER  Specify the user on "USAGE" tab.
-t {SERVER,FEATURE,EXPIRES,USAGE,UTILIZATION,COST}, --tab
{SERVER,FEATURE,EXPIRES,USAGE,UTILIZATION,COST}
                      Specify current tab, default is "FEATURE"
tab.
```

--help : 打印帮助信息。

--feature : 指定 FEATURE/EXPIRES/USAGE/UTILIZATION/COST 界面中要检索的 faeature，也可以在图形界面中指定。

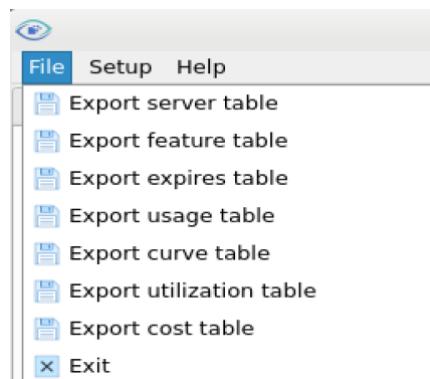
--user : 指定 USAGE 界面中要检索的 user，也可以在图形界面中指定。

--tab : 指定打开时的 tab 页，默认是 FEATURE 页。

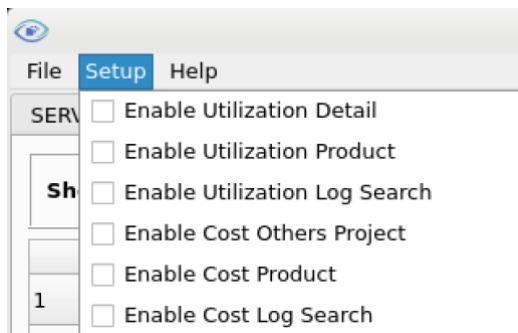
4.2.3 菜单栏

license_monitor 菜单栏包括 File, Setup, Help 三部分。

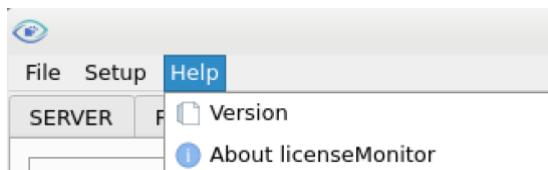
File: 包含 Export * table 功能和 Exit 功能。



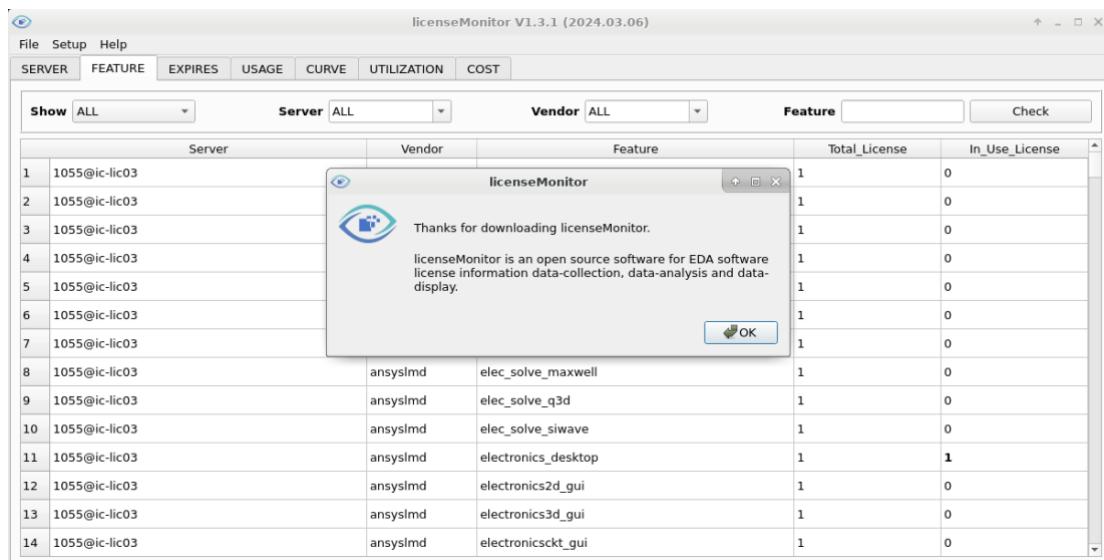
Setup: 包含 Enable *等几个复选框。



Help: 包含“Version”和“About ”两个信息项。



Help 菜单栏中的“About licenseMonitor”内容如下。



4.2.4 SERVER 页

打开 license_monitor 图形界面后，可以在 SERVER 页查看 license server 的状态和版本等信息。

licenseMonitor V1.3.1 (2024.03.06)						
File		Setup		Help		
SERVER		FEATURE	EXPIRES	USAGE	CURVE	UTILIZATION
1	1055@ic-lic03		UP	v11.19.4	/ic/data/CAD/lic/ansys/ansys_ic-lic03.lic	ansyslmd UP v11.19.4
2	1717@ic-lic03		UP	v11.19.0	/ic/data/CAD/lic/mentor/ic-lic03/...	saltd UP v11.19.0
3	1717@ic-lic03		UP	v11.19.0	/ic/data/CAD/lic/mentor/ic-lic03/...	mgclid UP v11.19.0
4	27003@ic-lic05		UP	v11.16.4	/ic/data/CAD/lic/n2-096-050/intel_mentor/...	alterad UP v11.18.2
5	27003@ic-lic05		UP	v11.16.4	/ic/data/CAD/lic/n2-096-050/intel_mentor/...	mgclid DOWN
6	27020@ic-lic03		UP	v11.19.4	/ic/data/CAD/lic/synopsys/synopsys_ic-...	snpsslmd UP v11.19.4

我们特意加入了“ic-lic05”这个 license server 来演示异常情况，可以看到，当 Server_Status 或者 Vendor_Status 不是 UP 的情况下，相关信息行会被标红，以提醒管理员及时处理。

4.2.5 FEATURE 页

打开 license_monitor 图形界面后，可以在 FEATURE 页查看 license feature 的信息，包括总量和在用量。

licenseMonitor V1.3.1 (2024.03.06)					
File		Setup		Help	
SERVER		FEATURE	EXPIRES	USAGE	CURVE
Show	All	Server	All	Vendor	All
1	1055@ic-lic03	ansyslmd	alinks_gui	1	0
2	1055@ic-lic03	ansyslmd	elec_solve_2d	1	0
3	1055@ic-lic03	ansyslmd	elec_solve_hfss	1	0
4	1055@ic-lic03	ansyslmd	elec_solve_icepak	1	0
5	1055@ic-lic03	ansyslmd	elec_solve_level1	1	0
6	1055@ic-lic03	ansyslmd	elec_solve_level2	1	0
7	1055@ic-lic03	ansyslmd	elec_solve_level3	1	0
8	1055@ic-lic03	ansyslmd	elec_solve_maxwell	1	0
9	1055@ic-lic03	ansyslmd	elec_solve_q3d	1	0
10	1055@ic-lic03	ansyslmd	elec_solve_siwave	1	0
11	1055@ic-lic03	ansyslmd	electronics_desktop	1	1
12	1055@ic-lic03	ansyslmd	electronics2d_gui	1	0
13	1055@ic-lic03	ansyslmd	electronics3d_gui	1	0
14	1055@ic-lic03	ansyslmd	electronicsckt_gui	1	0

可以按照“**Show**”来筛选，有“**ALL**”和“**IN_USE**”两个选项，前者会显示所有的 license feature，后者仅显示在用的 license feature。

可以按照“**Server**”来筛选。

可以按照“**Vendor**”来筛选。

可以按照“**Feature**”来筛选，此处的 Feature 支持模糊检索。比如输入“calibrepercadv”，可以精确地检索到“calibrepercadv”这个 license。

licenseMonitor V1.3.1 (2024.03.06)				
File	Setup	Help	SERVER	FEATURE
Show ALL		Server ALL	Vendor ALL	Feature calibrepercadv
Server	Vendor	Feature	Total_License	In_Use_License
1 1717@ic-lic03	mgcld	calibrepercadv	4	4

如果输入“calib”，除了“calibrepercadv”外，还可以检索到所有和 calib 相似的 license。

licenseMonitor V1.3.1 (2024.03.06)				
File	Setup	Help	SERVER	FEATURE
Show ALL		Server ALL	Vendor ALL	Feature calib
Server	Vendor	Feature	Total_License	In_Use_License
1 1717@ic-lic03	mgcld	calibreadp	4	0
2 1717@ic-lic03	mgcld	calibredrc	10	2
3 1717@ic-lic03	mgcld	calibrehdrc	10	0
4 1717@ic-lic03	mgcld	calibrehlvs	10	0
5 1717@ic-lic03	mgcld	calibrelvs	10	2
6 1717@ic-lic03	mgcld	calibreperc	4	0
7 1717@ic-lic03	mgcld	calibrepercadv	4	4
8 1717@ic-lic03	mgcld	calibrepercsp	4	0
9 1717@ic-lic03	mgcld	calibreqdb	4	1
10 1717@ic-lic03	mgcld	calibrexrc	4	0
11 1717@ic-lic03	mgcld	calibrexrctoact	4	0

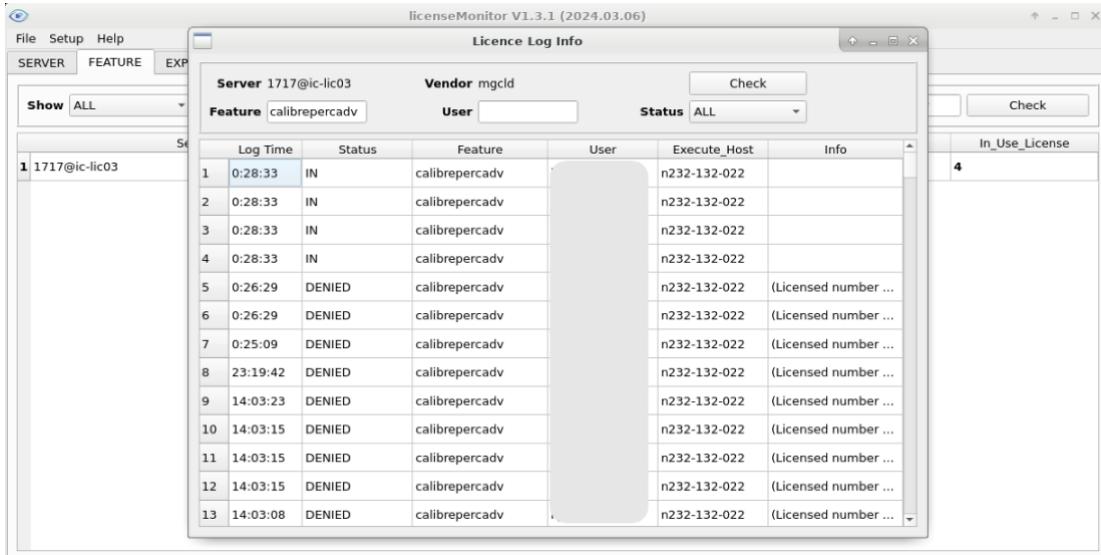
如果“In_Use_License”的数目大于 0，点击对应的数值，可以跳到 USAGE 页查看这个 license feature 的 usage 细节，比如点击“calibrepercadv”对应的 In_Use_License 数字，会跳转到如下界面。

	Server	Vendor	Feature	User	Submit_Host	Execute_Host	Num	Version	Start_Time
1	1717@ic-lic03	mgclD	calibrepercadv		n232-134-066	n232-132-197	1	v2022.11	2024-03-08 17:02
2	1717@ic-lic03	mgclD	calibrepercadv		n232-134-066	n232-132-197	1	v2022.11	2024-03-08 17:02
3	1717@ic-lic03	mgclD	calibrepercadv		n232-134-066	n232-132-197	1	v2022.11	2024-03-08 17:02
4	1717@ic-lic03	mgclD	calibrepercadv		n232-134-066	n232-132-197	1	v2022.11	2024-03-08 17:02

如果想检索 feature 对应的 checkin/checkout 记录，在 FEATURE 页 Feature 单元格右击，点击“View License Log”。

Show	Server	Vendor	Feature	View License Log	Total_License	In_Use_License
ALL	1717@ic-lic03	mgclD	calibrepercadv	4	4	4

此时可以在新弹出的界面中展示出 license log 中相应 feature 的操作记录（默认最多 1000 行）。



请确保当前用户在当前机器上，可以直接访问指定 license server 的 log file，或者有权限 ssh 到 license server 上访问 license file，否则无法获取有效信息。

另外，如果 license log 尺寸非常大的情况（几千万/上亿行），也非常难在 timeout 之前抓取 license log 中的信息。

4.2.6 EXPIRES 页

打开 license_monitor 图形界面后，可以在 EXPIRES 页查看 license feature 的过期信息。

Server	Vendor	Feature	Version	License_Num	Expires	
141 1055@ic-lic03	ansyslmd	Process_Configurator	9999.9999	1	2023-02-22	已过期
142 1055@ic-lic03	ansyslmd	Process_Configurator	9999.9999	1	2023-01-22	
143 1055@ic-lic03	ansyslmd	Tech_Viewer	9999.9999	3	2023-06-30	
144 1055@ic-lic03	ansyslmd	Tech_Viewer	9999.9999	3	2023-04-30	
145 1055@ic-lic03	ansyslmd	Tech_Viewer	9999.9999	2	2023-03-28	
146 1055@ic-lic03	ansyslmd	Tech_Viewer	9999.9999	1	2025-12-15	未过期
147 1055@ic-lic03	ansyslmd	Tech_Viewer	9999.9999	1	2023-02-22	
148 1055@ic-lic03	ansyslmd	Tech_Viewer	9999.9999	1	2023-01-22	
149 1717@ic-lic03	mgcld	caldesignrev	2022.030	32	2022-03-24	
150 1717@ic-lic03	mgcld	caldesignrev	2022.090	1	2022-09-09	
151 1717@ic-lic03	mgcld	caldesignrev	2024.030	4	2024-03-14	即将过期
152 1717@ic-lic03	mgcld	caldesignrev	2024.030	4	2022-04-29	
153 1717@ic-lic03	mgcld	caldfmproperty	2022.030	32	2022-03-24	
154 1717@ic-lic03	mgcld	caldfmproperty	2022.090	1	2022-09-09	

