

分析 CPU 使用率

Qt Creator与Linux Perf工具集成，可用于分析嵌入式设备上应用程序的CPU和内存使用情况，并在有限的程度上分析Linux桌面平台上的应用程序使用情况。性能分析器使用与Linux内核捆绑在一起的Perf工具定期拍摄应用程序的调用链快照，并在时间线视图中或火焰图中将其可视化。

使用性能分析器


性能分析器通常需要能够找到所涉及的二进制文件的调试符号。

配置文件生成具有单独调试符号的优化二进制文件，通常应用于分析。


若要手动设置生成配置以提供单独的调试符号，请编辑项目生成设置：

1. 若要也为在发布模式下编译的应用程序生成调试符号，请选择“**项目**”，然后在“**单独的调试信息**”字段中选择“**启用**”。
2. 选择“**是**”以重新编译项目。

可以通过以下方式启动性能分析器：

- › 选择“**分析>性能分析器**”以分析当前应用程序。
- › 选择“ (**启动**)”按钮以从**性能分析器**启动应用程序。




注意：如果数据收集未自动启动，请选择 (**收集配置文件数据**) 按钮。

当您开始分析应用程序时，将启动该应用程序，并且性能分析器会立即开始收集数据。这由“**记录**”字段中的运行时间指示。但是，由于数据通过Perf工具和与Qt Creator捆绑在一起的额外辅助程序传递，并且动态缓冲和处理它，数据可能会在生成几秒钟后到达Qt Creator。此延迟的估计值在**处理延迟**字段中给出。

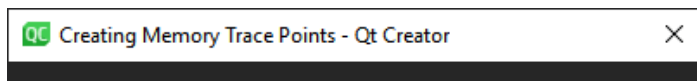
在选择“停止收集配置文件数据”按钮或终止应用程序之前，将**收集数据**。

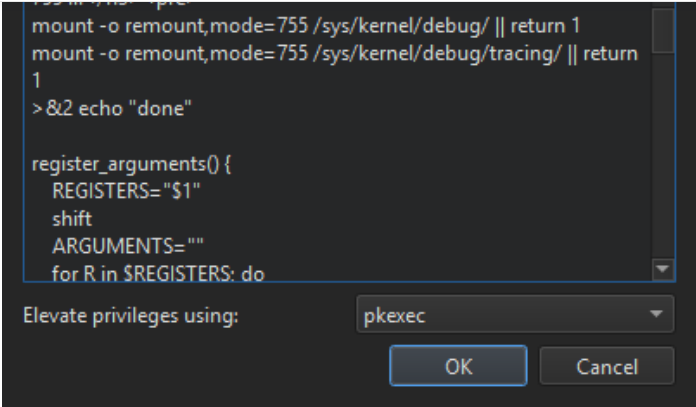
选择“**停止收集配置文件数据**”按钮以禁用在启动应用程序时自动启动数据收集。配置文件数据仍将生成，但Qt Creator将丢弃它，直到您再次选择该按钮。

分析设备上的内存使用情况

若要创建跟踪点以分析目标设备上的内存使用情况，请选择“**分析>性能分析器选项>创建内存跟踪点**”或在“**性能分析器**”工具栏上选择.

