

CMG 2024 软件在 Linux 上的安装流程说明

一、概览

在 Linux 工作站上安装 CMG 软件包含以下几个步骤：

1. 确保工作站满足系统要求。
2. 使安装程序可用于工作站。
3. 确定您使用的许可和 CEIP 配置。
4. 通过安装脚本在工作站上安装软件。
5. 配置环境。
6. 如果计算机是网络许可服务器或独立客户端，则安装许可文件。CMG 尽可能使用 Bourne shell 脚本自动化这些步骤。

注意：CMG 软件提供两个版本。Windows 版本名称中有'W'，而 Linux 版本名称中有'U'。例如，2023.108.GW 是 Windows 版本，2023.108.GU 是 Linux 版本。

二、快速开始

安全升级至 RLM 12.3

我们已将 Reprise 的 RLM 软件版本升级至 12.3。我们建议任何升级其许可服务器的人执行以下操作：

1. 检查是否有正在使用的浮动许可证。
2. 停止任何长时间运行的模拟并为它们编写重启文件。

注意：如果不执行上述操作，可能会导致不可预测的后果。

安装最佳实践

1. 在 Linux 工作站上安装 CMG 软件之前，应确保工作站满足最低系统要求。“最低系统要求”和“大型模型模拟推荐的硬件”部分分别描述了 CMG 软件的最低和推荐系统规格。
2. 安装 CMG 软件需要访问安装文件。这可能需要对计算机具有 root 访问权限。

系统要求

最低系统要求

CMG 软件可以在以下平台上运行：

- 至少安装有 Red Hat Enterprise Linux 7.2 操作系统的 Intel x64 计算机，推荐使用 Red Hat Enterprise Linux 8.4。注意：CMG 软件在列出的级别上进行了测试。只要与列出的 OS 级别向后兼容，就可以使用更高的操作系统级别。

每个平台还必须具备以下条件：

- 8 GB RAM
- 4 GB 硬盘空闲空间

大型矿场模型模拟推荐的硬件配置

- 模拟多达 2,000,000 个有效网格的中型模型（对于 IMEX™, GEM™, 和 STARS™）：
 - 1 或 2 个八核 Intel Xeons 或更好
 - 32 GB RAM 或更多
- 模拟多达 5,000,000 个有效网格的大型模型：
 - 2 个二十核 Intel Xeon Gold 6148 或类似产品
 - 96 GB RAM 或更多
- 模拟超过 5,000,000 个有效网格的非常大模型：
 - 4 个十八核 Intel Xeon Gold 6154 或类似产品
 - 192 GB RAM 或更多

检查您的操作系统级别

Linux 提供了多个选项来显示您的操作系统级别。在某些情况下，操作系统名称和级别会在登录计算机时显示。要确定每个平台的 OS 级别，请使用以下方法：

Red Hat Linux

文件 `.../etc/redhat-release` 包含当前版本信息。使用以下命令查看：

```
cat /etc/redhat-release
```

这将给出以下格式的输出：

```
Red Hat Enterprise Linux Server release 6.2 (Santiago)
```

要找到确切的内核和平台，使用 `uname` 命令和 `-inrs` 选项如下：

```
uname -inrs
```

这将给出以下格式的输出：

```
Linux valhalla 2.6.9-42.ELsmp x86_64
```

在这种情况下，“Linux”是操作系统名称。“valhalla”是计算机的名称。2.6.9-42.ELsmp 是版本（内核）级别。“x86_64”是 x64 的机器类型。对于 32 位计算机，机器类型将是“i686”。

准备安装

在开始安装过程之前，请按照以下步骤进行准备：

登录为 root 用户

您可能需要以 root 用户身份登录，以确保安装脚本能够正确运行。安装脚本可能无法以普通用户身份将文件安装到指定的位置。此外，普通用户可能无法挂载安装文件。

检查是否有正在运行的 CMG 软件

安装新版本的 CMG 软件不应干扰任何当前已安装的 CMG 版本。如果计算机上有任何 CMG 软件正在运行，可以在安装新软件的同时继续运行。

如果您确实需要确定是否有 CMG 软件正在运行，可以使用 ps 命令单独搜索每个模拟器的运行进程：

```
ps -edaf | grep st
```

```
ps -edaf | grep mx
```

```
ps -edaf | grep gm
```

这些命令将列出任何运行中的进程，其中包含“st”，“mx”或“gm”。CMG 的软件遵循其模拟器的特定命名约定：

```
styyyyxx.exe
```

```
mxyyyyxx.exe
```

```
gmyyyyxx.exe
```