筛选功能同 FEATURE 页相同，其中 **Show** 选项包含“Expired”、“Nearly_Expired”和“Unexpired”三种选择。

另外，Expires 项会按照不同的颜色来显示 feature 不同的 expires 状态：

黑色：未过期。

红色：两周内过期。

灰色：已过期。

4.2.7 USAGE 页

打开 license_monitor 图形界面后，可以在 USAGE 页查看 license feature 的使用详情。

Server	Vendor	Feature	User	Submit_Host	Execute_Host	Num	Version	Start_Time
1 1055@ic-lic03	ansyslmd	electronics_desktop		N/A	n232-130-198	1	v2023.0505	2024-03-07 19:01
2 1717@ic-lic03	mgclld	caldesignrev		n232-135-013	n232-135-140	1	v2022.04	2024-02-28 15:56
3 1717@ic-lic03	mgclld	caldesignrev		n232-135-012	n232-132-029	1	v2022.11	2024-03-01 16:11
4 1717@ic-lic03	mgclld	calibredrv		N/A	N/A	1	N/A	RESERVATION
5 1717@ic-lic03	mgclld	calibredrv		N/A	N/A	1	N/A	RESERVATION
6 1717@ic-lic03	mgclld	calibredrv		N/A	N/A	1	N/A	RESERVATION
7 1717@ic-lic03	mgclld	calibredrv		N/A	N/A	1	N/A	RESERVATION
8 1717@ic-lic03	mgclld	calibrepercadv		n232-134-066	n232-132-197	1	v2022.11	2024-03-08 17:02
9 1717@ic-lic03	mgclld	calibrepercadv		n232-134-066	n232-132-197	1	v2022.11	2024-03-08 17:02
10 1717@ic-lic03	mgclld	calibrepercadv		n232-134-066	n232-132-197	1	v2022.11	2024-03-08 17:02
11 1717@ic-lic03	mgclld	calibrepercadv		n232-134-066	n232-132-197	1	v2022.11	2024-03-08 17:02
12 1717@ic-lic03	mgclld	calibreqdb		n232-135-012	n232-130-087	1	v2022.11	2024-03-05 17:41
13 1717@ic-lic03	mgclld	calmultipattern		N/A	N/A	1	N/A	RESERVATION

可以按照**Server**来筛选。

可以按照**Vendor**来筛选。

可以按照**Feature**来筛选，此处的 Feature 支持模糊检索。比如输入“Verdi”，可以精确地检索到“Verdi”这个 license，如果输入“verdi”，除了“Verdi”外，还可以检索到所有和 verdi 相似的 license。

可以按照**Submit Host**来筛选。

可以按照**Execute Host**来筛选。

可以按照**User**来筛选，此处的 User 支持模糊检索。比如输入“zhangsan.three”，可以精确地检索到“zhangsan.three”这个用户的 license 使用情况，如果输入

“zhangsan”，除了“zhangsan.three”外，还可以检索到所有和 zhangsan 名字相似的用户的 license 使用情况。

The screenshot shows the licenseMonitor V1.3.1 interface with the 'FEATURE' tab selected. The search criteria are set to 'Feature: calibrepercadv' and 'User: huac'. The results table displays four rows of license usage data:

Server	Vendor	Feature	User	Submit_Host	Execute_Host	Num	Version	Start_Time
1 1717@ic-lic03	mgcl	calibrepercadv	huac	n232-134-066	n232-132-197	1	v2022.11	2024-03-08 17:02
2 1717@ic-lic03	mgcl	calibrepercadv	huac	n232-134-066	n232-132-197	1	v2022.11	2024-03-08 17:02
3 1717@ic-lic03	mgcl	calibrepercadv	huac	n232-134-066	n232-132-197	1	v2022.11	2024-03-08 17:02
4 1717@ic-lic03	mgcl	calibrepercadv	huac	n232-134-066	n232-132-197	1	v2022.11	2024-03-08 17:02

如果某个 feature 的 Start_Time 早于 3 天前，此处会显示为红色，警示占用过久的 license feature。

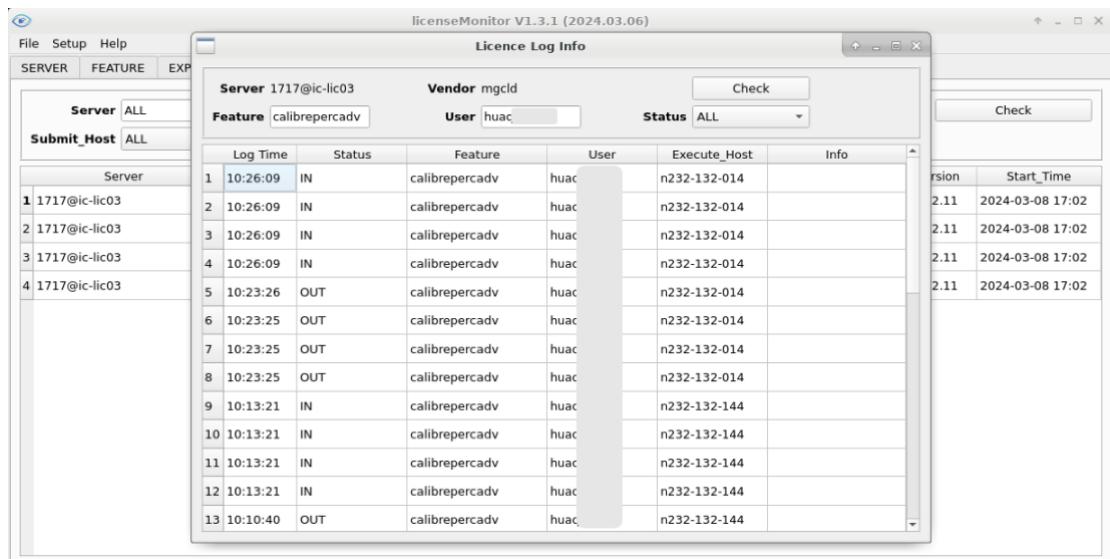
如果想检索 feature 对应的 checkin/checkout 记录，在 FEATURE 页 Feature 单元格或者 User 单元格右击，点击“View License Log”。

The screenshot shows the same licenseMonitor interface as above, but with a right-click context menu open over the 'Feature' column of the first row. The menu item 'View License Log' is highlighted with a blue background.

The results table remains the same as in the previous screenshot:

Server	Vendor	Feature	User	Submit_Host	Execute_Host	Num	Version	Start_Time
1 1717@ic-lic03	mgcl	calib	huac	n232-134-066	n232-132-197	1	v2022.11	2024-03-08 17:02
2 1717@ic-lic03	mgcl	calibrepercadv	huac	n232-134-066	n232-132-197	1	v2022.11	2024-03-08 17:02
3 1717@ic-lic03	mgcl	calibrepercadv	huac	n232-134-066	n232-132-197	1	v2022.11	2024-03-08 17:02
4 1717@ic-lic03	mgcl	calibrepercadv	huac	n232-134-066	n232-132-197	1	v2022.11	2024-03-08 17:02

此时可以在新弹出的界面中展示出 license log 中相应 feature/user 的操作记录（默认最多 1000 行）。



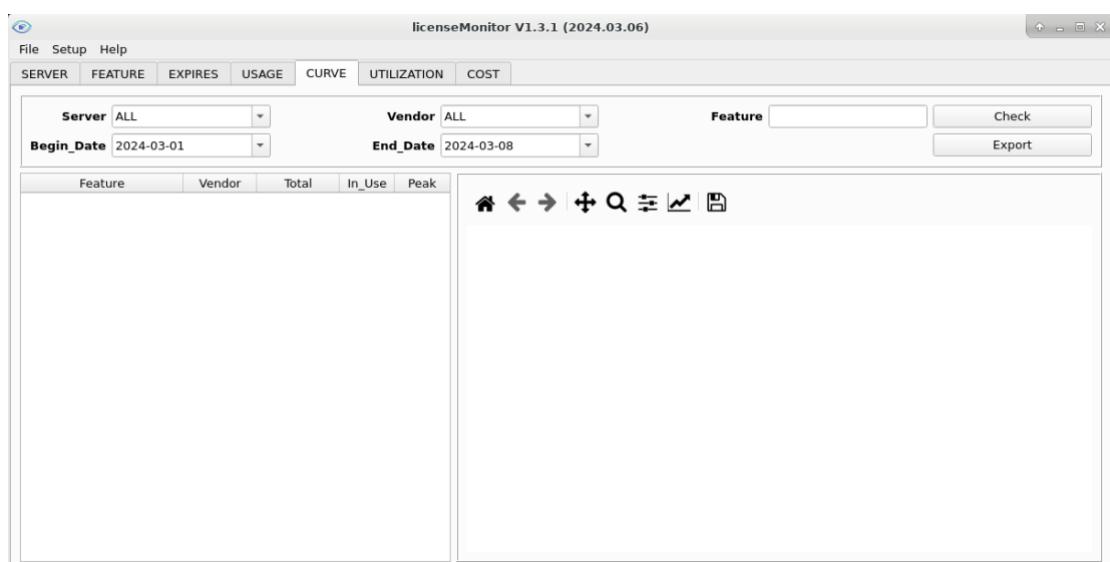
请确保当前用户在当前机器上，可以直接访问指定 license server 的 log file，或者有权限 ssh 到 license server 上访问 license file，否则无法获取有效信息。

另外，如果 license log 尺寸非常大的情况（几千万/上亿行），也非常难在 timeout 之前抓取 license log 中的信息。

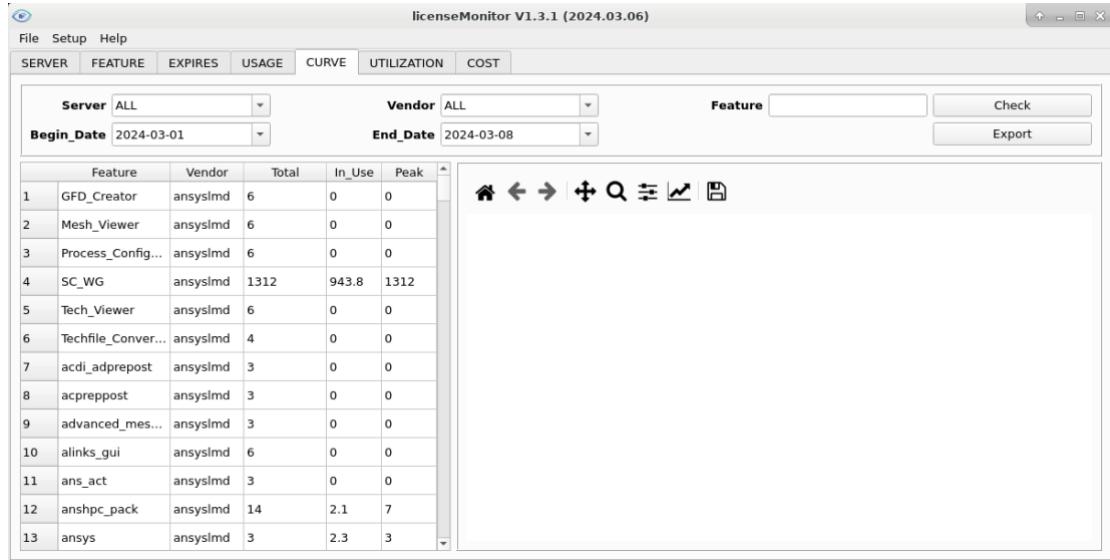
4.2.8 CURVE 页

打开 license_monitor 图形界面后，可以在 CURVE 页查看 license feature 的历史用量曲线信息，仅 licenseMonitor 管理员可查看 CURVE 页。

默认情况不展示任何信息。



选取 Begin_Date 和 End_Date 后（默认为最近一个月），点击“Check”按钮可以获取 EDA license feature 的使用统计信息。



左侧按照 EDA license feature 来展示选定时间段内的使用信息，包括所有 feature 的总量、平均用量、峰值用量，点击 Total/In_Use/Peak 列标题均可以排序。

可以按照“**Server**”来筛选。

可以按照“**Vendor**”来筛选。

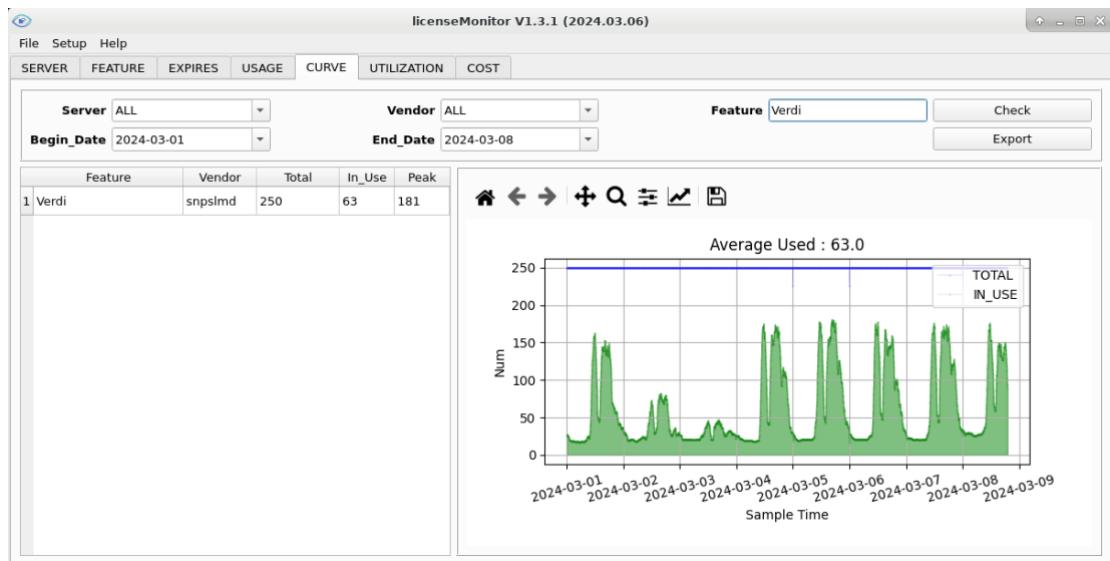
可以按照“**Feature**”来筛选，此处的 Feature 支持模糊检索。比如输入“Verdi”，可以精确地检索到“Verdi”这个 license，如果输入“verdi”，除了“Verdi”外，还可以检索到所有和 verdi 相似的 license。

“**Begin_Date**”用来指定使用分析的起始时间，默认为一个月前。

“**End_Date**”用来指定使用分析的结束时间，默认为今天。

点击左侧 Feature 单元格中的 feature 名字，会自动跳转到 FEATURE 页，展示选中的 license feature 的信息。

指定 Feature 后点击 Check 按钮，可以展示指定 license feature 的统计信息和使用量变化曲线。



点击曲线图上的放大镜图标，可以选取区域放大，点击 Home 按钮可以还原。

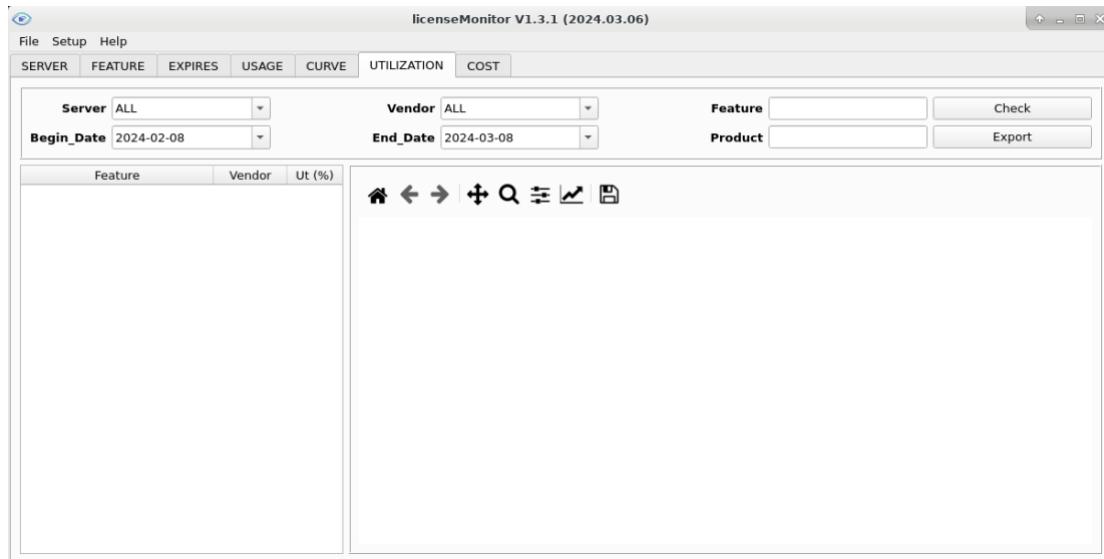


通过放大图可以看到，Verdi 的使用高峰是每天 10-21 点。（周六、周日这个时间段的用量就少得多）

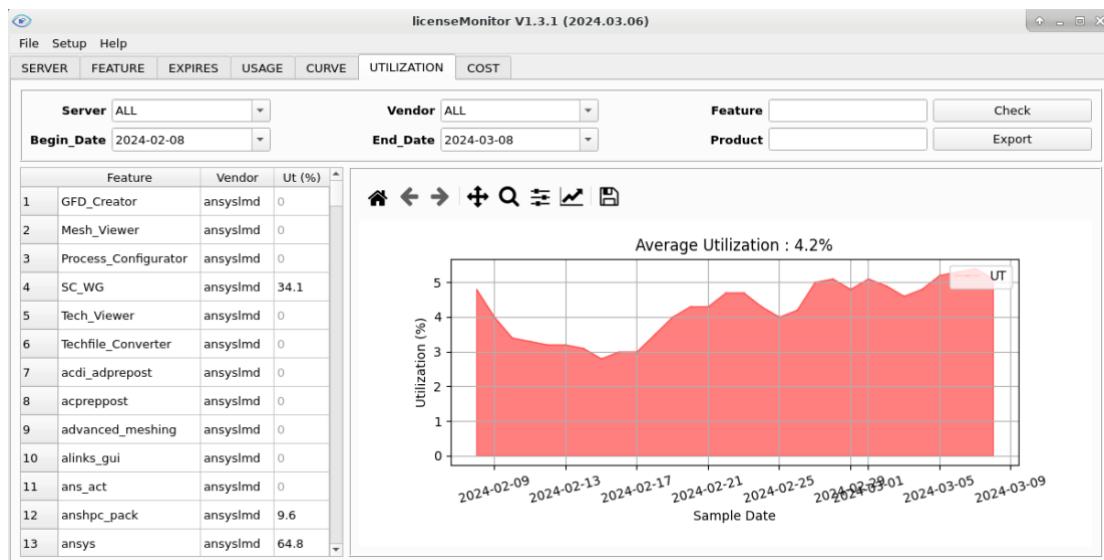
4.2.9 UTILIZATION 页

打开 license_monitor 图形界面后，可以在 UTILIZATION 页查看 license feature 使用率信息，仅 licenseMonitor 管理员可查看 feature 使用率页。

默认情况下不展示任何信息。



选取 Begin_Date 和 End_Date 后（默认为最近一个月），点击“Check”按钮可以获取 EDA license feature 的使用统计信息。



左侧按照 EDA license feature 来展示选定时间段内的使用率信息（均值），点击 Ut 列标题可以排序；右侧则显示所选中 feature 的日均使用率均值变化曲线。

可以按照“Server”来筛选。

可以按照“**Vendor**”来筛选。

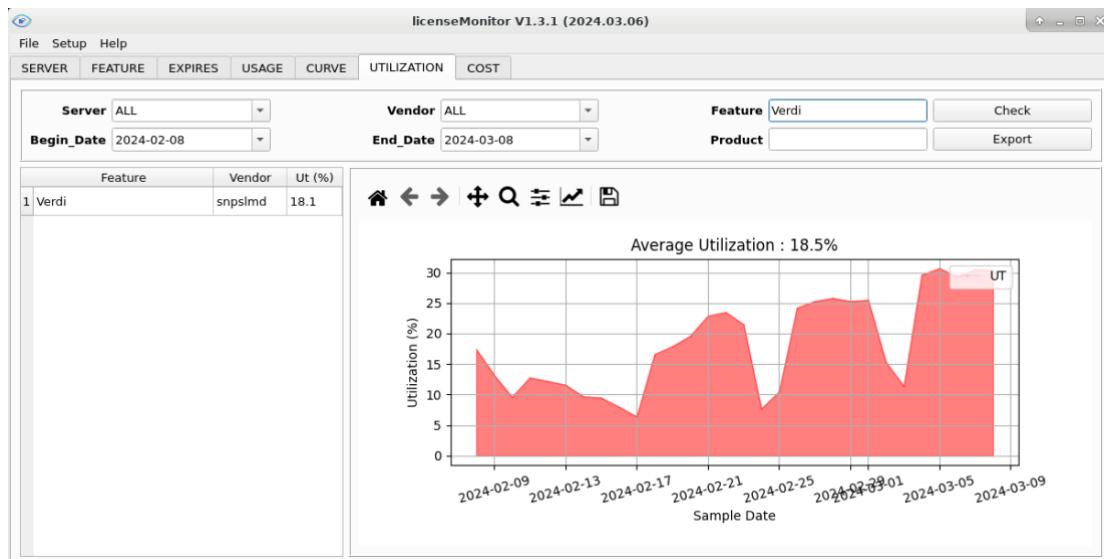
可以按照“**Feature**”来筛选，此处的 Feature 支持模糊检索。比如输入“Verdi”，可以精确地检索到“Verdi”这个 license，如果输入“verdi”，除了“Verdi”外，还可以检索到所有和 verdi 相似的 license。

“**Begin_Date**”用来指定使用率分析的起始时间，默认为一个月前。

“**End_Date**”用来指定使用率分析的结束时间，默认为今天。

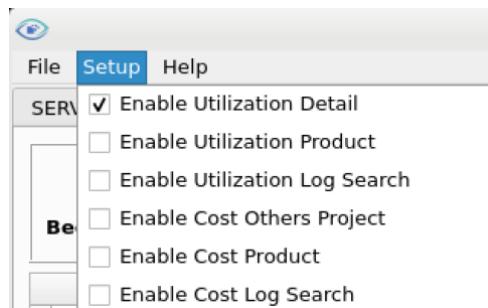
点击左侧 Feature 单元格中的 feature 名字，会自动跳转到 FEATURE 页，展示选中的 license feature 的信息。

可以检索指定 feature 的使用率信息，下面是 Verdi 的使用率过滤信息。

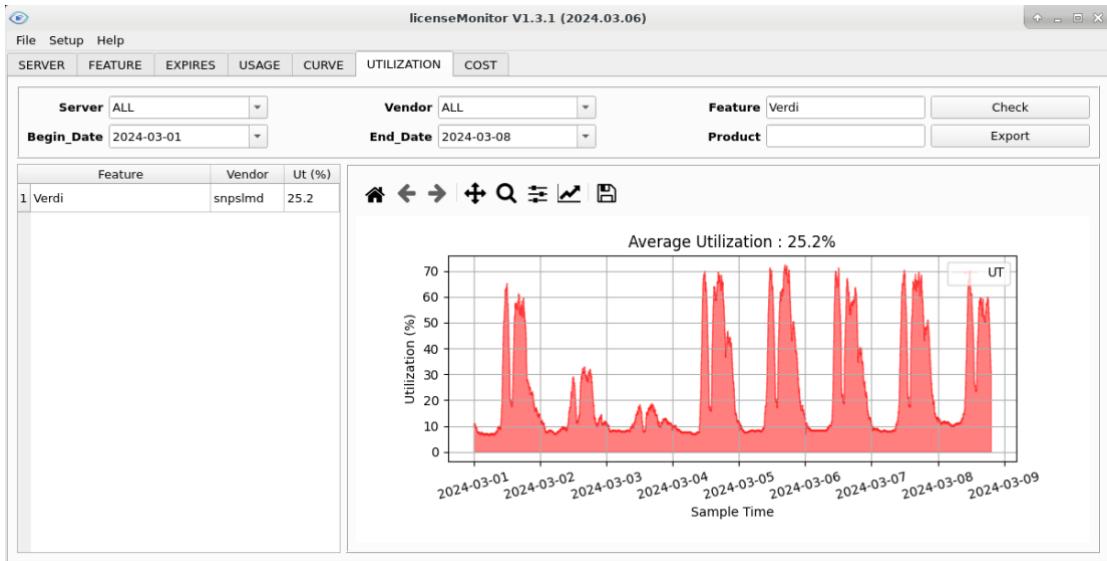


可以注意到，左侧和右侧 Feature 的 utilization 的值有细微差异，这是由于左侧的值是由原始采样信息汇聚得出，右侧的平均的值由日均 utilization 算出，两者在计算过程中由于保留精度有限导致了结果截取误差，理论上左侧的值更准确一些。

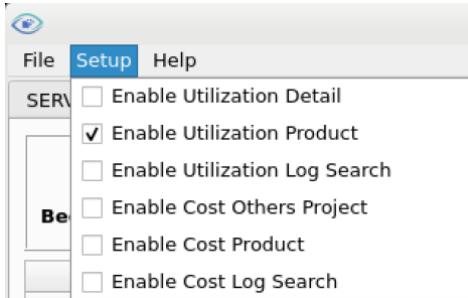
也可以查看使用率的细节信息，只需要在菜单栏“Setup”中选中“Enable Utilization Detail”项，此时，默认检索周期会从一个月变为一周。