在“**创建内存跟踪点**”对话框中，可以修改要运行的脚本。





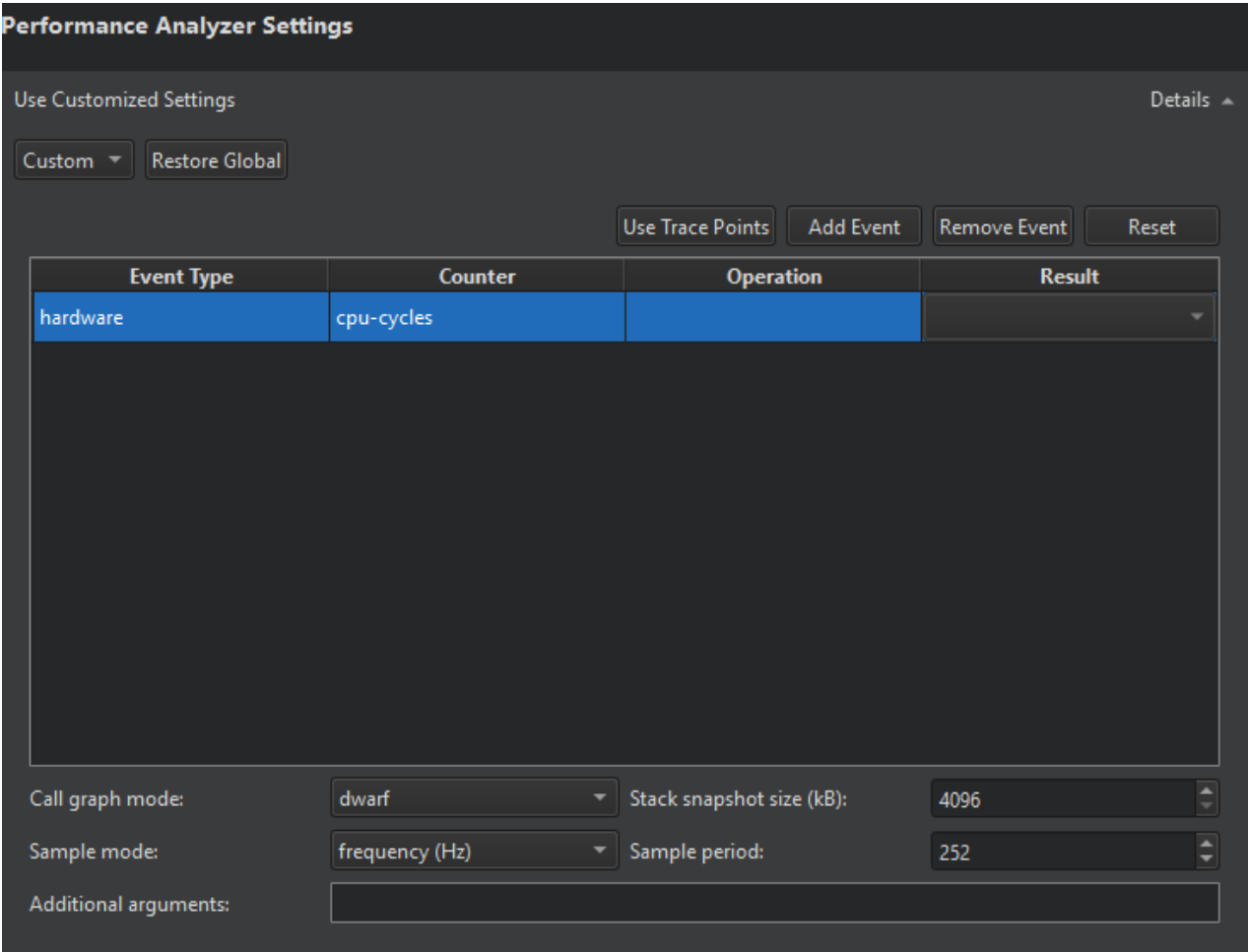
如果需要 root 权限才能以 root 用户身份运行脚本，请在提升权限使用字段中选择要使用的**权限**。
选择**“确定”**以运行脚本。

要为跟踪点添加事件，请参阅[选择事件类型](#)。

您可以记录内存跟踪，以便在时间线的示例行中查看使用情况图，并在火焰图中查看内存分配、峰值和释放。

指定性能分析器设置

若要指定性能分析器的全局设置，请选择**“编辑>首选项”>“分析器>CPU 使用率”**。对于每个运行配置，还可以使用专用设置。选择**“项目>运行”**，然后选择**“性能分析器设置”**旁边的**“详细信息”**。



若要编辑当前运行配置的设置，还可以选择**“收集配置文件数据”**按钮旁边的下拉菜单。

选择事件类型

在“事件”表中，可以指定哪些事件应触发性能分析器获取示例。分析 CPU 使用率的最常用方法涉及定期采样，由对执行的指令数或

器”列中，您可以选择应使用哪个特定**计数器**进行采样。例如，**硬件组**中的**指令**或**软件组**中的**CPU 时钟**。

可以进行更专业的采样，例如缓存未命中或缓存命中。但是，对它的支持取决于所涉及的 CPU 的特定功能。对于这些专用事件，您可以在**“操作”**和**“结果”**列中提供更详细的采样说明。例如，您可以在**加载操作**上为**L1-dcache选择缓存事件**，但结果是**未命中**。这将对读取时的 L1-dcache 未命中进行采样。