其中 yyyy 是发布年份，xx 是该年份内的版本号。例如，st202320.exe 是 2023 年 STARS™ 的第 20 版。

访问安装文件

确保计算机上可以访问安装文件，并记录它们的存放位置。这可能包括挂载外部媒体设备（CD-ROM、U 盘等）、直接从互联网下载文件，或简单地复制文件。文件到达后，目录和其中的文件应该是可读和可执行的。您的本地 IT 部门应该能够协助您完成这些操作。

确定许可配置

CMG 软件在 Linux 上支持两种不同的许可配置。您使用哪一种取决于您购买的许可模型类型。

- 网络许可客户端：许可证由网络许可服务器计算机在网络的其它位置提供。作为安装的一部分，您需要知道网络许可服务器计算机的名称。
- 网络许可服务器：计算机将向网络上的其他网络许可客户端计算机提供许可证。您可以选择在这台计算机上安装所有 CMG 软件，或者只选择安装许可服务器软件。网络许可服务器需要一个许可文件。

网络许可服务器也是客户体验改进计划（CEIP）服务器。如果您选择网络许可服务器安装，将有选项自动启动 CEIP 软件。

在安装过程中，系统会提示您是否希望为计算机生成 `hostid`。只有当这台计算机将作为网络许可服务器时，您才需要执行此操作。

安装完成后，您需要设置 `CMG_LIC_HOST` 环境变量，以指向提供许可证的计算机的名称。对于这两种许可配置都是如此。

安装软件

CMG 提供了一个名为 `unixinst.sh` 的安装脚本作为安装的一部分。这个脚本是一个 Bourne shell 脚本，您可以查看它以了解它的功能。

要开始使用脚本进行安装，请转到安装文件目录。例如，如果安装是 CD-ROM 的一部分，并且挂载在 `/cdrom` 下，您将使用以下命令：

```
cd /cdrom
```

如果文件在另一个目录中（例如文件共享），您需要更改到那个位置。

要从安装目录运行脚本，您将输入：

```
./unixinst.sh
```

按照脚本执行时的指示进行操作。脚本需要以下信息：

- 选择您希望安装软件的路径（`/opt/cmg` 是一个很好的位置，但只要有足够的空间，软件可以安装在任何位置）。
- 安装文件所在的路径。即上面的路径。

- 此计算机是否将用作网络许可服务器。
- 如果它是网络许可服务器，您是否希望 Customer Experience Improvement Program (CEIP) 软件自动运行。
- 是否希望为正在安装的计算机生成 Hostid (如果计算机是网络许可服务器，则需要)。

重要提示：如果此计算机将是一个计算机集群（如 LSF、OGE 或 PBS）的一部分，您将需要遵循额外的安装步骤，例如为新节点配置集群、确保文件正确共享，以及将 cmgsimrun 可执行文件安装到 PATH 中。从 2020.10 版本开始，CMG Linux 安装脚本现在在 Launcher 文件夹下安装 cmgsimrun 可执行文件，并在 /usr/local/bin 中添加符号链接。有关特定于您的集群软件的信息，请参阅 Launcher 用户指南中的“设置远程调度器”部分。

发送 Hostid 信息至 CMG

如果您选择在运行安装脚本时生成 hostid 信息，将在您的 CMG_HOME 目录中创建一个名为 hostid.info 的文件。接下来，您需要将此文件发送给我们，以便为您的计算机创建一个相应的许可文件。

hostid.info 文件中留有空间供您添加姓名、公司名称、地点、电话和传真号码。如果由于某种原因，您没有在这台计算机上安装贵公司有权安装的所有软件（例如，如果您在另一个位置运行一些软件），请在提供的空间中告诉我们您希望获得授权码的产品。

当您在 hostid.info 中添加完所需信息后，可以发送给我们，将生成一个许可文件，并将其发送回给您。

为不同的工作站生成 hostid

如果您希望在不同于安装 CMG 软件的计算机上运行许可证管理器，您可以通过运行 cmghostid 实用程序来生成 hostid。如果您将软件安装在 /opt/cmg 中，您将输入：（请注意，CMG_HOME 必须设置正确，以便此脚本正常工作）。

```
/opt/cmg/rlmsecure/cmghostid.sh
```

这将产生类似于以下输出的结果（实际数字将不同）：

```
HOST ID INFORMATION (VERSION 2022.00)
```

Please fax or email this form to CMGL at:

Computer Modelling Group Ltd.

Attention: License Coordinator

Address: 3710 33 Street NW. Calgary, AB., Canada T2L 2M1

Phone: 1-403-531-1300 Fax: 1-403-289-8502

Email: licenses@cmgl.ca

Customer Information

Name: Jane Doe Company: Doe & Associates Ltd.