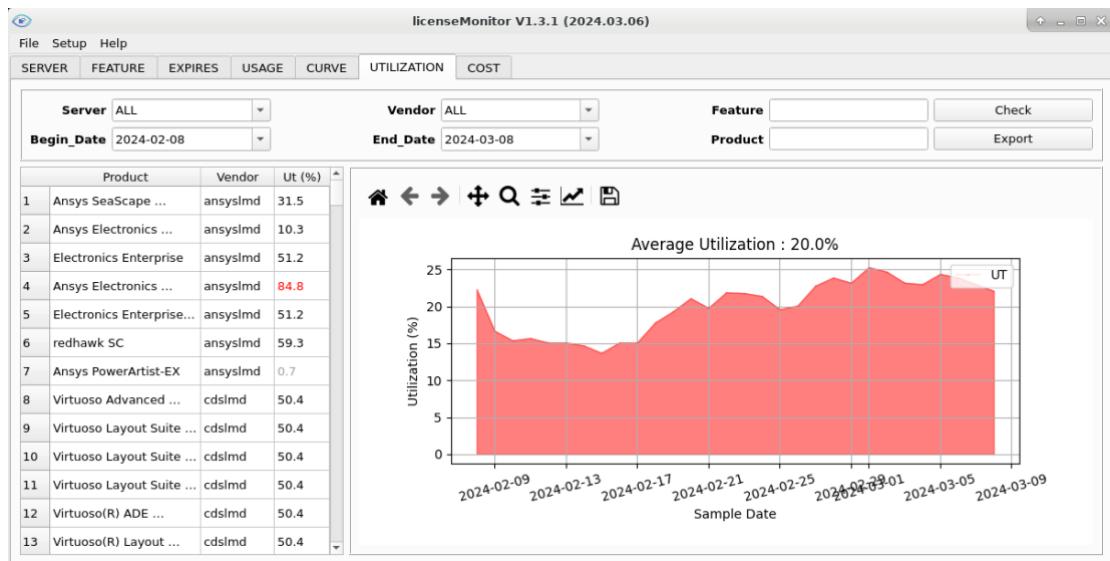
然后点击 Check 按钮，可以得到使用率采样细节。可以参照 4.2.8 的 Verdi 的使用信息，跟这个使用率信息是完全一致的。



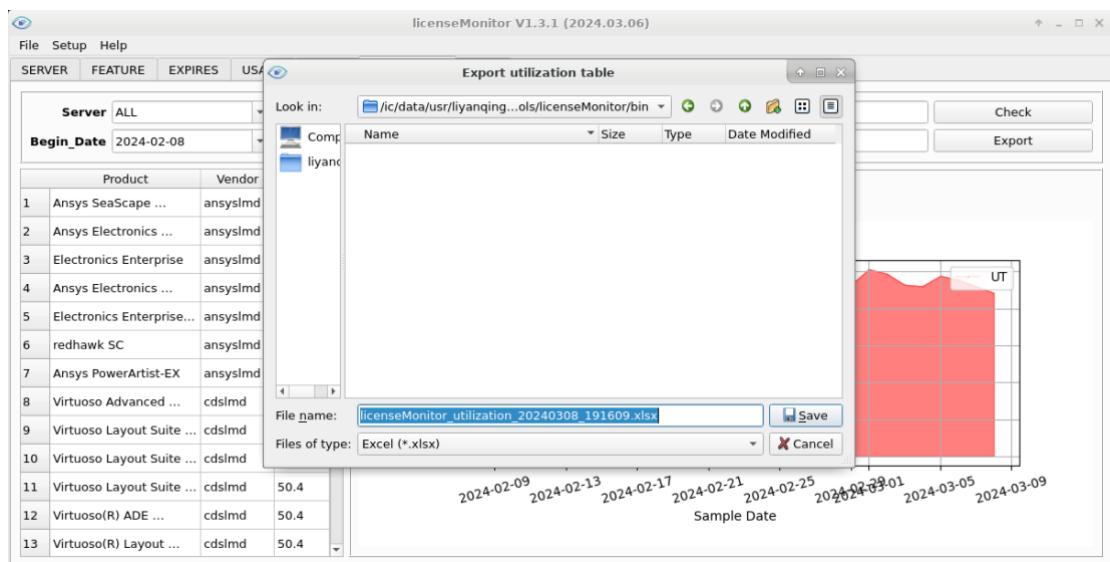
也可以按照 product 维度来统计使用率，只需要在菜单栏“Setup”中选中“Enable Utilization Product”项。



然后点击 Check 按钮，可以得到 product 维度的使用率信息。product 维度的使用率是取 product 内 feature 使用率的最高值，所以整体的平均使用率会更高。



点击“Export”按钮，选择数据存放路径和文件名，可以将左侧的 utilization 数据以 Excel 的方式导出，默认文件名为“licenseMonitor_utilization_<datetime>.xlsx”。

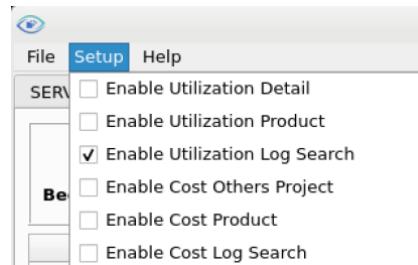


Excel 的内容和 UTILIZATION 页面左侧单元格内容一致。

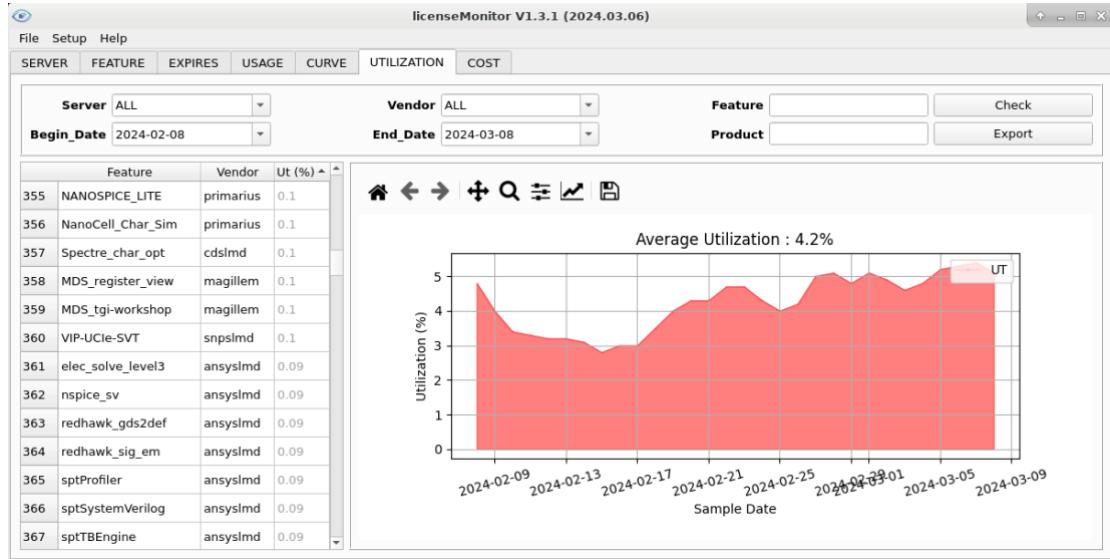
	A	B	C
1	Product	Vendor	Ut (%)
2	Ansys SeaScape Worker Group 256 Pack	ansyslmd	31.5
3	Ansys Electronics Desktop (HPCPack)	ansyslmd	10.3
4	Electronics Enterprise	ansyslmd	51.2
5	Ansys Electronics Desktop (SIWAVE+Q3D+Circuit+HFSS)	ansyslmd	84.8
6	Electronics Enterprise Prep/Post	ansyslmd	51.2
7	redhawk SC	ansyslmd	59.3
8	Ansys PowerArtist-EX	ansyslmd	0.7
9	Virtuoso Advanced Node Framework	cdslmd	50.4
10	Virtuoso Layout Suite EXL	cdslmd	50.4
11	Virtuoso Layout Suite GXL	cdslmd	50.4
12	Virtuoso Layout Suite MXL	cdslmd	50.4
13	Virtuoso(R) ADE Assembler	cdslmd	50.4
14	Virtuoso(R) Layout Suite XL	cdslmd	50.4

菜单栏 File -> Export * table 的功能，均与此一致，分别用于导出 SERVER/FEATURE/EXPIRES/USAGE/CURVE/UTILIZATION/COST 页面的表格内容。

如果 feature/product 没有 utilization 信息，但是不确定是采样没采到还是确实没有使用过，可以使用“Enable Utilization Log Search”功能。



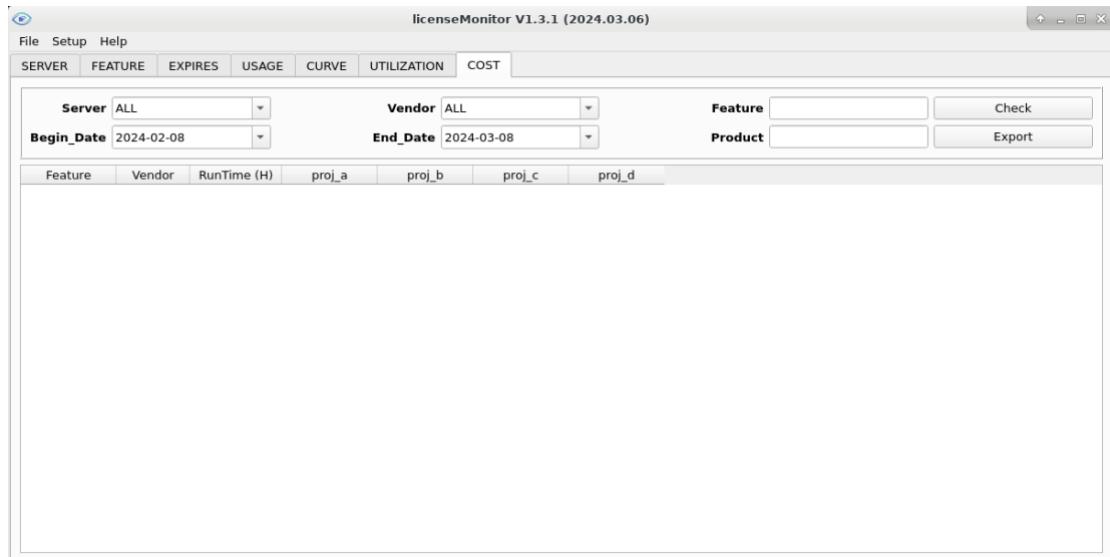
它会尝试在 license log 中检索 license feature 的 checkout 信息，如果检索到，会将 feature/product 的 utilization 置为 0.09%。（不开启这个功能的情况下，最小 Ut 值为 0.1%）



4.2.10 COST 页

打开 license_monitor 图形界面后，可以在 COST 页查看 license 计费分摊信息，仅 licenseMonitor 管理员可以查看 feature 用量费用分摊信息。

默认情况下不展示任何信息。



选取 Begin_Date 和 End_Date 后（默认为最近一个月），点击“Check”按钮可以获取 EDA license feature 的费用分摊信息。

	Feature	Vendor	RunTime (H)	proj_a	proj_b	proj_c	proj_d
1	caldesignrev	mgclld	391	0%	0%	43.84%	56.16%
2	caldfmprope...	mgclld	0	0%	0%	0%	0%
3	calibreadp	mgclld	0	0%	0%	0%	0%
4	calibredrc	mgclld	0	0%	0%	0%	0%
5	calibrehdrc	mgclld	0	0%	0%	0%	0%
6	calibrehlvs	mgclld	0	0%	0%	0%	0%
7	calibrelvls	mgclld	0	0%	0%	0%	0%
8	calibreperc	mgclld	0	0%	0%	0%	0%
9	calibrepercadv	mgclld	11	0%	0%	100%	0%
10	calibrepercsp	mgclld	0	0%	0%	0%	0%
11	calibreqdb	mgclld	74	0%	0%	100%	0%
12	calibrexrc	mgclld	0	0%	0%	0%	0%
13	calibrexrctox...	mgclld	0	0%	0%	0%	0%

可以按照“**Server**”来筛选。

可以按照“**Vendor**”来筛选。

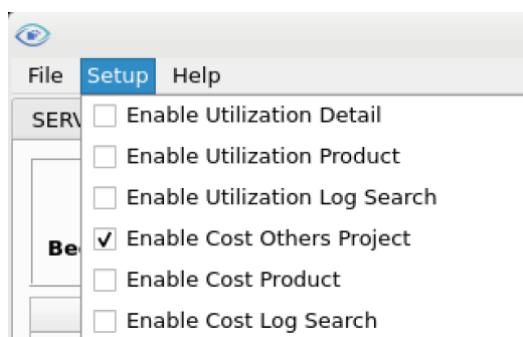
可以按照“**Feature**”来筛选，此处的 Feature 支持模糊检索。比如输入“Verdi”，可以精确地检索到“Verdi”这个 license，如果输入“verdi”，除了“Verdi”外，还可以检索到所有和 verdi 相似的 license。

“**Begin_Date**”用来指定费用分摊的起始时间，默认为一个月前。

“**End_Date**”用来指定费用分摊的结束时间，默认为今天。

点击左侧 Feature 单元格中的 feature 名字，会自动跳转到 FEATURE 页，展示选中的 license feature 的信息。

默认情况下 licenseMonitor 仅统计已知 project 的计费分摊信息，但是在菜单栏“Setup”中选中“Enable cost others Project”项，此时会把未知项目的 license feature 用量也统计进来，并并入“others”项。



然后点击 Check 按钮，可以得到增加了“others”分类的计费分摊信息。

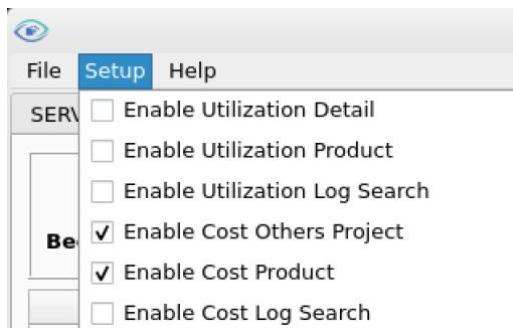
licenseMonitor V1.3.1 (2024.03.06)

File Setup Help

SERVER FEATURE EXPIRES USAGE CURVE UTILIZATION COST

	Server	Vendor	RunTime (H)	proj_a	proj_b	proj_c	proj_d	others
1	caldesignrev	mgcld	391	0%	0%	43.84%	56.16%	0%
2	caldfmpope...	mgcld	0	0%	0%	0%	0%	0%
3	calibreadp	mgcld	0	0%	0%	0%	0%	0%
4	calibredrc	mgcld	0	0%	0%	0%	0%	0%
5	calibrehdrc	mgcld	0	0%	0%	0%	0%	0%
6	calibrehlvs	mgcld	0	0%	0%	0%	0%	0%
7	calibrelvls	mgcld	0	0%	0%	0%	0%	0%
8	calibreperc	mgcld	0	0%	0%	0%	0%	0%
9	calibrepercadv	mgcld	11	0%	0%	100%	0%	0%
10	calibrepercsp	mgcld	0	0%	0%	0%	0%	0%
11	calibreqdb	mgcld	74	0%	0%	100%	0%	0%
12	calibrexrc	mgcld	0	0%	0%	0%	0%	0%
13	calibrexrctox...	mgcld	0	0%	0%	0%	0%	0%

如果想用 product 维度来统计不同项目的费用分摊信息（这是更常见的情况），可以在 Setup 菜单栏中选中“Enable Cost Product”选项。



然后点击 Check 按钮，可以得到 product 维度的费用分摊信息。

licenseMonitor V1.3.1 (2024.03.06)

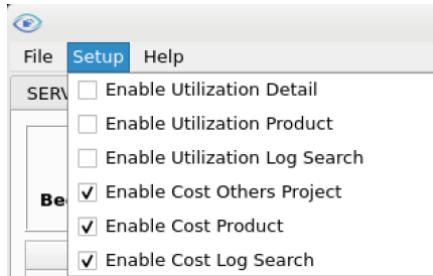
File Setup Help

SERVER FEATURE EXPIRES USAGE CURVE UTILIZATION COST

	Server	Vendor	RunTime (H)	proj_a	proj_b	proj_c	proj_d	others
1	Ansys ...	ansyslmd	0	0%	0%	0%	0%	0%
2	Electronics ...	ansyslmd	0	0%	0%	0%	0%	0%
3	Ansys ...	ansyslmd	24	0%	0%	0%	0%	100%
4	Electronics ...	ansyslmd	0	0%	0%	0%	0%	0%
5	Calibre ...	mgcld	391	0%	0%	43.84%	56.16%	0%
6	Calibre xRC ...	mgcld	0	0%	0%	0%	0%	0%
7	Calibre ...	mgcld	0	0%	0%	0%	0%	0%
8	Calibre ...	mgcld	0	0%	0%	0%	0%	0%
9	Calibre ...	mgcld	0	0%	0%	0%	0%	0%
10	Calibre nmL...	mgcld	0	0%	0%	0%	0%	0%
11	Calibre PERC...	mgcld	0	0%	0%	0%	0%	0%
12	Calibre PERC...	mgcld	0	0%	0%	0%	0%	0%
13	Calibre PERC...	mgcld	11	0%	0%	100%	0%	0%

同 UTILIZATION 页，费用分摊信息还可以通过“Export”按钮导出成 Excel 文档，用法不再赘述。

另外，如果担心某些 license feature 的使用信息不能正确采样而错失，可以在 Setup 菜单栏中启用“Enable Cost Log Search”选项，此时它会为未抓到使用信息的 license feature，去 license log 中检索 checkout 信息，作为计费分摊时候的补充。



此时，所有 RunTime 小于 1 的 feature/product，都属于这种情况，可以根据 RunTime 的值来辨别。(不开启这个功能的情况，最小 RunTime 值为 1)

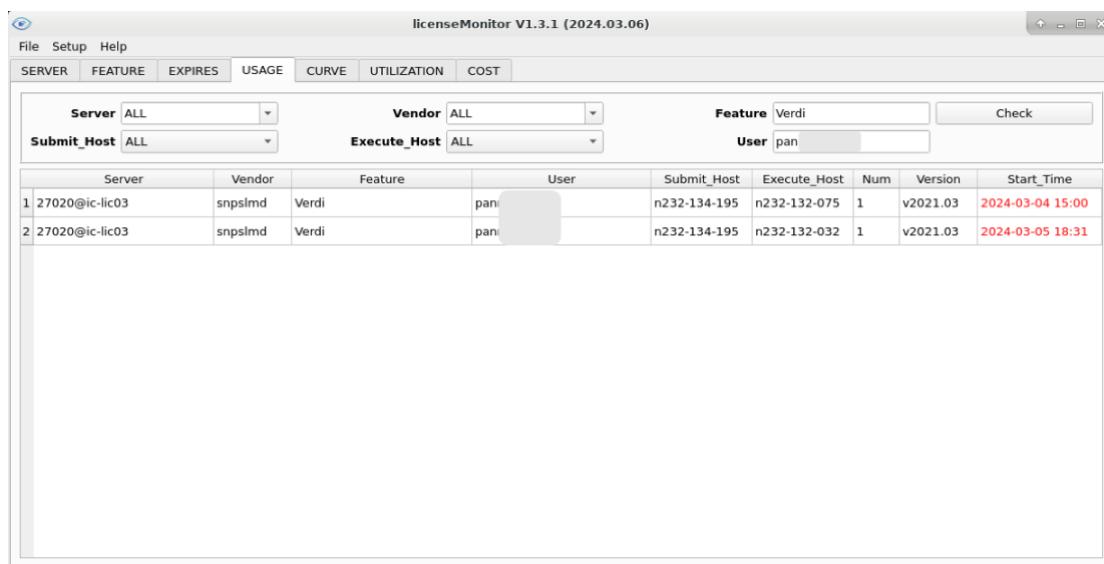
licenseMonitor V1.3.1 (2024.03.06)							
Server		FEATURE		EXPIRES		USAGE	
Product	Vendor	RunTime (H)	proj_a	proj_b	proj_c	proj_d	others
46 VC VIP PCIe ...	snpsslmd	62	0%	0%	100%	0%	0%
47 VC VIP MIPI ...	snpsslmd	28	0%	0%	100%	0%	0%
48 Ansys ...	ansyslmd	25	0%	0%	0%	0%	100%
49 Calibre PERC...	mgclid	11	0%	0%	100%	0%	0%
50 Electronics ...	ansyslmd	0.98	0%	0%	0%	0%	100%
51 HAPS-100 ...	snpsslmd	0.66	7.58%	74.24%	15.15%	3.03%	0%
52 ZSS Module	snpsslmd	0.66	7.58%	74.24%	15.15%	3.03%	0%
53 VCS Apex	snpsslmd	0.41	50494.07%	18027.41%	1225775%	912.4%	0%
54 Calibre xFRC ...	mgclid	0.4	0%	0%	13.75%	86.25%	0%
55 Calibre ...	mgclid	0.4	5.83%	5.83%	8.35%	80%	0%
56 Calibre ...	mgclid	0.4	5.83%	5.83%	7.1%	81.25%	0%
57 Calibre ...	mgclid	0.4	0%	0%	28.75%	71.25%	0%
58 Calibre nmL...	mgclid	0.4	0.83%	0.83%	30.84%	67.5%	0%

4.2.11 其它功能

如果希望在 license_monitor 启动的时候直接查看***对 Verdi 相关的 feature 的使用情况，并直接切换到 USAGE 页面，可以通过如下命令行指定。

