

[Qt创作者手册](#) > [性能分析函数执行](#)

# 性能分析函数执行

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，以浏览性能结果。

下载并安装 Valgrind 工具和 KCachegrind 后，您可以使用 Qt Creator 中的 Callgrind 和 KCachegrind。

**注意：**您可以在 Linux 上本地安装和运行 Callgrind 和 KCachegrind。您可以从任何开发计算机在远程 Linux 计算机或设备上运行 Callgrind。

Qt 创建者手册 8.0.2  
Topics >

Callgrind 记录应用程序运行时执行的函数的调用历史记录。它收集执行的指令数、它们与源行的关系、函数之间的调用方和被调用方的关系以及此类调用的数量。还可以使用缓存模拟或分支预测来收集有关应用程序运行时行为的信息。

由于调试和发布生成配置的运行时特征差异很大，因此一个生成配置的分析结果可能与另一个 **生成配置** 无关。分析调试版本通常会发现大部分时间都花在低级代码上，例如容器实现，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同时使用调试信息构建优化的应用程序。例如，GCC 的典型选项是：建议将此类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“**调用格林德**”。



3. 选择按钮  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。

选择重置 所有事件计数器。

选择以 放弃所有收集的数据。

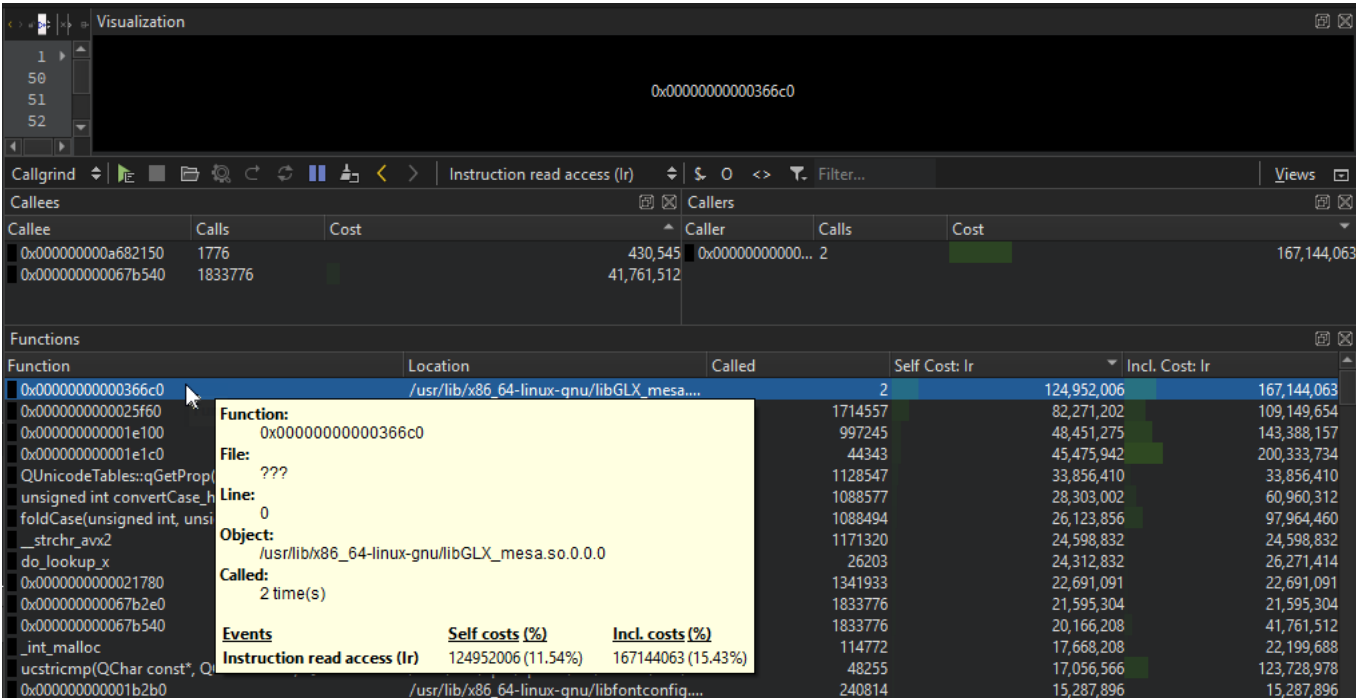
选择以查看 KCachegrind 中的数据。Qt Creator启动KCachegrind并将数据加载到其中以进行可视化。