Location: Calgary, AB Phone: 403-555-1234

Fax: 403-555-5678

Email: jane.doe@da.com

Site Code and License Configuration Information:

Host name: valhalla

Host ID Information:

001e4fdf75a7

ip=123.45.67.89

host=valhalla

Operating System Red Hat Enterprise Linux Server release 6.2 (Santiago)

OS Info Linux 2.6.18-128.el5 x86_64

MACHTYPE x86_64-redhat-linux-gnu

CMG_PLATFORM linux_x64

Environment Variables:

CMG_HOME=/opt/cmg

CMG_LIC_HOST=valhalla

启动许可服务器

如果您的计算机将作为许可服务器使用,您可以使用 `RestartRLM` 脚本手动启动许可证服务。该脚本可以在 `CMG_HOME` 目录中找到。此外,您还可以设置在 Unix 系统启动时自动启动许可证服务器守护进程。

注意: 尽管您现在可以启动许可服务器,但在安装由 CMG 提供的许可文件之前,它不会为您提供任何许可证。

您可以通过 **Web** 浏览器检查许可证的状态。启动浏览器并访问 URL

<http://localhost:5054/>。有关更多信息，请参考《CMG 许可使用指南》。

手动启动 RLM 服务器

在 shell 终端中运行 RestartRLM.sh 脚本。

在 Unix 系统上设置在系统启动时启动 RLM 服务器

在大多数 Unix 系统上，系统服务在启动时通过位于/etc/rc.<something>的启动脚本启动。例如，在 Linux 系统上，脚本可能位于/etc/init.d/rlm，并且链接到/etc/rc5.d/S98rlm。注意，您必须以 root 用户身份安装此启动脚本。

启动脚本应该切换到不同的用户，以便 rlm 服务器不会以 root 身份运行。

以下是一个在 Unix 系统上设置在系统启动时启动 rlm 的脚本示例。请根据您的目标系统修改前 5 个变量。

```
#!/bin/sh

#

# rlm Start/Stop rlm

#

#-----

# NOTE: Configure these 6 variables for your system
# Set rlmuser to the user under which rlm will run
rlmuser=maxwell

# Set root directory for sub-platforms
rlmsecure=/opt/cmgl/rlmsecure

# Set rlmdir to the directory where the rlm binary is found
rlmdir=${rlmsecure}/linux_x64

# Set rlmdowndir to the directory where the rlmdown binary is found
rlmdowndir=${rlmdir}

# Set licfile to the path to the license file
licfile=${rlmsecure}/cmgl.lic

# Set debuglog to the path to the debug log
debuglog=+${rlmsecure}/logs/rlm.dl

#-----

start() { echo $debuglog su - $rlmuser -c "$rlmdir/rlm -c $licfile -dlog $debuglog
```

```

&" }
stop() { su - $rlmuser -c "$rlmdowndir/rlmdown RLM -q" }
case "$1" in
    start) start ;;
    stop) stop ;;
    restart) stop; sleep 2; start ;;
    *) echo $"Usage: $0 {start|stop|restart}"
    exit 1
esac
exit 0

```

这个脚本提供了一个基本的框架，您可能需要根据您的具体系统环境进行调整。

设置环境

为了在 Linux 计算机上正确运行 CMG 软件，您必须确保环境已正确配置。这包括设置一些环境变量，并确保您已安装适当的运行时库。

必需的环境变量

CMG 软件需要设置一些环境变量以确保正常运行。您必须在安装结束时设置这些变量，无论是手动每次使用软件时设置，还是在适当的启动文件中设置（例如 `.cshrc`, `.kshrc`, `.login` 等）。

下表描述了这些环境变量：

环境变量	描述
CMG_HOME	CMG 软件安装的目录。建议所有版本的 CMG 软件都安装在 CMG_HOME 下。
CMG_LIC_HOST	运行许可服务器的计算机的名称或 IP 地址，该计算机将被此计算机上运行的 CMG 应用程序所使用。您可以指定计算机名称、带有域的完全合格的计算机名称，或 IP 地址。如果此计算机将用作许可服务器，您可以使用 127.0.0.1 或计算机的名称。如果许可服务器使用了除默认端口以外的端口，您可以使用 name:port 语法。

LD_LIBRARY_PATH	<p>模拟器使用的库文件的位置。每个模拟器都有一个 lib 目录，应将其添加到路径中。给定模拟器的 lib 将位于： <CMG_HOME>/<simulator>/<version>/<platform>/lib 例如，如果 CMG_HOME 是 /opt/cmg, STARS 2023.20 在 Linux_x64 上的库目录将是 /opt/cmg/stars/2023.20/Linux_x64/lib</p>
-----------------	---

