

Incredibuild 加速开发
基准测试：

使用 Visual Studio 构建 CMake 项目

目录

使用 Visual Studio + CMake + Incredibuild	4
Incredibuild 助力 Cmake 项目，构建速度可提升多少倍？	5
如何进行基准测试？	6
步骤 1 – 云和基础设施	6
步骤 2 – 安装证书	6
步骤 3 – 检查构建需求	7
步骤 4 – 准备构建	7
步骤 5 – 运行构建	7
步骤 6 –使用 Incredibuild Build Monitor 分析构建结果	7

从 VS 2017 开始，Visual Studio 就与现代 Cmake 深度捆绑。因此，使用 CMake Visual Studio 操作简单，甚至在大项目中也是如此。Visual Studio 既可以生成 Cmake 项目，又能使用 C# 语言进行构建，在 Windows 世界广泛流行。

下面是选择 Visual Studio CMake 的理由：

- 无需在不同的分支中维护多个项目和解决方案，由 Cmakefiles.txt 文件自动生成。
- Visual Studio 新版本不需要迁移解决方案和项目，时刻追踪 Visual Studio 前沿技术。
- 合并不同分支更为简单，因为没有需要合并的项目或解决方案，更没有相关的更改。
- 无需维护 batch 或 Per 脚本，因 CMake 可以自动化构建。
- 基于 Cmake 构建的 CI/CD 管道比自制脚本的解决方案更易维护。



使用现代 CMake 的建议：如果你的项目仍在使用低于 CMake 2.6 的版本，请先进行升级，建议使用编译器版本之后的 CMake。

使用 Visual Studio + CMake + Incredibuild

除了 Cmake 之外，Visual Studio 还与 Incredibuild 深度捆绑，无缝衔接，加速 CMake 构建。

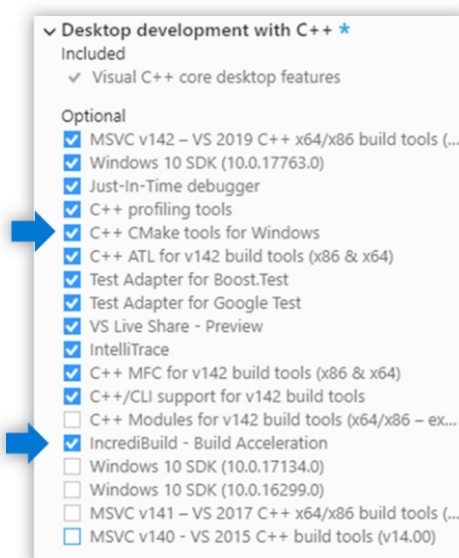
Incredibuild 的核心，Virtualized Distributed Processing™ 技术，帮助分发多个并发进程组成的工作负载，自动、动态地调用您网络或公有云上的远程计算机资源，CPU 飞速扩展。

CMake 和 C++ 的构建，通常由数百个可并行执行的编译任务组成，因此，这些成百上千的内核可帮助分担构建压力，构建时间大幅缩短。这正是分布式计算的原理。

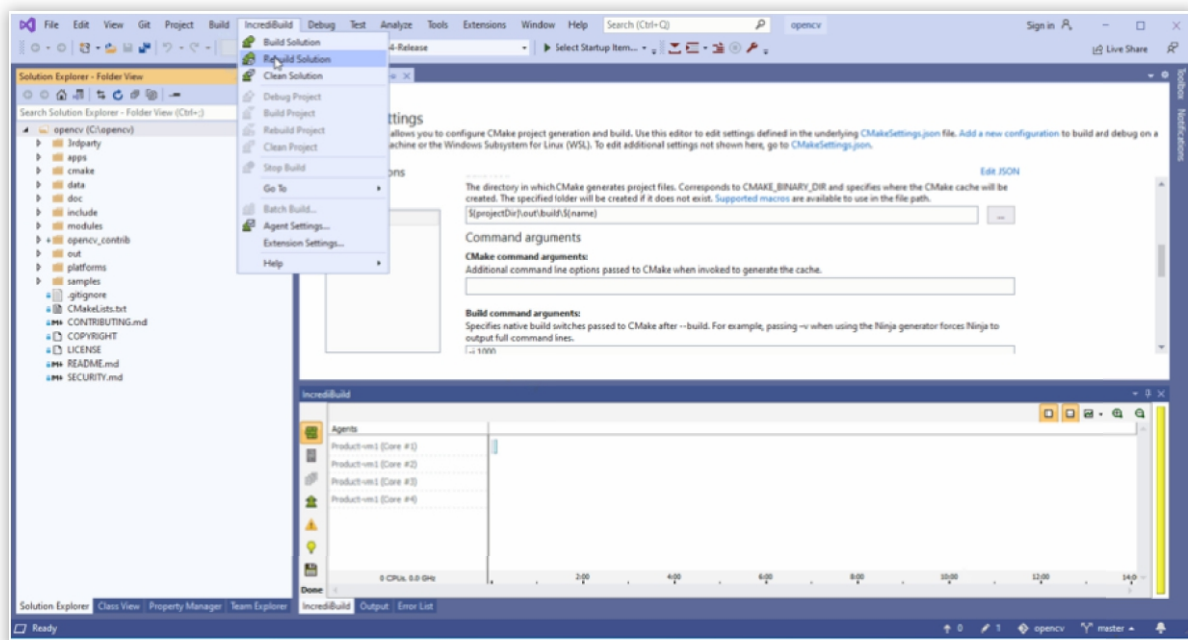
Virtualized Distributed Processing™ 甚至可以调取远程机器上的空闲 CPU，用户同时可以毫无察觉地在操作使用。一般来说，组织机构中的很多机器和计算资源都没有充分得到利用，任意时刻的空闲 CPU 总数都能破千。这些都是浪费的内核，而 Incredibuild 重新搜集这些资源，加速耗时构建。

