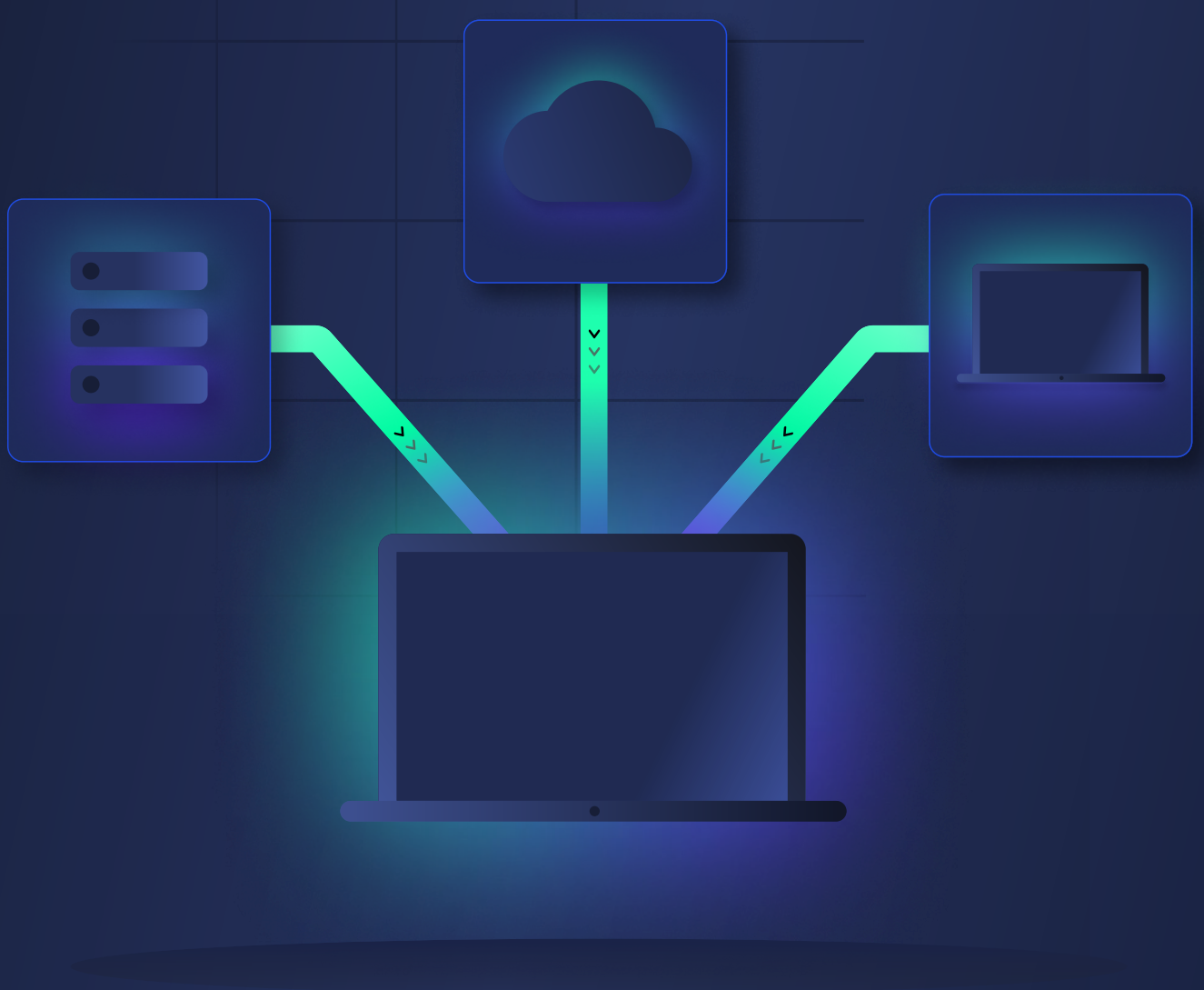
三大策略，无忧加速



INCREDIBUILD

引言

对开发者来说，处理构建问题是日常工作的一部分，就像太阳东升西落，咖啡机总在关键时刻掉链子一样无法避免。

大部分开发任务都存在构建瓶颈，构建对于开发者来说是一个头疼的问题。

在实际的开发任务中，许多场景和原因需要开发人员重新发起构建，例如源代码更改、后置处理事件等等。这正是构建环节拖长整个开发周期的原因，即使是在一个分支中的增量构建也会明显影响开发速度。

