分析函数执行

您可以使用 Valgrind 工具套件中包含的呼叫格林德工具来检测与执行函数相关的问题。此外，您可以将 Callgrind 生成的数据文件加载到 KCachegrind 配置文件数据可视化工具中，以便浏览性能结果。

下载并安装瓦尔格林德工具和卡切格林德后，您可以使用Qt创建者中的呼叫格林德和卡切格林德。

> **注意：** 您可以在 Linux 上本地安装和运行呼叫格林德和 KCatch 格林德。您可以从任何开发计算机在远程 Linux 计算机或设备上运行调用网格。

构建用于分析的应用

调用格林记录在运行应用程序时执行的函数的调用历史记录。它收集已执行的指令数、指令与源线路的关系、函数之间调用方和被调用方的关系以及此类调用的数。还可以使用缓存模拟或分支预测来收集有关应用程序的运行时行为的信息。

由于调试和发布生成配置的运行时特征有很大不同，因此一个生成配置的分析结果可能与另一个生成配置的分析结果无关。分析调试版本通常会发现花费在低级代码（如容器实现）上的大部分时间，而相同的代码不会显示在同一应用程序的发布版本的配置文件中，因为内联和其他优化通常在那里完成。

许多最新的编译器允许您构建同时存在调试信息的优化应用程序。例如，GCC 的典型选项包括：。建议将这样的设置用于呼叫格林分析。`-g -O2`

收集数据

要分析应用程序：

1. 在“**项目**”模式下，选择发布生成配置。
2. 选择“**调试**”以打开“**调试**”模式，然后在工具栏上选择“**调用**”。

3. 选择该按钮以启动应用程序。
4. 使用应用程序对其进行分析。
5. 选择该按钮可在“**函数**”视图中查看分析结果。

选择此选项可通过暂停事件日志记录在分析期间加快程序执行速度。暂停日志记录时不计算任何事件。

选择此选项可放弃所有收集的数据。

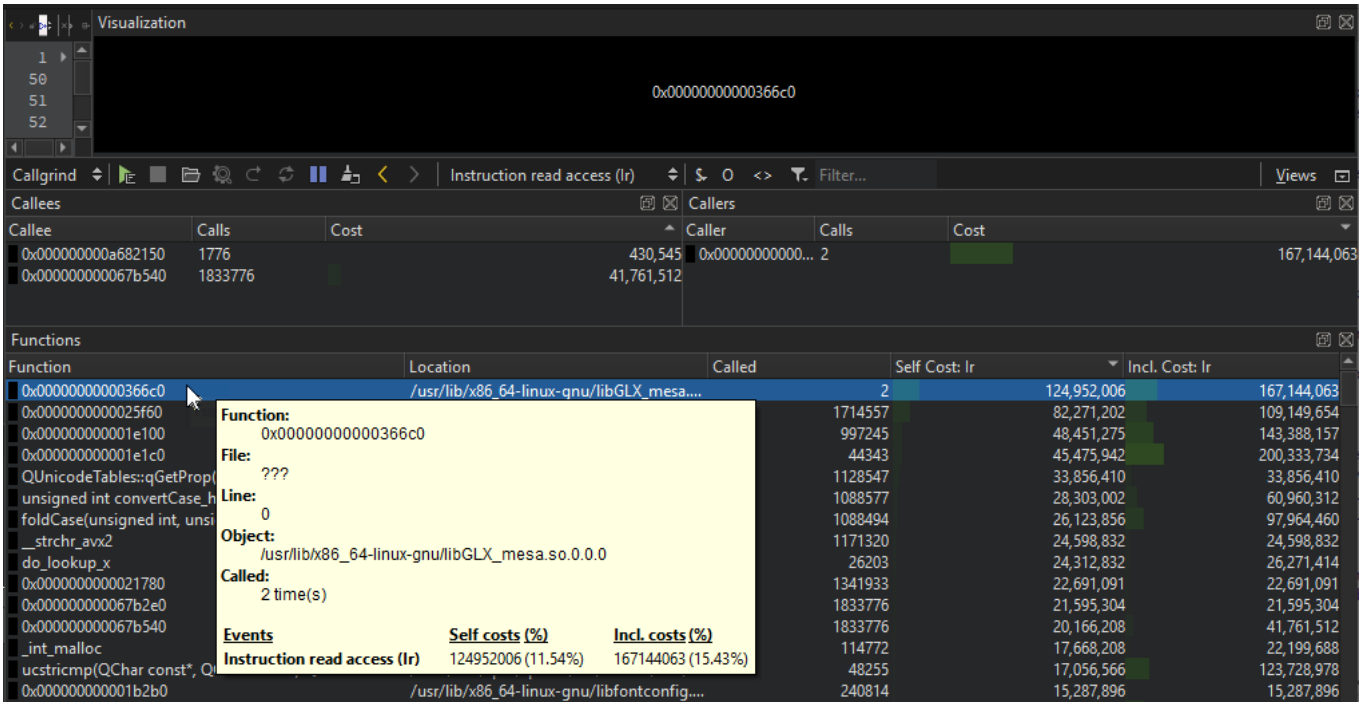
选择此选项可在卡切格林德中查看数据。Qt Creator启动K卡切格林德并将数据加载到其中以进行可视化。

查看收集的数据

分析结果显示在“**调用项**”视图中。您可以分离视图并四处移动。若要还原更改，请选择“**视图**”>“**重置为默认布局**”。

选择“**视图**”以显示和隐藏视图以及视图标题。默认情况下，**可视化视图**处于隐藏状态。选择此选项可在显示数据时刷新其中显示的数据。

作为收集数据的替代方法，您可以选择将外部日志文件加载到 **Callgrind** 视图中。



Enter a string in the **Filter** field to filter the results.