选择**“删除事件”**以从表中删除所选**事件**。

选择**“使用跟踪点”**将当前选择的事件替换为目标设备上定义的**跟踪点**，并将**“采样”**模式设置为**“事件计数”**，将**“采样周期”**设置为。如果目标上的跟踪点是使用**“创建跟踪点”**选项定义的，则性能分析器将自动使用它们来分析内存使用情况。¹

选择**“重置”**以将事件选择以及**“采样模式”**和**“采样周期”**还原为默认值。

选择采样模式和周期

在**采样模式**和**采样周期**字段中，您可以指定如何触发样本：

- 按事件计数采样指示内核在每次发生所选**事件**之一时取样，其中在**采样周期**字段中指定。nn
- 按频率采样（Hz）**指示内核通过自动调整采样周期来尝试每秒采样次数。在**采样期间**字段中指定。nn

高频或低事件计数会产生更准确的数据，但代价是更高的开销和生成的数据量更大。实际采样周期由目标设备上的 Linux 内核确定，该内核仅为建议为 Perf 设置的采样周期。您请求的采样周期与实际结果之间可能存在显著差异。

通常，如果将性能分析器配置为收集的数据多于通过目标和主机设备之间的连接可以传输的数据，则在 Perf 尝试发送数据时，应用程序可能会被阻止，并且处理延迟可能会过度增长。然后，您应该更改**采样周期**或**堆栈快照大小**。

选择调用图模式

在**“调用图模式”**字段中，可以指定性能分析器如何从应用程序中恢复调用链：

- 帧指针或模式依赖于分析应用程序中可用的帧指针，并将指示目标设备上的内核遍历**帧指针链**，以便检索每个样本的调用链。fp
- 矮人模式**也可以在没有帧指针的情况下工作，但生成的数据要多得多。每次触发示例时，它都会拍摄当前应用程序堆栈的快照，并将该快照传输到主机进行分析。
- “最后分支记录”**模式不使用内存缓冲区。每次执行停止时，它都会自动解码最后 16 个分支。它仅在最新的英特尔 CPU 上受支持。

默认情况下，Qt 和大多数系统库都是在没有帧指针的情况下编译的，因此帧指针模式仅对自定义系统有用。

设置堆栈快照大小

性能分析器将分析并**展开**Perf 在矮模式下生成的堆栈快照。在堆栈快照大小字段中设置**堆栈快照**的大小。大型堆栈快照会导致要传输和处理的数据量更大。小型堆栈快照可能无法捕获高度递归应用程序的调用链或其他大量堆栈使用。

为性能添加命令行选项

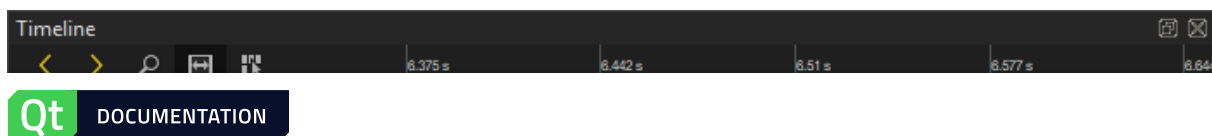
在**“其他参数”**字段中记录数据时，可以指定要传递给 Perf 的其他命令行选项。您可能需要指定 or 以减少处理延迟。但是，并非所有版本的 Perf 都支持这些选项，如果提供了不受支持的选项，Perf 可能无法启动。--no-delay--no-buffering