```
[liyanqing.1987@n232-135-067 licenseMonitor]$ bin/license_monitor
-f Verdi -u pan*** -t USAGE
* [2023-12-20 14:34:43] Loading license info, please wait a
moment ...
*Warning*: No feature is specified, will not generate curve.
```

```
* [2023-12-20 14:34:57] Process utilization info, please wait a moment ...
* [2023-12-20 14:34:57] Loading license info, please wait a moment ...
* [2023-12-20 14:35:09] Loading utilization info, please wait a moment ...
* [2023-12-20 14:35:09] Process utilization info, please wait a moment ...
* [2023-12-20 14:35:09] Loading cost info, please wait a moment ...
```



五、辅助工具

出于提升易用性和扩展功能的目的，licenseMonitor 自带了一些辅助工具。

5.1 gen_LM_LICENSE_FILE

用途：生成 config/LM_LICENSE_FILE 配置文件。

使用场景：用户使用 module files 管理 EDA 环境变量设置，module files 配置文件中的 LM_LICENSE_FILE 环境变量已经包含了环境中所有的 license server 设置。

gen_LM_LICENSE_FILE 的帮助信息如下。

```
[liyanqing.1987@n232-135-067
licenseMonitor]$ tools/gen_LM_LICENSE_FILE -h
usage: gen_LM_LICENSE_FILE.py [-h] -m MODULE_FILES_DIRS
[MODULE_FILES_DIRS ...] [-f LM_LICENSE_FILE_FILE]