Move the cursor on a function in the **Functions** view for more information about it.

Double-click a function to view information about the calling functions in the **Callers** view and about the called functions in the **Callees** view.

Select < or > To move between functions in the **Callee** view.

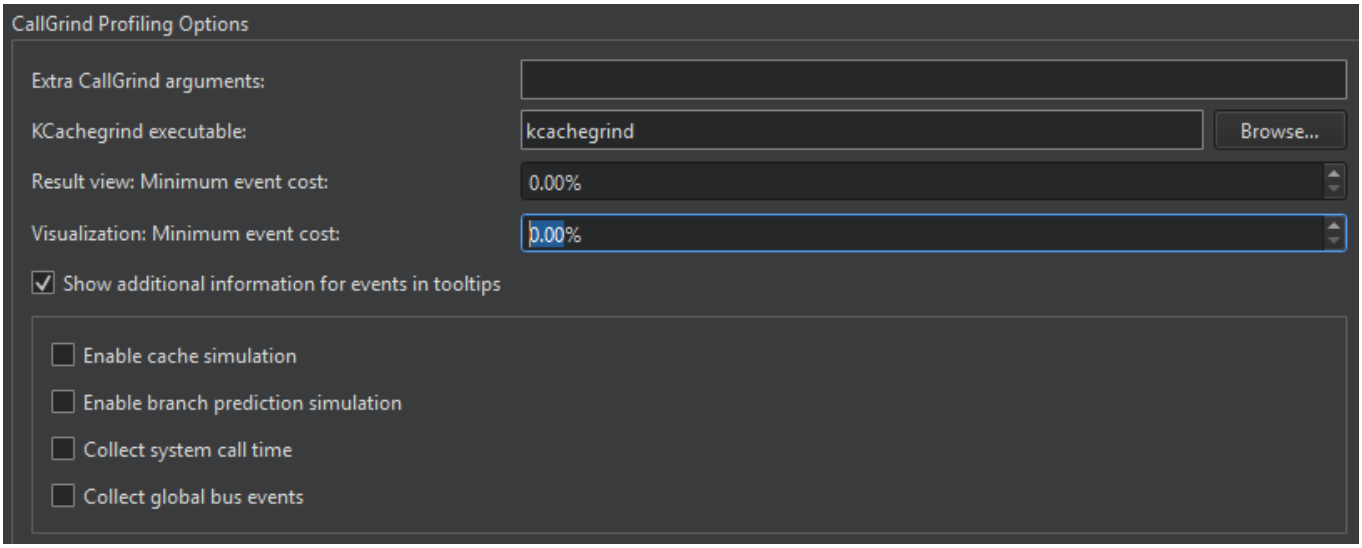
To set the cost format, select **$**. You can view absolute or relative costs, as well as relative costs to parent. Select to view only profiling info that originated from the project.

To properly handle recursive or circular function calls, enable cycle detection by selecting **O**.

To remove template parameter lists when displaying function names, select **<>**.

Selecting Profiling Options

To specify global settings for Valgrind, select **Edit** > **Preferences** > **Analyzer**. The **Callgrind Profiling Options** group contains Callgrind options.



In the **KCachegrind executable** field, enter the path to the KCachegrind executable to launch.

In **Extra Callgrind arguments**, specify additional arguments for launching the executable.

In the **Result view: Minimum event cost** and **Visualization: Minimum event cost** fields, limit the amount of results the profiler presents and visualizes to increase profiler performance.

To show additional information about events in tooltips, select **Show additional information for events in tooltips**.

To collect information about the system call times, select **Collect system call time**. To collect the number of global bus events of the event type that are executed, select **Collect global bus events**.Ge

Enabling Full Cache Simulation

By default, only instruction read accesses (Ir) are counted. To fully simulate the cache, select the **Enable cache simulation** check box. This enables the following additional event counters:

- Cache misses on instruction reads (I1mr/I2mr)
- Data read accesses (Dr) and related cache misses (D1mr/D2mr)
- Data write accesses (Dw) and related cache misses (D1mw/D2mw)

Enabling Branch Prediction Simulation

To enable the following additional event counters, select the **Enable branch prediction simulation** check box:

- Number of conditional branches executed and related predictor misses (Bc/Bcm)
- Executed indirect jumps and related misses of the jump address predictor (Bi/Bim)

‹ Detecting Memory Leaks with Memcheck Running Valgrind Tools on External Applications ›

Qt DOCUMENTATION

The Qt Company

Contact Us

Company

About Us
Investors
Newsroom
Careers
Office Locations

Licensing

Terms & Conditions
Open Source
FAQ

Support

Support Services
Professional Services
Partners
Training

For Customers

Support Center
Downloads
Qt Login
Contact Us
Customer Success

Community

Contribute to Qt
Forum
Wiki
Downloads
Marketplace

Feedback Sign In