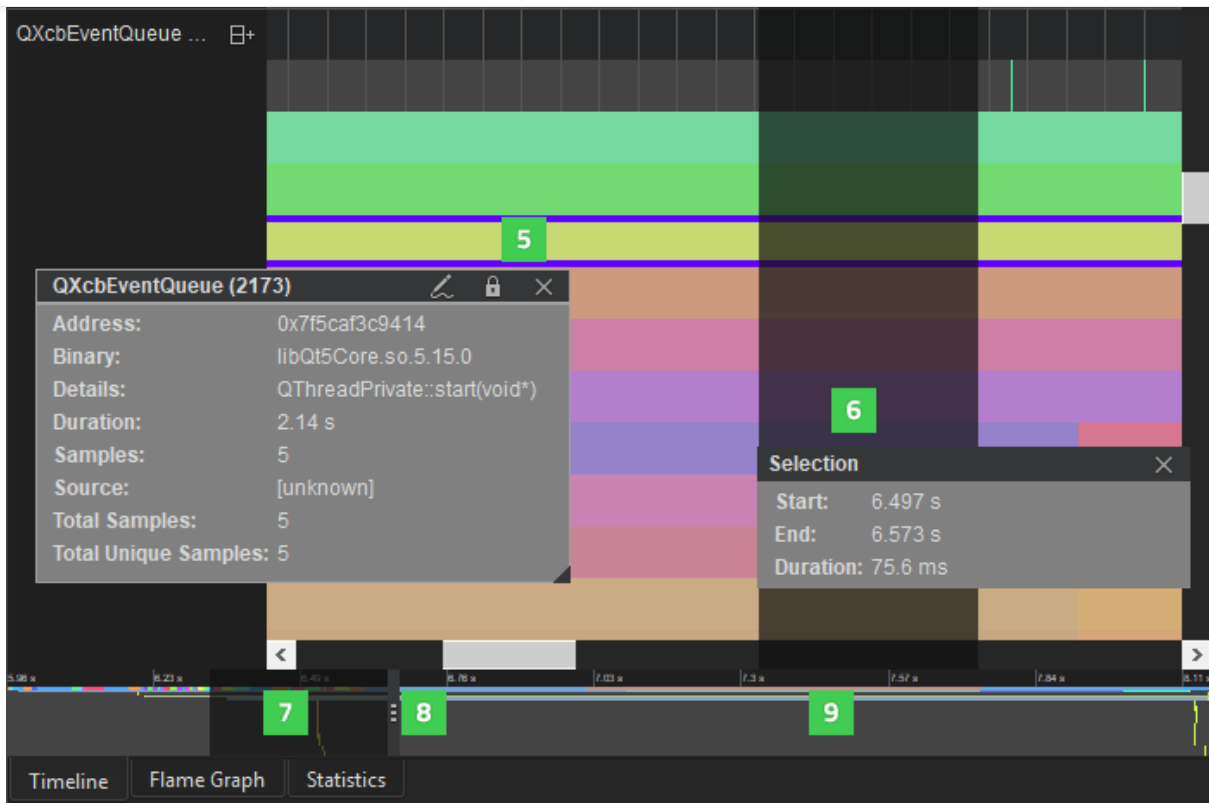
解析 JIT 编译的 JavaScript 函数的名称

从 5.6.0 版本开始，Qt 可以生成 perf.map 文件，其中包含有关 JavaScript 函数的信息。性能分析器将读取它们，并在**“时间线”**、**“统计信息”**和**“火焰图”**视图中显示函数名称。仅当正在分析的进程在主机上运行而不是在目标设备上运行时，这才有效。若要打开 perf.map 文件的生成，请将环境变量添加到**运行环境**并将其值设置为。QV4_PROFILE_WRITE_PERF_MAP1

分析收集的数据

“时间线”视图显示每个线程的 CPU 使用率的图形表示形式，以及所有记录事件的精简视图。





时间轴中的每个类别描述应用程序中的一个线程。将光标移到行上的事件（5）上，以查看它需要多长时间以及它所代表的源中的哪个函数。若要仅在选定事件时显示信息，请禁用“鼠标悬停时查看事件信息”按钮（4）。

大纲（9）总结了收集数据的时期。拖动缩放范围（7）或单击轮廓以在轮廓上移动。您还可以通过选择“跳转到上一个事件”和“跳转到下一个事件”按钮（1）在事件之间移动。

选择“显示缩放滑块”按钮（2）以打开可用于设置缩放级别的滑块。您也可以拖动缩放手柄（8）。要重置默认缩放级别，请右键单击时间轴以打开上下文菜单，然后选择重置缩放。