optional arguments:
  -h, --help            show this help message and exit
  -m MODULE_FILES_DIRS [MODULE_FILES_DIRS ...], --
module_files_dirs MODULE_FILES_DIRS [MODULE_FILES_DIRS ...]
                                Required argument, specify the directories
where save module configuration files.
  -f LM_LICENSE_FILE_FILE, --LM_LICENSE_FILE_file
LM_LICENSE_FILE_FILE
                                Specify output file, default is
"/ic/data/usr/liyanqing.1987/tools/licenseMonitor/LM_LICENSE_FILE"
  .
```

--module_files_dirs: 必选项，指定 module files 的目录。

--LM_LICENSE_FILE_file: 指定输出 LM_LICENSE_FILE 文件的路径， 默认生成在当前路径下。

下面是一个示例。

```
[liyanqing.1987@n232-135-067
licenseMonitor]$ tools/gen_LM_LICENSE_FILE -m
/ic/software/modules/config/EDA -f ./config/LM_LICENSE_FILE
```

```
>>> Parse "/ic/software/modules/config/EDA/ansys/helic/2023.R2.3"
    Find 1055@ic-lic03
...
>>> Parse "/ic/software/modules/config/EDA/synopsys/zebu/S-
2021.09-1"
    Find 27020@ic-lic03

>>> Checking license server status ...
*Warning*: vendor daemon status is "DOWN" for "1717@ic-mentor-
lic03/saltd".
*Warning*: vendor daemon status is "DOWN" for "1717@ic-mentor-
lic02/saltd".