如果您使用的是 C-shell，您将希望在 .cshrc 文件中添加以下行：

```
set CMG_HOME=/put/your/CMG/path/in/here
set CMG_LIC_HOST=computer_name:port
set LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/lib/path/here
```

如果您运行的是 Korn shell（或其衍生产品），请添加这些行：

```
export CMG_HOME=/put/your/CMG/path/in/here
export CMG_LIC_HOST=computer_name:port
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/lib/path/here
```

注意：当前版本的 CMG 软件支持在单个 LD_LIBRARY_PATH 环境变量上所有三个模拟器的多个版本。然而，未来版本可能会出现冲突。您可能想要限制 LD_LIBRARY_PATH 中的版本数量以避免冲突。您也可以使用 RunSim 脚本（在 Error! Reference source not found. 中描述）在每次运行时设置变量。

Linux_x64 运行时环境

为了在不同的 Linux 实现中运行单个可执行文件，CMG 模拟器使用 glibc 运行时库的动态链接，IMEX 和 STARS 使用 Intel 运行时库的动态链接。GEM 与 Intel 库静态链接。

GEM、IMEX 和 STARS 2023.20 版本是在 RHEL 7.2 上构建的，并且支持 RHEL 7.2、RHEL 7.5、RHEL 7.6、RHEL 7.7、RHEL 8.1、RHEL 8.2、RHEL 8.3 和 RHEL 8.4。

GEM、IMEX 和 STARS 2023.20 版本是使用 INTEL Fortran 和 INTEL C/C++ 编译器版本 19.0.5.281 构建的。编译器版本可以通过使用命令行参数，不加引号地输入 -compiler_info 来确定。

需要标准 C++ 和 gcc x64 运行时库。有关详细信息，请参阅

Readme_Linux_libs_2023.20.pdf 文件。

对于 IMEX 和 STARS，必须设置环境变量 LD_LIBRARY_PATH，以便可执行文件可以定位 Intel 运行时库。对于 GEM，不需要设置 LD_LIBRARY_PATH，因为 GEM 与 Intel 库静态链接。

设置 LD_LIBRARY_PATH

为此，运行时库存储在以下位置：

```
${CMG_HOME}/simulator/version/platform/lib
```

其中 simulator = IMEX/STARS； version = 当前版本，例如 2023.20； platform = linux_x64。

这假设在安装期间设置的环境变量 CMG_HOME 指向模拟器安装的位置。

例如，如果 IMEX 安装在 /usr/cmg，那么对于 bash shell 和 IMEX 2023.20 在 linux_x64 上，环境变量 LD_LIBRARY_PATH 必须设置如下：

```
export LD_LIBRARY_PATH=/usr/cmg/imex/2023.20/linux_x64/lib
```

CMG Launcher 的 2023.20 版本（支持 LSF、SGE 或 PBS/Torque）如果用户在 Linux_x64 机器上提交作业运行模拟器，将自动为用户设置此环境变量。必须正确设置 CMG_HOME 环境变量。

有关如何确定运行时库版本的详细信息，请参阅文档 Readme_Linux_libs_xxxx.xx.pdf。

要获取所需运行时库的列表，包括 Intel OpenMP 和 MKL 运行时库，可以使用 ldd 命令。例如，在安装有 Linux x64 RHEL 8 更新 4 的机器上，获得的结果是：

```
$ ldd mx202320.exe
```

```
linux-vdso.so.1 (0x00007fffd65cc000)
libz.so.1 => /lib64/libz.so.1 (0x00007f017e1f3000)
librt.so.1 => /lib64/librt.so.1 (0x00007f017dfcb000)
libmkl_intel_lp64.so => /usr/cmg/imex/2023.10/linux_x64/lib/libmkl_intel_lp64.so
(0x00007f017d473000)
libmkl_def.so => /usr/cmg/imex/2023.10/linux_x64/lib/libmkl_def.so
(0x00007f017ac22000)
libmkl_intel_thread.so => /usr/cmg/imex/2023.10/linux_x64/lib/libmkl_intel_thread.so
(0x00007f01786b5000)
```

```
libmkl_core.so      =>      /usr/cmg/imex/2023.10/linux_x64/lib/libmkl_core.so
(0x00007f0174384000)

libm.so.6 => /lib64/libm.so.6 (0x00007f0174002000)

libiomp5.so        =>      /usr/cmg/imex/2023.10/linux_x64/lib/libiomp5.so
(0x00007f0173c0d000)

libpthread.so.0 => /lib64/libpthread.so.0 (0x00007f01739ed000)

libc.so.6 => /lib64/libc.so.6 (0x00007f0173628000)

/lib64/ld-linux-x86-64.so.2 (0x00007f017e40a000)

libgcc_s.so.1 => /lib64/libgcc_s.so.1 (0x00007f0173410000)

libdl.so.2 => /lib64/libdl.so.2 (0x00007f017320c000)
```