选择事件范围

您可以选择事件范围（6）以查看其表示的时间或放大跟踪的特定区域。选择“选择范围”按钮（3）以激活选择工具。然后在时间轴中单击以指定事件范围的开始。拖动选择手柄以定义范围的末尾。

您还可以使用事件范围来测量两个后续事件之间的延迟。在第一个事件的结束和第二个事件的开始之间放置一个范围。“持续时间”字段显示事件之间的延迟（以毫秒为单位）。

要放大事件范围，请双击它。

若要删除事件范围，请关闭“选择”对话框。

了解数据

通常，时间线视图中的事件指示函数调用所花费的时间。将鼠标移到它们上以查看详细信息。详细信息始终包括函数的地址、调用的大致持续时间、函数所在的 ELF 文件、此函数调用处于活动状态时收集的样本数、线程中遇到此函数的总次数以及至少遇到一次此函数的样本数。

对于具有可用调试信息的函数，详细信息包括源代码中的位置和函数的名称。可以单击此类事件，将代码编辑器中的光标移动到与事件关联的代码部分。

由于 Perf 工具仅提供定期示例，因此性能分析器无法确定调用函数或返回函数的确切时间。但是，您可以在每个线程的第二行中准确查看采样的时间。性能分析器假定，如果同一函数存在于多个连续样本中的调用链中的同一位置，则表示对相应函数的单个调用。当然，这是一种简化。此外，在采集的样本之间可能会调用其他函数，这些函数不会显示在配置文件数据中。但是，从统计上讲，数据可能会最突出地显示花费最多 CPU 时间的函数。

如果遇到没有调试信息的函数，堆栈的进一步展开可能会失败。对于某些以汇编语言实现的符号，展开也会失败。如果展开失败，则仅显示调用链的一部分，并且周围的功能可能似乎被中断。这并不一定意味着它们在应用程序执行期间实际上被中断了，而只是在平铺失败的堆栈中找不到它们。

在 JIT 模式下运行的 QML 引擎中的 JavaScript 函数可以展开。但是，它们的名称仅在设置时显示。由 Qt Quick Compiler 生成的编译

可视化 按钮允许您选择要在火焰图中显示的范围。

- › “**示例**”是默认可视化效果。水平条的大小表示给定函数记录的样本数。
- › 在**峰值使用**模式下，水平条的大小表示在分配的内存使用量达到**峰值**的时间点由各个函数分配的内存量。
- › 在**分配**模式下，水平条的大小表示由相应函数触发的内存分配数。
- › 在**释放**模式下，水平条的大小表示由相应函数触发的内存**释放**数。

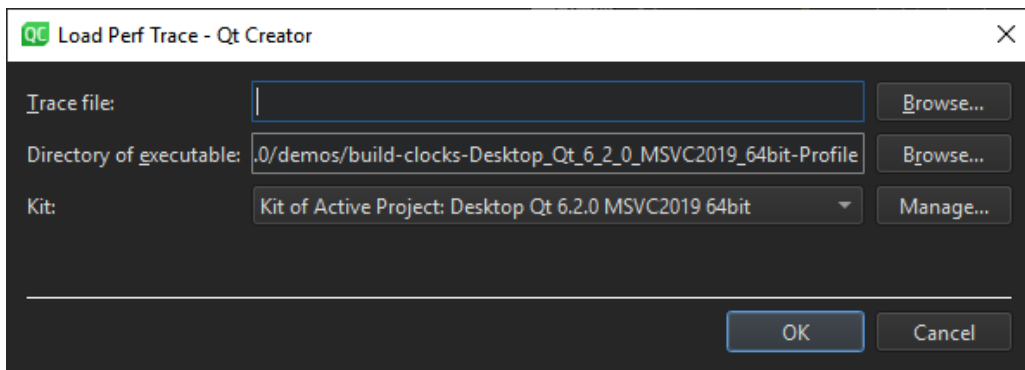
“**峰值使用**”、“**分配**”和“**释放**”模式仅在记录了来自内存跟踪点的样本时显示任何数据。