>>> Write "./LM_LICENSE_FILE"
    1055@ic-lic03
    1717@ic-lic03
    27020@ic-lic03

* LM_LICENSE_FILE is saved on "./config/LM_LICENSE_FILE".
[liyanqing.1987@n232-135-067 licenseMonitor]$
[liyanqing.1987@n232-135-067 licenseMonitor]$ cat
config/LM_LICENSE_FILE
1055@ic-lic03
1717@ic-lic03
27020@ic-lic03
```

注意，获取到的 license server 需要逐个验证其服务状态，如果 license server 比较多，这个过程会比较慢（几分钟），请耐心等待。

5.2 get_product_feature_relationship

用途：获取 EDA license 的 product 和 feature 的映射关系，生成 config/others/product_feature.yaml 配置文件。

使用场景：仅支持 cdslmd/snpslmd/mgclm 三个 vendor daemon，需要能够访问对应的 license file(s)。

get_product_feature_relationship 帮助信息如下。

```
[liyanqing.1987@n232-135-067
licenseMonitor]$ tools/get_product_feature_relationship -h
usage: get_product_feature_relationship.py [-h] -v
{cdslmd,snpsslmd,mgcld} [{cdslmd,snpsslmd,mgcld} ...] -l
LICENSE_FILES [LICENSE_FILES ...] [-f PRODUCT_FORMAT] [-o
OUTPUT_FILE]

optional arguments:
-h, --help            show this help message and exit
-v {cdslmd,snpsslmd,mgcld} [{cdslmd,snpsslmd,mgcld} ...], --
vendors {cdslmd,snpsslmd,mgcld} [{cdslmd,snpsslmd,mgcld} ...]
                                         Required argument, specify vendor daemon
list, must be the same order of license_files.
-l LICENSE_FILES [LICENSE_FILES ...], --license_files
LICENSE_FILES [LICENSE_FILES ...]
                                         Required argument, specify license files,
must be the same order of vendors.
-f PRODUCT_FORMAT, --product_format PRODUCT_FORMAT
                                         Specify product format, default is
"<product_name>", only support "<product_id>" and "<product_name>""
two variables.
-o OUTPUT_FILE, --output_file OUTPUT_FILE
                                         Output file, default is
"./product_feature.yaml".
```

--vendors: 必选项，指定 vendor daemon 的名字，请注意，此处建议跟 lmstat 获取到的 vendor daemon 信息一致（比如 cadence 的 vendor daemon 名其实是“cdslmd”）。

--license_files: 必选项，指定 vendor daemon 对应的 license file。

--product_format: 指定 product 的样式，默认为“<product_name>”，也可以写成“<product_name>:<product_id>”等样式，当前仅支持“<product_name>”和“<product_id>”两个变量。

--output_file: 指定输出文件路径，默认为跟 license_file 同名的 yaml 文件。

下面是一个示例。

```
[liyanqing.1987@n232-135-067
licenseMonitor]$ tools/get_product_feature_relationship -v cdslmd
-l
/ic/data/usr/liyanqing.1987/license/license_file/cadence_license_f
```

```
ile.txt
>>> Parse cdslmd license file
"/ic/data/usr/liyanqing.1987/license/license_file/cadence_license_
file.txt"
>>> Verify feature product_id/product_name ...
*Warning*: No product_id/product_name information for feature
"LicFileVersion".
*Warning*: No product_id/product_name information for feature
"Use_Server_Options".

>>> Write output file
"/ic/data/usr/liyanqing.1987/tools/licenseMonitor/product_feature.
yaml".
[liyanqing.1987@n232-135-067 licenseMonitor]$
[liyanqing.1987@n232-135-067 licenseMonitor]$ cat
product_feature.yaml
cdslmd:
  '111':
    - Virtuoso Advanced Node Framework
    - Virtuoso Layout Suite EXL
...
...
```

请确保 license file 是原始信息，没有被篡改，以防止出现解析错误。

另外，部分 vendor 临时 license 是没有 product 信息的，这种也无法获取映射关系。

5.3 config_product_feature_relationship

用途：手工配置 config/others/product_feature.yaml 配置文件，补齐 get_product_feature_relationship 脚本所不能自动抓取的 product-feature 映射关系。

使用场景：需要能够访问对应的 license file(s)。

config_product_feature_relationship 为图形界面程序，打开后样式如下。

```
[liyanqing.1987@n232-135-067
licenseMonitor]$ tools/config_product_feature_relationship
```



Vendor Daemon: 必填项，填写 license server 的 vendor daemon 信息（比如 cadence 的 vendor daemon 名其实是“cdslmd”）。

License File: 必填项，填写 vendor daemon 对应的 license file 的路径，用于解析获取 feature list。

Yaml File: 选填项，填写 vendor daemon 的 product-feature 映射关系文件，用于为从 license file 中解析到的 feature 自动填写已知的 product 映射。

Output: 选填项，默认为当前路径下的 product_feature.yaml 文件。

下面是一个示例。

Config product-feature relationship		
Feature		Product
1	111	product 1#product 2
2	940	product 3
3	945	product 4
4	ACCEL_MEM_PLUS	
5	ACCEL_MEM_VIP	
6	ACC_VDEBUG	
7	ACC_VIP_AMBA	
8	ACC_VIP_AMBA_APB	
9	ACC_VIP_AXI4_ACE	
10	ACC_VIP_AXI_3_4	

Output : /ic/data/usr/liyanqing.1987/tools/licenseMonitor/tools/cdslmd.yaml Gen

加载了 cadence 的 license file，并配置 Vendor Daemon 为“cdslmd”，Output 的文件名改为“cdslmd.yaml”。点击“Update”按钮，可以展示出 license file 中的 feature。

为 111/940/945 等 feature 配置了 product，一个 feature 可以对应多个 product，不同 product 之间用“#”隔开。点击“Gen”按钮，会生成 yaml 文件。

```
[liyanqing.1987@n232-135-067
licenseMonitor]$ tools/config_product_feature_relationship
>>> Parse license file
"/ic/data/usr/liyanqing.1987/license/license_file/cadence_license_
file.txt".
*Warning*: "ACCEL_MEM_PLUS": Not find related product setting.
*Warning*: "ACCEL_MEM_VIP": Not find related product setting.
...
>>> Write output file
"/ic/data/usr/liyanqing.1987/tools/licenseMonitor/tools/cdslmd.yaml".
```

生成的 cdslmd.yaml 内容如下。

```
[liyanqing.1987@n232-135-067 licenseMonitor]$ cat cdslmd.yaml
cdslmd:
  '111':
    - product 1
    - product 2
  '940':
    - product 3
  '945':
    - product 4
...
```

5.4 update_product_feature_relationship

用途：通过解析 config/LM_LICENSE_FILE 文件获取所有的 license server 及 license file，对 cdslmd/snpslmd/mgclm 三个 vendor daemon 的 license file 解析其 product-feature 映射关系，并自动更新 config/others/product_feature.yaml 配置文件。

使用场景：需要能够访问对应的 license file(s)。

update_product_feature_relationship 的帮助信息如下。

```
[liyanqing.1987@n232-135-067
licenseMonitor]$ tools/update_product_feature_relationship -h
usage: update_product_feature_relationship.py [-h] -l
LM_LICENSE_FILE_FILE [-p PRODUCT_FEATURE_RELATIONSHIP_FILE] [-f
PRODUCT_FORMAT] [-i] [-o OUTPUT_FILE]

optional arguments:
  -h, --help            show this help message and exit
  -l LM_LICENSE_FILE_FILE, --LM_LICENSE_FILE_file
LM_LICENSE_FILE_FILE
                                Required argument, specify licenseMonitor
LM_LICENSE_FILE file.
  -p PRODUCT_FEATURE_RELATIONSHIP_FILE, --
product_feature_relationship_file
PRODUCT_FEATURE_RELATIONSHIP_FILE
                                Specify original product feature
relationship file.
  -f PRODUCT_FORMAT, --product_format PRODUCT_FORMAT
                                Specify product format, default is
"<product_name>", only support "<product_id>" and "<product_name>" two variables.
  -i, --incremental_mode
                                Enable incremental mode, default will
replace old product feature relationship settings.
  -o OUTPUT_FILE, --output_file OUTPUT_FILE
                                Output file, default is
"./product_feature.<CURRENT_TIME>.yaml".
```

--LM_LICENSE_FILE_FILE: 必选项，指定 licenseMonitor 的 LM_LICENSE_FILE 位置。

--PRODUCT_FEATURE_RELATIONSHIP_FILE: 指定旧的 product_feature.yaml 文件，如指定，则基于此修改，如未指定，则全新创建。

--PRODUCT_FORMAT: 指定 product 的样式，默认为“<product_name>”，也可以写成“<product_name>”和“<product_id>”相结合的样式。

--incremental_mode: 开启增量模式，不改变旧的 product_feature.yaml 中的信息，只增加新信息。默认情况下会替换掉旧的 product_feature.yaml 中的同类信息。

--output_file: 指定输出文件。

下面是一个示例。

```
[liyanqing.1987@n232-135-067
licenseMonitor]$ tools/update_product_feature_relationship -l
config/LM_LICENSE_FILE -p config/others/product_feature.yaml -i
>>> Setup Environment variable LM_LICENSE_FILE with
"config/LM_LICENSE_FILE"
>>> Get vendor daemon license files information
>>> Parse mgcld license file "/ic/data/CAD/lic/mentor/ic-
lic03/ActiveLicenses/mgcld.lic"
>>> Verify feature product_id/product_name ...
>>> Parse snpslmd license file
"/ic/data/CAD/lic/synopsys/Synopsys_Key_Site_48730_snpslmd_0225.tx
t"
*Warning*: Not find product_id "0" for feature "SSS".
>>> Verify feature product_id/product_name ...
*Warning*: No product_id/product_name information for feature
"SSS".
...
*Warning*: No product_id/product_name information for feature
"Formality-LP".

>>> Write output file "./product_feature.20231220_163242.yaml".
```

如果确认新生成的 product_feature 映射文件没问题，可以用来取代旧的映射文件。

5.5 view_product_feature_relationship

用途：图形化查看 product_feature.yaml 中的 product 和 feature 映射关系。

使用场景：存在一个可用非空的 product_feature.yaml 文件。

view_product_feature_relationship 帮助信息如下。

```
[liyanqing.1987@n232-135-067
licenseMonitor]$ tools/view_product_feature_relationship -h
usage: view_product_feature_relationship.py [-h] [-i INPUT_FILE]

optional arguments:
-h, --help            show this help message and exit
-i INPUT_FILE, --input_file INPUT_FILE
```

Specify input product_feature relationship
yaml file.

--input_file: 指定输入文件路径, 即 product_feature.yaml 文件, 默认为 licenseMonitor 安装路径下的 config/others/product_feature.yaml 文件。

下面是一个示例。

```
[liyanqing.1987@n232-135-067
licenseMonitor]$ tools/view_product_feature_relationship
QStandardPaths: XDG_RUNTIME_DIR not set, defaulting to
'/tmp/runtime-liyanqing.1987'
```

View product-feature relationship			
File		Check	
Vendor	Product	Feature	
1	ansyslmd	Ansys SeaScape Worker Group 256 Pack	SC_WG
2	ansyslmd	Electronics Enterprise Prep/Post	a_spaceclaim_dirmod
3	ansyslmd	Electronics Enterprise Prep/Post	electronics2d_gui
4	ansyslmd	Electronics Enterprise Prep/Post	electronics3d_gui
5	ansyslmd	Electronics Enterprise Prep/Post	electronicsckt_gui
6	ansyslmd	Ansys Electronics Desktop (HPCPack)	anshpc_pack
7	ansyslmd	Ansys Discovery Modeling	disco_level1
8	ansyslmd	Electronics Enterprise	elec_solve_2d
9	ansyslmd	Electronics Enterprise	elec_solve_icepak
10	ansyslmd	Electronics Enterprise	elec_solve_level1
11	ansyslmd	Electronics Enterprise	elec_solve_level2
12	ansyslmd	Electronics Enterprise	elec_solve_level3
13	ansyslmd	Electronics Enterprise	elec_solve_maxwell

可以按照 product 或者 feature 来检索，支持模糊匹配。

View product-feature relationship			
File	/ic/data/usr/liyanqing.1987/tools/licenseMonitor/config/others/product_feature.yaml	Product	Feature
Vendor	ALL	Product	Feature
1	snpsslmd	VCS Elite	VCS-2-Elite-Base-Compile-Pkg
2	snpsslmd	VCS Elite	VCS-2-Elite-Base-Runtime-Pkg
3	snpsslmd	VCS Elite	VCS-2-Elite-Compile-Pkg
4	snpsslmd	VCS Elite	VCS-2-Elite-Runtime-Pkg

5.6 update_project_execute_host_with_lsf

用途：解析 LSF 的 queue-host 映射关系，据此更新 config/project/project_execute_host 配置文件。

使用场景：使用 LSF 来管理所有 EDA 仿真节点，且 LSF queue 的命名符合“<project>”或者“<project>_*”或者“<project>-*”的样式。

update_project_execute_host_with_lsf 的帮助信息如下。