不断增加的构建时间成为开发者面临的一个普遍问题。根据 2023 年的一项调查，将近 81% 的 C++ 开发人员认为他们的日常开发中存在构建问题，其中高达 43% 的开发者认为构建时间是他们工作任务的主要痛点。

现实情况是，像C++ 和虚幻引擎这类备受欢迎的语言和工具，都存在编译、构建时间长的问题。

除此之外，越来越复杂的软件、分布式混合团队、分散的数据以及不同项目的 pipeline 汇集整合，都会增加构建时间。

因此，长时间的构建时间变得越来越普遍。

一些开发人员可能经常花费数周将不同项目的任务额汇集整合成为一个项目，为了尽可能的加快构建速度，需要花费更多的时间手动分析构建日志，以重新编写整个构建过程。

这就是我们即将探讨的问题。

无论哪个领域的开发人员，都在越来越明显的感受到长时间的构建问题越来越严重，趋于失控。

然而，也有很多开发人员却将这视为一种必然，所有开发进程都会经历这一流程，构建时间长是无法解决的。

但是长时间的构建究竟给开发团队带来什么量级的成本？从何从根本上解决这一问题？如何做到真正的将本增效？在这份白皮书中，我们将探讨我们所说的“长时间构建”的具体含义，由它带来的潜在成本有哪些？以及真正减少构建时间的三大策略。

“长时间构建”到底意味着什么?

通常情况下，构建不是不间断地进行的，有很多情况会打断正在进行的构建。

事实上，有很多因素可能会妨碍开发周期，例如不流畅且低效的工作流程以及常见的上下文切换等情况都会影响构建时间。

为了更详细地了解构建时间，我们可以把它分为三个层级：

- 较短的构建 -这些快速构建无需停下手头的工作，也不需要从一个无关的任务转到另一个任务。增量构建——在这种情况下，软件产品能够流畅地被设计定型、实现功能并完成测试，这种情况下，开发人员的思路是流畅的，不会被打断。

- 明显的构建等待 - 在这些情况下，开发人员并没有完全切换上下文，开发人员可能会在工作流程中有足够长的休息时间，可以考虑同时进行其他事情。

- 超长时间的构建等待 -对开发人员来说是最棘手的。超长时间构建可能会导致项目完全停滞下来，开发人员等待新的构建完成，以便他们可以回去看看到底发生了什么。在完全构建等待期间常见的情况主要有上下文切换、思维中断，甚至是无聊。

由长时间构建带来的隐藏成本

无论如何衡量，长时间构建都会转化为较高的成本。
一些围绕业务成本计算的统计数据相当令人震惊：

- 根据 StackOverfow 的 2023 年开发者调查，美国一名开发者的年度成本约为每年16.5万美元，包括税收等。(与上一年的开发者调查数据相比，已经增加了1.5万美元，增长了10%。)

- 在这一年中，这名开发者的成本大约是每小时83美元。

- 如果一名开发者花费 1 个小时等待构建，并且在此期间没有参与其他任务，那么公司仍然每小时平均花费 83 美元的开发者时间。这意味着公司错失了让开发者利用这 1 个小时编写更多代码的机会。