视图之间的交互

在“**时间轴**”、“**火焰图**”或“**统计**”视图中选择堆栈帧时，有关该帧的信息将显示在其他两个视图中。若要在“**统计信息**”和“**火焰图**”视图中查看时间范围，请选择“**分析>性能分析器选项>限制为“时间轴”中选择的范围**”。要显示全堆栈帧，请选择“**显示全范围**”。

加载性能数据文件

您可以加载最新版本的Linux Perf工具生成的任何文件，并在Qt Creator中查看它们。选择“**分析>性能分析器选项>加载 perf.data文件**”以加载文件。perf.data



性能分析器需要知道记录数据的上下文才能查找调试符号。因此，您必须指定用于生成应用程序的工具包以及应用程序可执行文件所在的文件夹。

Perf 数据文件是通过调用生成的。确保在记录数据时生成调用图，方法是使用选项启动 Perf。还要检查性能分析器是否在标准位置（或二进制文件旁边）或作为您正在使用的 Qt 包的一部分提供必要的调试符号。perf record--call-graph/usr/lib/debug

性能分析器可以读取在帧指针或矮模式下生成的 Perf 数据文件。但是，要正确生成文件，必须满足许多先决条件。**Boot2Qt: 支持的目标设备和开发主机**的所有系统映像都已正确设置为在矮模式下进行分析。对于其他设备，通过检查输出 oforfor 记录的 Perf 数据文件，检查 Perf 是否可以以合理的方式读回自己的数据。perf reportperf script

加载和保存跟踪文件

可以使用“**分析>性能分析器选项**”中的相应条目以特定于性能**分析器**的格式保存和加载跟踪数据。此格式是独立的，因此加载它不需要您指定录制环境。您可以将此类跟踪文件传输到另一台计算机，而无需任何工具链或调试符号，并在那里对其进行分析。

故障排除

由于以下原因，性能分析器可能无法记录数据：

- › 性能事件可能在系统上全局禁用。预配置的**Boot2Qt**映像启用了性能事件。对于自定义配置，您需要确保文件包含小于的值。为了在记录跟踪时获得最大的灵活性，您可以将值设置为。这允许任何用户记录任何类型的跟踪，即使使用原始内核跟踪点也是如此。/proc/sys/kernel/perf_event_paranoid2-1

启用 Perf 事件的方式取决于您的 Linux 发行版。在某些发行版上，您可以使用 root 权限（或等效权限）运行以下命令：

```
echo -e "kernel.perf_event_paranoid=-1\nkernel.kptr_restrict=0" | sudo tee /etc/sysctl.d/10-perf.conf
```

- › Perf 可能会永远缓冲数据，从不发送数据。添加到“**其他参数**”字段。--no-delay--no-buffering
- › 某些版本的 Perf 除非给定某个最小采样频率，否则不会开始录制。尝试使用**采样周期**值 1000。
- › 在某些设备上，特别是各种i.MX6板，硬件性能计数器功能失调，Linux内核可能会在一段时间后随机无法记录数据。Perf 可以使用不同类型的事件来触发样本。您可以通过在设备上运行来获取可用事件类型的列表，然后在设置中选择相应的事件类型。事件类型的选择会影响采样的性能和稳定性。事件是一个安全但相对缓慢的选项，因为它不使用硬件性能计数器，而是驱动软件采样。采样失败后，重新启动设备。内核可能已禁用性能计数器系统的重要部分。perf listcpu-clocksoftware
- › 可能未安装 Perf。安装它的方式取决于您的 Linux 发行版。例如，您可以尝试以下命令：
 - › 在 Ubuntu 22.04 上: sudo apt install linux-tools-\$(uname -r)
 - › 在 Debian 上: apt install linux-perf

处理数据的帮助程序的输出显示在“**常规消息**”中。

即使性能分析器显示错误消息，应用程序**输出**中也会显示某些信息。

< 使用 Heob 检测内存泄漏

使用 Cppcheck 分析代码 >

©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。



The Qt Company



联系我们

公司

关于我们

投资者

编辑部

职业

办公地点

发牌

条款和条件

开源

常见问题

支持

支持服务

专业服务

合作伙伴

训练

对于客户

支持中心

下载

Qt登录

联系我们

客户成功案例

社区

为Qt做贡献

论坛

维基

下载

市场