## 查看收集的数据

分析结果将显示在**呼叫格林德**视图中。您可以分离视图并移动视图。若要还原更改，请选择“**视图>重置为默认布局**”。

选择“**视图**”以显示和隐藏视图和视图标题。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。



在“**筛选器**”字段中输入字符串以筛选结果。

在“**函数**”视图中将光标移到函数上以获取有关该函数的详细信息。

双击函数可查看有关“**调用方**”视图中的调用函数和“**调用方**”视图中被调用函数的信息。

选择 < 或在 > 被调用方视图中的函数之间移动。

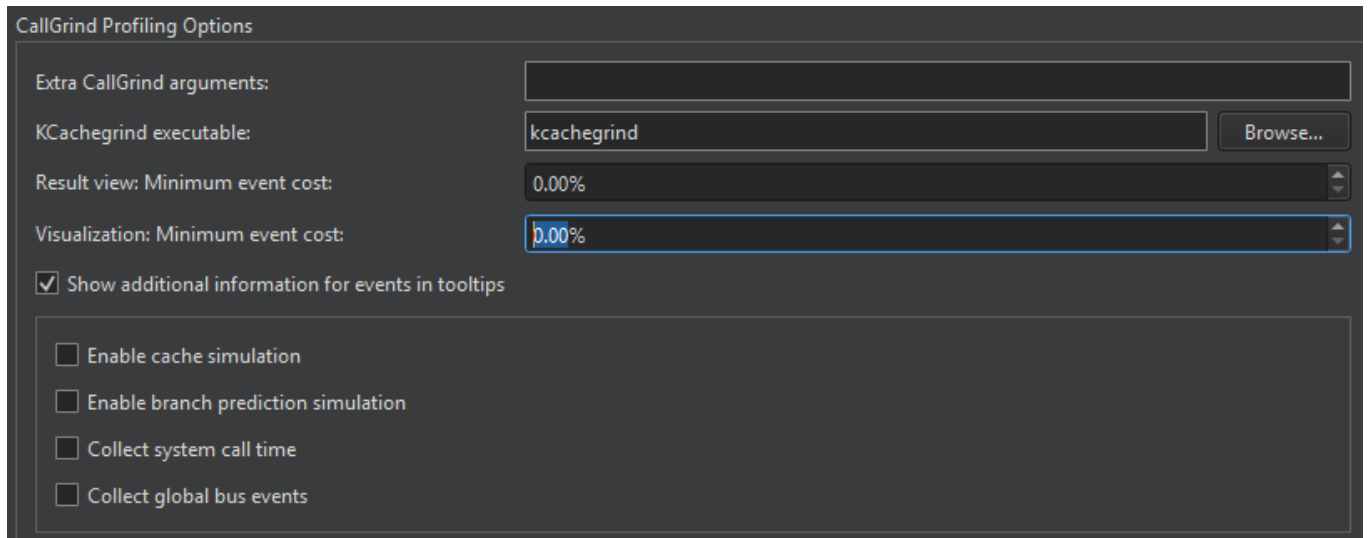
若要设置成本格式，请选择 \$。您可以查看绝对成本或相对成本，以及父级的相对成本。选择此选项 可仅查看源自项目的分析信息。

若要正确处理递归或循环函数调用，请通过选择 O 启用周期检测。

要在显示函数名称时删除模板参数列表，请选择 <>。

可以为所有项目全局指定分析器设置，也可以在项目的运行设置中为每个项目单独指定分析器设置。

要指定 Valgrind 的全局设置，请选择**编辑>首选项>分析器**。“呼叫格林分析选项”组包含“呼叫格林德”选项。



在“KCachegrind 可执行文件”字段中，输入要启动的KCachegrind 可执行文件的路径。

在“额外调用格林德”参数中，指定用于启动可执行文件的其他参数。

在“结果视图：最低事件成本”和“可视化：最小事件成本”字段中，限制探查器显示和可视化的结果量，以提高探查器性能。

若要在工具提示中显示有关事件的其他信息，请选择“在工具提示中显示事件的其他信息”。

若要收集有关系统调用时间的信息，请选择“收集系统调用时间”。若要收集执行的事件类型的全局总线事件数，请选择“收集全局总线事件”。

## 启用完全缓存模拟

默认情况下，仅计算指令读取访问（Ir）。若要完全模拟缓存，请选中“启用缓存模拟”复选框。这将启用以下附加事件计数器：

- › 指令读取时缓存未命中（I1mr/I2mr）
- › 数据读取访问（Dr）和相关缓存未命中（D1mr/D2mr）
- › 数据写入访问（Dw）和相关缓存未命中（D1mw/D2mw）

## 启用分支预测模拟

若要启用以下附加事件计数器，请选中“启用分支预测模拟”复选框：

- › 执行的条件分支数和相关预测因子未命中数（Bc/Bcm）
- › 执行的间接跳转和跳转地址预测器的相关未命中（Bi/Bim）

[< 使用 Memcheck 检测内存泄漏](#)

[在外部应用程序上运行Valgrind工具。>](#)

©2022 Qt有限公司 此处包含的文档贡献的版权归 他们各自的所有者。此处提供的文档根据自由软件基金会发布的GNU 自由文档许可证版本 1.3的条款进行许可。Qt和相应的徽标是Qt有限公司在芬兰和/或全球其他国家的商标。所有其他商标均为财产 其各自所有者。



联系我们

公司

- 关于我们
- 投资者
- 编辑部
- 职业
- 办公地点

支持

- 支持服务
- 专业服务
- 合作 伙伴
- 训练

社区

- 为Qt做贡献
- 论坛
- 维基
- 下载
- 市场

发牌

- 条款和条件
- 开源
- 常见问题

对于客户

- 支持中心
- 下载
- Qt登录
- 联系我们
- 客户成功案例