“问题肯定出在上下文切换上” 很多开发者认为这是解决问题的关键，这需要视情况而定。

虽然理论上上下文切换可能会帮助开发人员在构建等待期间完成更多工作，但事实上，研究发现上下文切换会分散开发人员的注意力，妨碍他们专注和高效的工作。

加州大学的研究发现，人们需要平均 23 分钟 15 秒的时间才能重新专注到原来的任务上。

这是在更广泛的人群和职业范围内进行考量的，由于开发工作通常需要非常深入的专注和注意力，这个数字对于开发人员可能会高得多。

花费数小时进行上下文切换，而不是在项目上工作，不仅浪费时间，而且还会使开发团队付出资源成本，包括云端资源、本地资源、网络成本等。

且这个浪费相当严重。根据开发人员需要等待构建和改变任务的次数，这可能会占用大约 3 个小时（相当于 8 小时的 38%）。

如果把重点聚焦在上下文切换方面，公司仅在这一方面每天可能会损失近 250 美元的人员成本。

等待构建的时间完全被浪费了

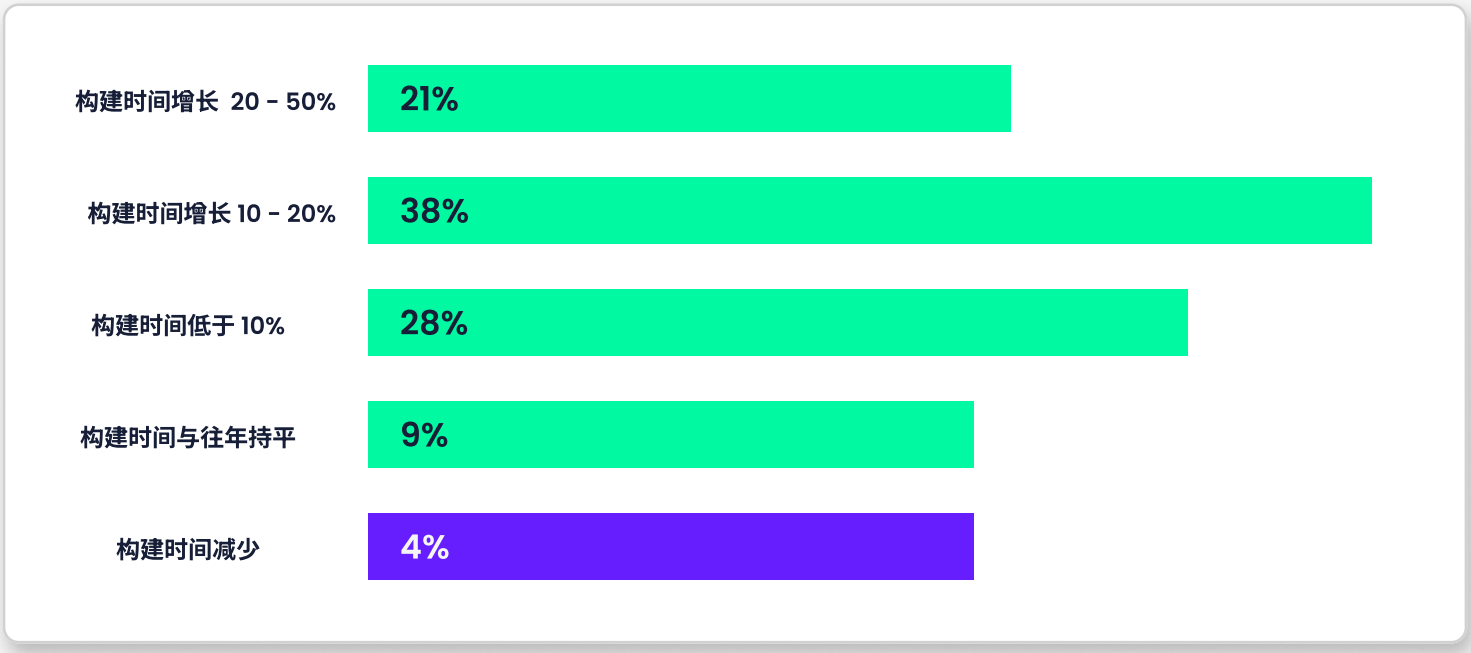
我们已经研究了长时间构建和上下文切换的经济成本，对于等待构建而浪费的开发时间，又该如何衡量呢?