```
[liyanqing.1987@n232-135-067
licenseMonitor]$ tools/update_project_execute_host_with_lsf -h
usage: update_project_execute_host_with_lsf.py [-h] [-p
PROJECT_EXECUTE_HOST_FILE] [-o OUTPUT_FILE]

optional arguments:
  -h, --help            show this help message and exit
  -p PROJECT_EXECUTE_HOST_FILE, --project_execute_host_file
PROJECT_EXECUTE_HOST_FILE
                                Specify original project execute host
file.
  -o OUTPUT_FILE, --output_file OUTPUT_FILE
                                Output file, default is
"./project_execute_host.<CURRENT_TIME>".
```

--project_execute_host_file: 指定原始的 project_execute_host_file, 如未指定, 则默认重新生成, 如指定, 则增量修改。

--output_file: 指定新生成的 project_execute_host_file 路径, 默认为“./project_execute_host.<CURRENT_TIME>”。

下面是一个示例。

```
[liyanqing.1987@n232-135-067
licenseMonitor]$ tools/update_project_execute_host_with_lsf -p
config/project/project_execute_host
>>> Get project list from
"/ic/data/usr/liyanqing.1987/tools/licenseMonitor/config/project/p
roject_list"
>>> Get project information for LSF queue(s)
    * Get project information for LSF queue "comet"
...
>>> Get execute_host - project relationship from queue-host
information

>>> Write output file "./project_execute_host.20231220_170815".
```

如果确认新生成的 project - execute_host 的映射关系没有问题, 可以用来取代旧的映射文件。

5.7 get_license_log

用途: 获取 license server 和 license log path 的映射关系, 生成 config/others/product_feature.yaml 配置文件。

使用场景: 存在一个可用非空的 LM_LICENSE_FILE 文件, 且当前账号在当前机器上可以 ssh 登陆所有的 license server。

get_license_log 帮助信息如下。

```
[liyanqing.1987@n232-135-067
licenseMonitor]$ tools/get_license_log -h
usage: get_license_log.py [-h] -l LM_LICENSE_FILE [ -o
OUTPUT_FILE]
```

```
optional arguments:
  -h, --help            show this help message and exit
  -l LM_LICENSE_FILEFILE, --LM_LICENSE_FILE_file
LM_LICENSE_FILEFILE
                                         Required argument, specify licenseMonitor
LM_LICENSE_FILE file.
  -o OUTPUT_FILE, --output_file OUTPUT_FILE
                                         Output file, default is
"<CWD>/license_log.<CURRENT_TIME>.yaml".
```

--LM_LICENSE_FILE_file: 必选项，指定 LM_LICENSE_FILE，一般使用 licenseMonitor 安装目录下的 config/LM_LICENSE_FILE 即可。

--output_file: 输出文件路径，默认为 "<CWD>/license_log.<CURRENT_TIME>.yaml"。

下面是一个示例。

```
[root@ic-admin2 licenseMonitor]# tools/get_license_log -l
config/LM_LICENSE_FILE
>>> Processing license server "ic-lic03" ...
    Port(1055) find license log "/ic/data/CAD/lic/logs/ansys_ic-
lic03.log".
    Port(27020) find license log
"/ic/data/CAD/lic/logs/synopsys_ic-lic03.log".
    Port(1717) find license log "/ic/data/CAD/lic/mentor/ic-
lic03/saltd.log".
>>> Save output file
"/ic/data/usr/liyanqing.1987/tools/licenseMonitor/license_log.2024
0310_151334.yaml".
```

此时我们得到的 output_file 文件内容如下。

```
[root@ic-admin2 licenseMonitor]# cat
license_log.20240310_151334.yaml
1055@ic-lic03: /ic/data/CAD/lic/logs/ansys_ic-lic03.log
1717@ic-lic03: /ic/data/CAD/lic/mentor/ic-lic03/saltd.log
27020@ic-lic03: /ic/data/CAD/lic/logs/synopsys_ic-lic03.log
```

5.8 collect_feature_record_frome_license_log

用途：从 license log 中搜集指定（默认所有）license feature 的历史记录， 默认只记录最后 10 条。

使用场景：存在一个可用非空的 LM_LICENSE_FILE 文件，且当前账号在当前机器上可以 ssh 登陆所有的 license server，且可以读取对应的 license log 文件。

collect_feature_record_frome_license_log 帮助信息如下。

```
[liyanqing.1987@n232-135-067
licenseMonitor]$ tools/collect_feature_record_frome_license_log -h
usage: collect_feature_record_frome_license_log.py [-h] -L
LM_LICENSE_FILE [-l LICENSE_LOG_CONFIG_FILE] [-o OUTPUT_FILE]

optional arguments:
  -h, --help            show this help message and exit
  -L LM_LICENSE_FILE, --LM_LICENSE_FILE_file
LM_LICENSE_FILE_FILE
                                Required argument, specify licenseMonitor
LM_LICENSE_FILE file.
  -l LICENSE_LOG_CONFIG_FILE, --license_log_config_file
LICENSE_LOG_CONFIG_FILE
                                Specify license log configuration file,
must be yaml format.
  -o OUTPUT_FILE, --output_file OUTPUT_FILE
                                Output file, default is
"<CWD>/feature_record_on_license_log.<CURRENT_TIME>.yaml".
```

--LM_LICENSE_FILE_file: 必选项，指定 LM_LICENSE_FILE，一般使用 licenseMonitor 安装目录下的 config/LM_LICENSE_FILE 即可。

--license_log_config_file: 指定 license_server 和 license_log 对应关系的配置文件，如未指定，会默认去寻找 licenseMonitor 安装目录下的 config/others/license_log.yaml。

--output_file: 输出文件路径，默认为 "<CWD>/feature_record_on_license_log.<CURRENT_TIME>.yaml"。

下面是一个示例。

```
[root@ic-admin2 licenseMonitor]#
tools/collect_feature_record_frome_license_log -L
```

```
config/LM_LICENSE_FILE -l config/others/license_log.yaml -o
feature_record_on_license_log.yaml
>>> Getting license log list ...
>>> Setup env variable LM_LICENSE_FILE with
"config/LM_LICENSE_FILE"
>>> Getting license feature list ...
>>> Collecting license feature record info ...
    Searching record for license_server(1055@ic-lic03)
vendor_daemon(ansyslmd) feature(alinks_gui)
    Searching record for license_server(1055@ic-lic03)
vendor_daemon(ansyslmd) feature(elec_solve_2d)
    ...
>>> Saving output file "feature_record_on_license_log.yaml".
```

5.9 show_license_log_info

用途：图形化界面的形式查看指定 license_log 中的指定内容，主要在 bin/license_monitor 中调用，也可以单独执行。

使用场景：当前账号在当前机器上可以访问对应 license server 的 log 文件。

show_license_log_info 帮助信息如下。

```
[liyanqing.1987@n232-135-067
licenseMonitor]$ tools/show_license_log_info -h
usage: show_license_log_info.py [-h] -s SERVER -v VENDOR -f
FEATURE [-u USER] -l LIC_FILES [LIC_FILES ...]

optional arguments:
-h, --help            show this help message and exit
-s SERVER, --server SERVER
                      Required argument, specify license server.
-v VENDOR, --vendor VENDOR
                      Required argument, specify vendor daemon.
-f FEATURE, --feature FEATURE
                      Required argument, specify license
feature.
-u USER, --user USER  Specify user.
-l LIC_FILES [LIC_FILES ...], --lic_files LIC_FILES
[LIC_FILES ...]
                      Required arguments, specify license files.
```

- server: 必选项，指定 license server。
- server: 必选项，指定 vendor daemon。
- feature: 必选项，指定 license feature。
- user: 可选项，指定 license server 的 user。
- lic_files: 必选项，指定 license files。

下面是一个示例。

```
[liyanqing.1987@n232-135-067
licenseMonitor]$ tools/show_license_log_info -s 1717@ic-lic03 -v
mgcld -f calibrepercadv -l /ic/data/CAD/lic/mentor/ic-
lic03/ActiveLicenses/mgcld.lic
[2024-03-10 15:50:46] Searching feature info from log, please wait
a moment ...
```

Licence Log Info

Server 1717@ic-lic03		Vendor mgcld		Check	
Feature	calibrepercadv	User		Status	ALL
Log Time	Status	Feature	User	Execute_Host	Info
1 0:28:33	IN	calibrepercadv		n232-132-022	
2 0:28:33	IN	calibrepercadv		n232-132-022	
3 0:28:33	IN	calibrepercadv		n232-132-022	
4 0:28:33	IN	calibrepercadv		n232-132-022	
5 0:26:29	DENIED	calibrepercadv		n232-132-022	(Licensed number ...)
6 0:26:29	DENIED	calibrepercadv		n232-132-022	(Licensed number ...)
7 0:25:09	DENIED	calibrepercadv		n232-132-022	(Licensed number ...)
8 23:19:42	DENIED	calibrepercadv		n232-132-022	(Licensed number ...)
9 14:03:23	DENIED	calibrepercadv		n232-132-022	(Licensed number ...)
10 14:03:15	DENIED	calibrepercadv		n232-132-022	(Licensed number ...)
11 14:03:15	DENIED	calibrepercadv		n232-132-022	(Licensed number ...)
12 14:03:15	DENIED	calibrepercadv		n232-132-022	(Licensed number ...)
13 14:03:08	DENIED	calibrepercadv		n232-132-022	(Licensed number ...)

请注意，此处 feature 和 user 均不支持模糊匹配。

5.10 seedb

用途：查看 sqlite3 数据库内容。

使用场景：存在可用非空的 sqlite3 database 文件。

seedb 帮助信息如下。

```
[liyanqing.1987@n232-135-067 licenseMonitor]$ 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：查看指定 database 下所有的 tables。

```
[liyanqing.1987@n232-135-067 licenseMonitor]$ seedb -
d ../../database/licenseMonitor/db/license_server/1717@ic-
lic03/mgcl0d/utilization.db
DB_FILE : ../../database/licenseMonitor/db/license_server/1717@ic-
lic03/mgcl0d/utilization.db
TABLES  :
=====
```

```
caldesignrev  
caldfmproperty  
calibreadp  
calibredrc  
calibrehdrc  
calibrehlvs  
calibrelvls  
calibreperc  
calibrepercadv  
...
```

示例二，查看指定 database 下指定 table 的内容。

```
[liyanqing.1987@n232-135-067 licenseMonitor]$ seedb -  
d ../../database/licenseMonitor/db/license_server/1717@ic-  
lic03/mgcl/d/utilization.db -t calibrepercadv  
DB_FILE : ../../database/licenseMonitor/db/license_server/1717@ic-  
lic03/mgcl/d/utilization.db  
TABLE   : calibrepercadv  
=====
```

sample_second in_use	sample_time	issued
	utilization	
----	----	----
----	----	
1709899860 4	20240308_201100 100.0	4
1709899880 4	20240308_201120 100.0	4
1709900096 4	20240308_201456 100.0	4

```
=====
```

示例三，查看指定 database 下指定 table 中指定 keys 的内容。

```
[liyanqing.1987@n232-135-067 licenseMonitor]$ seedb -  
d ../../database/licenseMonitor/db/license_server/1717@ic-  
lic03/mgcl/d/utilization.db -t calibrepercadv -k sample_time issued  
in_use  
DB_FILE : ../../database/licenseMonitor/db/license_server/1717@ic-  
lic03/mgcl/d/utilization.db  
TABLE   : calibrepercadv
```

```

=====
sample_time      issued      in_use
-----
20240308_201100 4          4
20240308_201120 4          4
20240308_201456 4          4
=====

```

示例四，查看指定 database 下指定 table 中指定 keys 的第一行内容。

```

[liyanqing.1987@n232-135-067 licenseMonitor]$ seedb -
d ../../database/licenseMonitor/db/license_server/1717@ic-
lic03/mgcl03/utilization.db -t calibrepercadv -k sample_time issued
in_use -n 1
DB_FILE : ../../database/licenseMonitor/db/license_server/1717@ic-
lic03/mgcl03/utilization.db
TABLE   : calibrepercadv
=====
sample_time      issued      in_use
-----
20240308_201100 4          4
=====

```

5.11 patch

patch 是帮助 licenseMonitor 打补丁的工具，其帮助信息如下。

```

[liyanqing.1987@n232-135-067 licenseMonitor]$ 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，licenseMonitor 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。