Incredibuild 在远程计算机的安全沙箱中运行进程。进程正常运行所需的一切都 Incredibuild 在主机或远程机器中动态模拟。换句话说，**您只需要在远程计算机上安装 Incredibuild Agent，就可以实现进程分布**，而无需安装 Visual Studio、源代码或任何其他构建工具。远程执行的进程生成的任何输出（std 输出、错误、返回代码、生成文件等）都会自动同步回本地主机，就好像该进程是在本地执行一样。

在 Visual Studio 安装程序中，选择“C++ CMake 工具 Windows 版”的复选框，并选择“Incredibuild-加速构建”：



安装后，Visual Studio 开发环境中将出现一个 Incredibuild 工具栏和菜单，提供 Incredibuild 的分布式生成和重建服务。



Incredibuild 助力 Cmake 项目，构建速度可提升多少倍？

以下数据，通过编译流行的主要 OpenCV 开源项目，使用 MSBuild 运行 CMake，建立基线。案例中使用的是 Ninja 生成器，但 Incredibuild 支持所有 Visual Studio 生成器。 以下是使用 Incredibuild 后的结果：

机器数量 / 内核数

构建时间

1 台 8 核本地机器

16 分钟

4 台 22 核机器

6:26 小时

4 台 30 核机器

4:42 小时

10 台机器，总计 112 核

1:41 小时

总的来说，计算性能提升了 9.5 倍，这意味着开发人员可以花更多的时间精炼代码，避免无意义地构建时间。在这个特定的用例中，在使用的 100 个内核的基础上，向 Incredibuild 池添加更多内核资源，编译时间将更短。

重点：您可以使用相同的 Incredibuild 基础设施来加速 CI/CD 中的 Cmake 项目，以获得完整的体验，还可以加速作为构建或其他计算密集型过程（如代码分析、代码签名）一部分的单元测试，各种测试类型等等。

如何运行基准测试？

步骤 1 – 云和基础设施

我们决定使用 AWS C5 系列基础设施来运行基准测试，以兼顾性能与灵活性。另外，我们使用了纯云模型。由于基础设施的性质，我们使用 RDP 进行访问，在速度、延迟和安全性方面获得最佳效果。

步骤 2 – 安装和证书

安装 Incredibuild 非常简单，只需轻松点击几下就能实现。

在新机器上安装时，您需要安装的工具只有 Incredibuild—Incredibuild 将使用 Incredibuild 专有 VE 执行分发，该 VE 还处理工具集的分发。

我们使用了最新版本的 Incredibuild 并安装了协调器。协调器的作用是作为中介，并提供证书许可。

协调器管理请求构建编译的进程，并调用闲置机器协助运行构建。

安装了协调器之后，先加载许可证。同时，在另外 4 台机器上安装相同的版本的协调器，验证是否都使用了相同的协调器 Ip。一旦所有机器都在协调器监视器上注册，环境就准备好了。

步骤 3 – 检查构建需求

运行 CMake/OpenCV 需要使用 Visual Studio 或命令行调用。

由于 Incredibuild 同时支持这两种选择，并且我们已经使用了 Visual Studio，因此我们需要更新许可证。

步骤 4 – 准备构建

我们已经在其中一台云计算上安装了 Visual Studio，这将作为发起程序（启动生成的计算机）。

下载并安装了 OpenCV，并将其设置为通过 CMake 运行。准备开始构建。

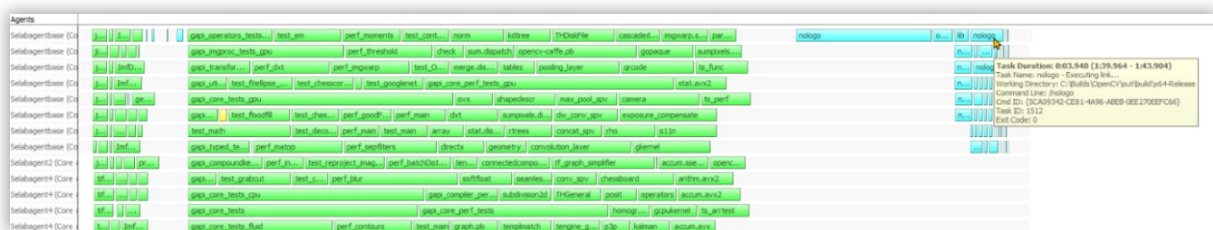
步骤 5 – 运行构建

我们已经运行了几次构建，以生成基准数据及评估性能。

首先，我们只在一台机器上运行构建。因此，我们能够得出构建在一台机器上编译的时间，分别包括使用或不使用 Incredibuild 的数据。

基准测试涉及不同数量的内核，使用所有辅助程序。

步骤 6 – 使用 Incredibuild Build Monitor 分析构建结果



分析结果表明，使用所有内核进行构建时，效果最好，时间为 1 分 42 秒。

构建监视器在构建开始时自动启动，为我们提供所用机器、当前运行的工作负载和各种构建部分的实时概览。蓝色代表连接性，帮助识别工作负载中的瓶颈。

更多内容，请查阅[网站](#)或[下载免费证书试用](#)。