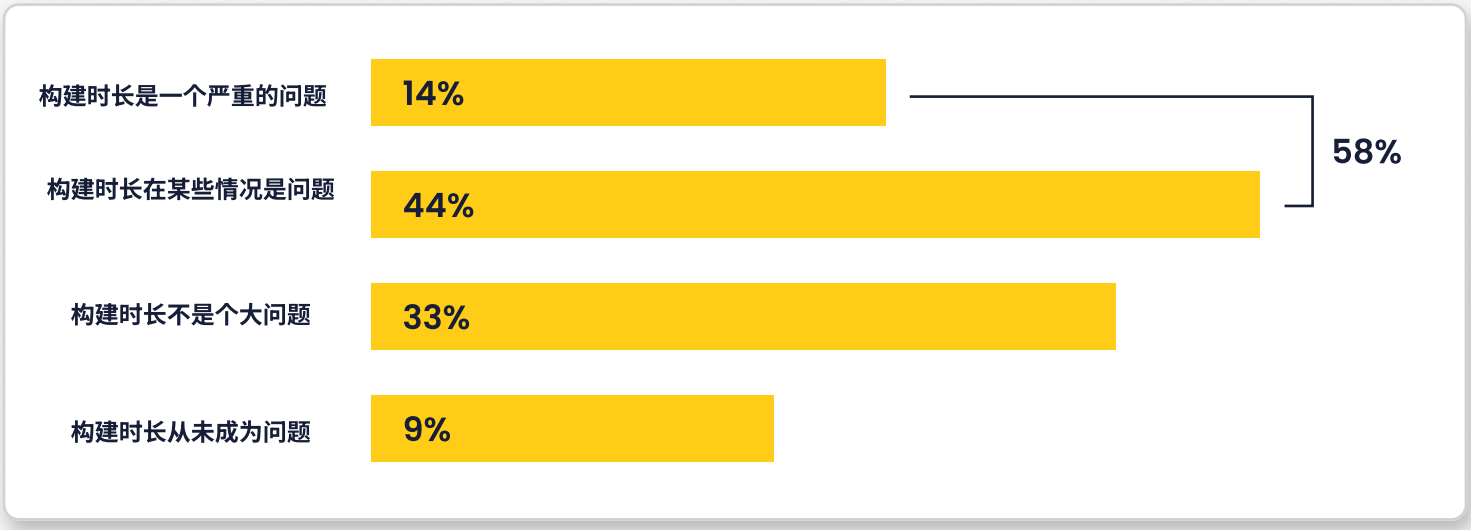
如今的构建时间很长。而且越来越长。

长时间构建不仅没有让开发人员能够快速测试、获取结果，反而减缓了开发周期，并直接影响了生产力。在许多情况下，开发人员需要等待构建完成才能继续下一项任务，因此这种等待实际上就是时间的浪费。

根据 Incredibuild 《2022 大型开发构建时间调查报告》显示:

- 尽管构建时间可能有很大的差异，但平均运行一个构建需要 20 分钟。
- 将近 25% 的受访者每次构建花费的时间超过 30 分钟，Cl/Release 构建的比例上升到 32%。
- 受访者每天平均花费 57 分钟等待构建完成。96% 的受访者表示，2021 年构建时间平均增加了15.9%。
- 69% 的公司每次运行构建花费的时间超过 10 分钟，其中 24% 的公司每次花费的时间超过 30 分钟。
- 98% 的受访者承认他们在等待构建完成时浪费了时间。
- 长时间构建对 91% 的公司来说是一个问题。只有 33% 的公司表示他们不认为这是一个巨大的问题，但 58% 的公司表示它们是一个重大问题。
- 构建等待时间也因行业而异。2022 年，国防和军事行业的平均构建时间最长，达到了 72 分钟，而银行业的平均构建时间最短，只有 38 分钟。
- 受到长时间构建影响最严重的三个行业分别是国防和军工（73%）、IT 服务（67%）和汽车（63%）。