```
[liyanqing.1987@n232-135-067 licenseMonitor]$ tools/patch -p  
/ic/data/usr/liyanqing.1987/tools/licenseMonitor.patch  
Install Path : /ic/data/usr/liyanqing.1987/tools/licenseMonitor  
Patch path :  
/ic/data/usr/liyanqing.1987/tools/licenseMonitor.patch  
  
*Warning*: current install path name is "licenseMonitor", but  
patch path name is "licenseMonitor.patch".  
Do you want to continue? (y|n) y  
  
> Copying python file  
"/ic/data/usr/liyanqing.1987/tools/licenseMonitor.patch/install.py"  
" into  
"/ic/data/usr/liyanqing.1987/tools/licenseMonitor/install.py".  
> Copying python file  
"/ic/data/usr/liyanqing.1987/tools/licenseMonitor.patch/bin/licenses_  
monitor.py" into  
"/ic/data/usr/liyanqing.1987/tools/licenseMonitor/bin/licenses_  
monitor.py".  
> Copying python file  
"/ic/data/usr/liyanqing.1987/tools/licenseMonitor.patch/tools/mess  
age.py" into  
"/ic/data/usr/liyanqing.1987/tools/licenseMonitor/tools/message.py  
".  
> Copying python file  
"/ic/data/usr/liyanqing.1987/tools/licenseMonitor.patch/tools/patch.py"  
" into  
"/ic/data/usr/liyanqing.1987/tools/licenseMonitor/tools/patch.py".
```

六、最佳实践

licenseMonitor 在费用分摊功能上配置文件较多，且配置内容经常面临变更，会给使用方带来一定的困扰，为了解决这个问题，我们总结了一些使用上的最佳实践，将核心文件的更新做到自动化，从而在一定程度上减轻 licenseMonitor 维护者的工作量。

6.1 config/LM_LICENSE_FILE 自动化更新

如果采用 module files 来管理环境，那么在 config/LM_LICENSE_FILE 初始化完成后，可以使用工具 tools/gen_LM_LICENSE_FILE 来定时自动更新 config/LM_LICENSE_FILE 配置文件。

可以采用 crontab 的方式定时自动更新，更新 crontab 如下。

```
[root@ic-admin2 license_server]# crontab -l
SHELL=/bin/bash
PATH=/ic/software/tools/anaconda/Anaconda3-
2021.05/bin:/ic/software/cad_tools/bin:/usr/local/bin:/bin:/usr/bi
n:/usr/local/sbin:/usr/sbin:/ic/software/tools/lsf/10.1/linux3.10-
glibc2.17-x86_64/bin

# For licenseMonitor, update config/LM_LICENSE_FILE every hour.
1 * * * *
/ic/data/usr/liyanqing.1987/tools/licenseMonitor/tools/gen_LM_LICE
NSE_FILE -m /ic/software/modules/config/EDA -f
/ic/data/usr/liyanqing.1987/tools/licenseMonitor/config/LM_LICENSE
_FILE
```

6.2 config/project/project_execute_host 自动化更新

如果采用 LSF 作为集群管理系统，绝大部分 EDA 仿真节点都在 LSF 中，并且使用 EDA license feature 的 execute_host 作为费用分摊的主要依据，那么在 config/project/project_execute_host 初始化完成后，可以使用工具 tools/update_project_execute_host_with_lsf 来自动更新这一配置文件。

可以采用 crontab 的方式自动更新，更新 crontab 如下。

```
[root@ic-admin2 license_server]# crontab -l
SHELL=/bin/bash
PATH=/ic/software/tools/anaconda/Anaconda3-
```

```
2021.05/bin:/ic/software/cad_tools/bin:/usr/local/bin:/bin:/usr/bin:/usr/local/sbin:/usr/sbin:/ic/software/tools/lsf/10.1/linux3.10-glibc2.17-x86_64/bin

# For licenseMonitor, update config/project/project_execute_host
every hour.
2 * * * *
/ic/data/usr/liyanqing.1987/tools/licenseMonitor/tools/update_project_execute_host_with_lsf -p
/ic/data/usr/liyanqing.1987/tools/licenseMonitor/config/project/project_execute_host -o
/ic/data/usr/liyanqing.1987/tools/licenseMonitor/config/project/project_execute_host
```

6.3 config/others/product_feature.yaml 自动化更新

工具 tools/update_product_feature_relationship 可以自动根据 config/LM_LICENSE_FILE 抓取所有的 license file，然后自动解析所有的 cdsImd/snpsImd/mgclD 的 license file，并将映射关系更新到指定的文件中。

tools/others/product_feature.yaml 初始化完毕后，后续可以采用 crontab 的方式定时自动更新，更新 crontab 如下。

```
[root@ic-admin2 license_server]# crontab -l
SHELL=/bin/bash
PATH=/ic/software/tools/anaconda/Anaconda3-
2021.05/bin:/ic/software/cad_tools/bin:/usr/local/bin:/bin:/usr/bin:/usr/local/sbin:/usr/sbin:/ic/software/tools/lsf/10.1/linux3.10-glibc2.17-x86_64/bin

# For licenseMonitor, update config/others/product_feature.yaml
every hour.
3 * * * *
/ic/data/usr/liyanqing.1987/tools/licenseMonitor/tools/update_product_feature_relationship -l
/ic/data/usr/liyanqing.1987/tools/licenseMonitor/config/LM_LICENSE_FILE -i -o
/ic/data/usr/liyanqing.1987/tools/licenseMonitor/config/others/product_feature.yaml
```

6.4 config/others/license_log.yaml 自动更新

工具 tools/get_license_log 可以自动根据 config/LM_LICENSE_FILE 获取所有的 license server，然后解析 license server 和 license log 的对应关系，并将之生成到指定的 license_log.yaml 文件中。

tools/others/license_log.yaml 初始化完毕后，后续可以采用 crontab 的方式定时自动更新，更新 crontab 如下。

```
[root@ic-admin2 license_server]# crontab -l
SHELL=/bin/bash
PATH=/ic/software/tools/anaconda/Anaconda3-
2021.05/bin:/ic/software/cad_tools/bin:/usr/local/bin:/bin:/usr/bi
n:/usr/local/sbin:/usr/sbin:/ic/software/tools/lsf/10.1/linux3.10-
glibc2.17-x86_64/bin

# For licenseMonitor, update config/others/license_log.yaml
1 0 * * *
/ic/data/usr/liyanqing.1987/tools/licenseMonitor/tools/get_licens
e_log -l
/ic/data/usr/liyanqing.1987/tools/licenseMonitor/config/LM_LICENSE
_FILE -o
/ic/data/usr/liyanqing.1987/tools/licenseMonitor/config/others/lic
ense_log.yaml
```

6.5 config/others/feature_record_on_license_log.yaml 自动更新

工具 tools/collect_feature_record_from_license_log 可以自动根据 config/LM_LICENSE_FILE 获取所有的 license feature，根据 config/others/license_log.yaml 获取 license server 和 license log 的对应关系，然后据此从 license log 中检索所有 license feature 的记录，并将之生成到指定的 feature_record_on_license_log.yaml 文件中。

tools/others/feature_record_on_license_log.yaml 初始化完毕后，后续可以采用 crontab 的方式定时自动更新，更新 crontab 如下。

```
[root@ic-admin2 license_server]# crontab -l
SHELL=/bin/bash
PATH=/ic/software/tools/anaconda/Anaconda3-
2021.05/bin:/ic/software/cad_tools/bin:/usr/local/bin:/bin:/usr/bi
```

```
n:/usr/local/sbin:/usr/sbin:/ic/software/tools/lrf/10.1/linux3.10-
libc2.17-x86_64/bin

# For licenseMonitor, update
config/others/feature_record_on_license_log.yaml
10 0 * * 7
/ic/data/usr/liyanqing.1987/tools/licenseMonitor/tools/collect_fea-
ture_record_from_license_log -L
/ic/data/usr/liyanqing.1987/tools/licenseMonitor/config/LM_LICENSE
_FILE -l
/ic/data/usr/liyanqing.1987/tools/licenseMonitor/config/others/lic-
ense_log.yaml -o
/ic/data/usr/liyanqing.1987/tools/licenseMonitor/config/others/fea-
ture_record_on_license_log.yaml
```

七、常见问题及解决

7.1 图形显示问题

问题描述：

安装后 license_monitor 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 3.8.8 或者兼容版本。
- * python 库安装不全。

解决方案：

使用推荐的 python3.8.8 或者 anaconda3-2021.05，按照 3.2 章节的方法安装 requirements.txt 的 python 依赖库。

7.2 license 信息没有或者不全

问题描述：

表现为如下几种现象。

- * tools/gen_LM_LICENSE_FILE 无法抓取全量的 license server 信息。
- * bin/license_sample 抓取不到或者抓取不全 license 信息。
- * bin/license_monitor 抓取不到或者抓取不全 license 信息。

问题原因：

licenseMonitor 使用 lmstat 工具抓取 license 信息，所以应该是在执行 lmstat 环节出现了问题，有如下几种可能。

- * config/LM_LICENSE_FILE 没有配置。
- * tools/lmstat 存在兼容性问题。（小概率）
- * config/config.py 中 lmstat_bsub_command 配置问题。（大概率）

解决方案：

* 检查 config/LM_LICENSE_FILE 是否配置，如未配置，按照 3.3.2 章节的建议格式配置。

- * 在执行上述工具的机器上直接执行"lmstat -a -i"命令，看能否获取有效输出。
 - 如果能，那么把 config/config.py 中的 lmstat_bsub_command 设置为默认值“”。

- 如果不能，那么把 config/config.py 中的 lmstat_bsub_command 设置上争取的 bsub 命令。

* 如果上述方法不解决问题，尝试在 config/config.py 中将 lmstat 指向现在在用的 EDA 工具安装目录下的 lmstat 试试。

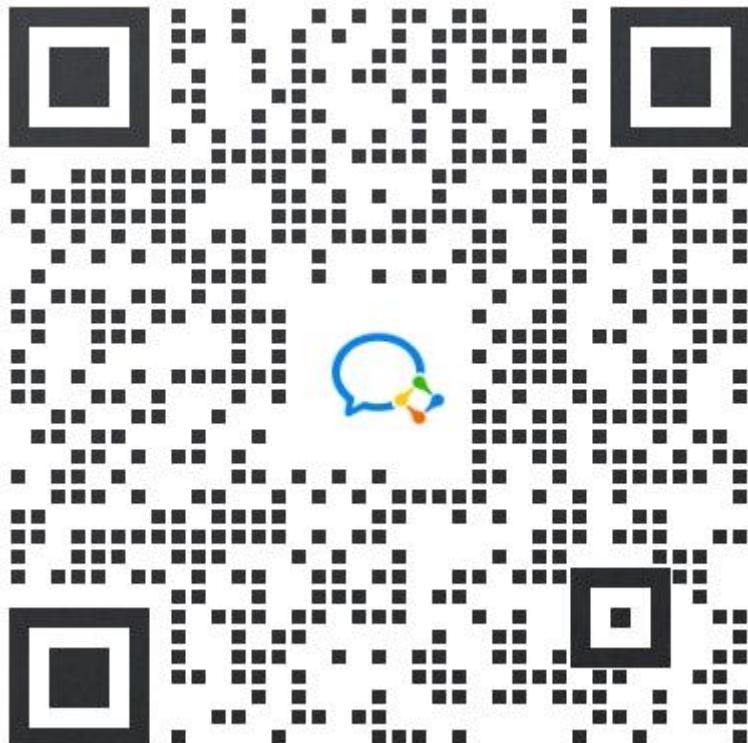
八、技术支持

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。
- 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	备注
2023.1	1.0	发布第一个正式 release 版本	
2023.7	1.1	<ul style="list-style-type: none">增加采用程序 license_sample。license_monitor 增加了 license feature 使用率 (utilization) 分析和计费 (cost) 分摊功能。	
2023.11	1.2	<ul style="list-style-type: none">增加 license server 日志查询功能。增加 LM_LICENSE_FILE 配置文件生成脚本 gen_LM_LICENSE_FILE.py。UTILIZATION/COST 页面支持按照 feature/product 的维度进行过滤。	数据库目录结构跟旧版本不兼容。
2023.12	1.3	<ul style="list-style-type: none">增加了 CURVE 页，用于查询 feature 的 Total/In_Use/Peak 信息。修复了 utilization 计算方式的 bug。增加了 logo 和菜单栏图标。优化了折线图显示效果。增加了 tools/update_*脚本，用于配置文件更新自动化。	
2023.3	1.3.1	<ul style="list-style-type: none">EXPIRES 页增加了 license file 检索功能。UTILIZATION/COST 页支持 license log 的检索功能。UTILIZATION/COST 页支持 product 的快速模糊检索。增加 tools/collect_feature_record_from_license_log.py、tools/get_license_log.py、	

	tools/seedb.py、 tools/show_license_log_info.py、 tools/view_product_feature_relationship.py 等 辅助工具。	
--	---	--