Linux_x64 运行时库

对于 2023.20 版本的 Linux_x64 GEM、IMEX 和 STARS, 需要安装三个 Red Hat RPM 软件包以提供 CMG 模拟器所需的运行时库。对于 RHEL 7.2 的软件包如下:

- glibc-2.17-105.el7.x86_64
- libgcc-4.8.5-4.el7.x86_64
- libstdc++-4.8.5-4.el7.x86_64

随着 2022.10 版本的发布, CMG 支持 RHEL 8.4。2023.20 版本是在装有 RHEL 7.2 的服务器上构建的。

检查运行时库

要确定是否安装了所需的 RPM 软件包, 请运行以下命令:

```
rpm -qa glibc
```

```
rpm -qa libgcc
```

```
rpm -qa libstdc++
```

这些命令的输出可能列出了零个、一个或两个已安装的软件包。对于 libstdc++, 可能会列出两个以上, 但请忽略那些名字不完全为 libstdc++-x.x.x-xx 的软件包 (例如, 忽略 libstdc++-devel-x.x.x-xx)。

注意: 32 位软件包以.i686 结尾, 而 64 位软件包以.x86_64 结尾。

安装运行时库

可以通过以下两种方式之一安装所需的 RPM 软件包：

1. 自动使用 Red Hat Linux 更新代理 yum：更新代理自动解决任何依赖关系，因此任何作为另一个软件包先决条件的软件包将首先被安装。有关使用 yum 的更多信息，请参考您的 Red Hat Linux 文档。
2. 手动从 Red Hat Linux 网站下载 RPM，然后安装：请注意，在这种情况下，任何依赖性都需要手动解决。

请与您的系统管理员联系，以帮助选择这些安装方法之一。

安装许可文件

如果您正在安装网络许可客户端，那么您的安装现在已完成。

如果您正在安装独立的客户端或网络许可服务器，并且这台计算机上已经有正在运行的 CMG 软件，那么您的安装现在也已完成。

如果您正在安装一个新的独立客户端或一个新的网络许可服务器，那么您需要将安装过程中生成的 hostid 信息发送给 CMG。我们将生成一个授权您使用已许可的 CMG 应用程序的许可文件，并通过电子邮件将此文件发回给您。

有关安装您的许可文件的信息，请参考《CMG 许可使用指南》。

使用 RunSim 脚本运行模拟

三个模拟器（GEM、IMEX、STARS）都有详尽的手册描述如何运行它们。CMG 还提供了一个名为 RunSim.sh 的脚本来简单地调用模拟器。

RunSim.sh 脚本默认安装在 CMG_HOME 目录中。它被设计为接收用户输入，描述要运行的模拟器的名称和版本，以及输入数据集的位置。基于这些信息，脚本将执行以下操作：

- 根据用户提供的模拟器名称和版本，结合 CMG_HOME 环境变量，确定模拟器可执行文件的完整路径，
- 根据模拟器可执行文件的完整路径确定 LD_LIBRARY_PATH 环境变量的值，
- 用输入数据集的名称启动模拟器。

模拟器的输出将被放置在当前工作目录中。如果数据集的路径包含任何空格，应该使用引号括起来。

您可以查看 RunSim.sh 脚本的内容，并根据需要对其进行定制。还可以给

RunSim.sh 传递额外的命令行选项，这些选项将直接传递给模拟器。

示例 1： 在本地数据文件 sttst01.dat 上运行 STARS 2023.20:

```
bash
```

```
RunSim.sh stars 2023.20 sttst01.dat
```

示例 2： 使用相对路径在数据文件上运行 GEM 2023.20:

```
bash
```

```
RunSim.sh gem 2023.20 ../tpl/frr/gmfr001.dat
```

示例 3： 使用绝对路径在数据文件上运行 IMEX 2023.20:

```
bash
```

```
RunSim.sh imex 2023.20 /opt/cmg/imex/2022.10/tpl/spe/mxspe001.dat
```

示例 4： 在本地数据文件上运行 IMEX 2023.20，并使用额外的命令行选项进行并行处理:

```
RunSim.sh imex 2023.20 mxspe001.dat -doms -parasol 4
```