即使银行业中最快的构建等待时间为 38 分钟，从生产力的角度来看，这也意味着大量的时间被浪费了。

长时间构建降低了快速测试、快速定位、快速失败和修复错误的能力，使产品迭代周期变得更长。长时间构建意味着有可能发生错误修正滞后，对于开发人员来说，在最终测试时才发现错误是一种灾难。

如果在发布前没有足够的时间解决这些问题，DevOps 团队和 IT 部门可能需要花费更多的经济成本来解决问题。

3大策略，无忧加速

策略一：构建缓存

缓存是将数据的多个副本存储在临时位置的过程，以便以后更快地访问。它用于软件应用程序、服务器、Web 浏览器等各种用途。

缓存的一个关键优势在于，用户和应用程序无需每次启动时都从头开始。例如，网站使用缓存来加速网页加载过程。

因此，开发人员可以用一种便捷的方式使用缓存从而加快构建时间。在软件构建中，无论是增量构建还是重新构建，一些构建输出和工件可以存储以供后续复用，这是一种强大的策略，可以在减少构建时间的同时减少工作量。

下面列出了利用缓存进行构建的优势：

提高开发人员的生产力	真正的“随时随地工作”，而不影响速度
• 在切换分支或引入新代码时无需完全重新构建 • 最小化等待时间 • 减少冗余 • 优化资源使用 • 在流程加速时，确保开发人员保持专注和动力 • 鼓励开发人员采用最佳实践 • 更加流畅和有效的软件开发工作流程	• 减少网络延迟的影响 • 使用下行带宽而不是上行带宽，使开发人员可以顺畅的在任何地方工作 • 解决远程和分布式工作环境带来的挑战， • 减少对集中式构建服务器的依赖 • 使开发人员更加灵活地应对不断变化的业务需求和市场需求

在 CI 构建之间重复使用缓存的数据	使构建更快
• 减少浪费时间 • 优化开发人员使用资源的方式 • 简化持续集成流程 • 存储和检索已下载的工件和中间构建状态，以实现更高效的构建 • 帮助公司优化基础设施成本，减少资源消耗 • 最小化对大量计算资源和存储容量的需求 • 允许组织更有效地扩展开发环境，优化资源配置 • 提供更具成本效益的资源管理方法 • 实现更高的效率和投资回报率	• 通过从构建的缓存版本中提取，显著减少构建时间 • 左移，使单次提交构建成为可能，缩短开发时间 • 开发人员的响应时间更快 • 更好地应对 deadline - 执行的构建越多，缓存的效率越高 • 更快地解决错误和缩短上市时间 • 降低成本-特别是在使用云资源进行计算时 • 节省 Cl 服务器许可证成本，并行完成更多的构建

Incredibuild 如何处理构建缓存？

Inccredibuild 开发了一种独特的构建缓存方法，基于强大且经过反复验证的并行分发技术。这项技术将低级系统钩子潜入到构建进程中，类似于反病毒软件的运作方式。通过这种方法，我们可以自动跟踪进程读取每个文件以及其他输入。这种无缝且通用的映射能力有助于识别任务的依赖关系，从而减轻开发人员的和工具面临的额外复杂性。

策略二： 编写有利于构建的更高质量的代码

无论开发团队使用的工具有多强大，减少构建时间的最简单方法就是编写有利于更快构建的优质代码。

在编写代码时，开发人员很容易专注于自己的便利性，并采取一些更快捷的方式以加快编码的速度。

（事实上，我们已在博客上介绍过如何通过 GitHub 提升整体代码质量，点击查看）

但是，这些快速编码实践不一定能与快速构建兼容，它们可能会增加依赖项，使代码更复杂，或者只是增加了编译工作量。

以下是解决此问题的三种简单方法

减少依赖关系

减少依赖关系对于有效的构建至关重要。如果将所有不同的文件、组件、模块和层联系在一起，构建将变得越来越复杂，构建时间将会变得越来越长。

相反，尽可能地简化构建才是重点所在。发现关键的依赖关系，找出可以摆脱的依赖关系，并评估哪些环节可以进行更多的解耦工作。

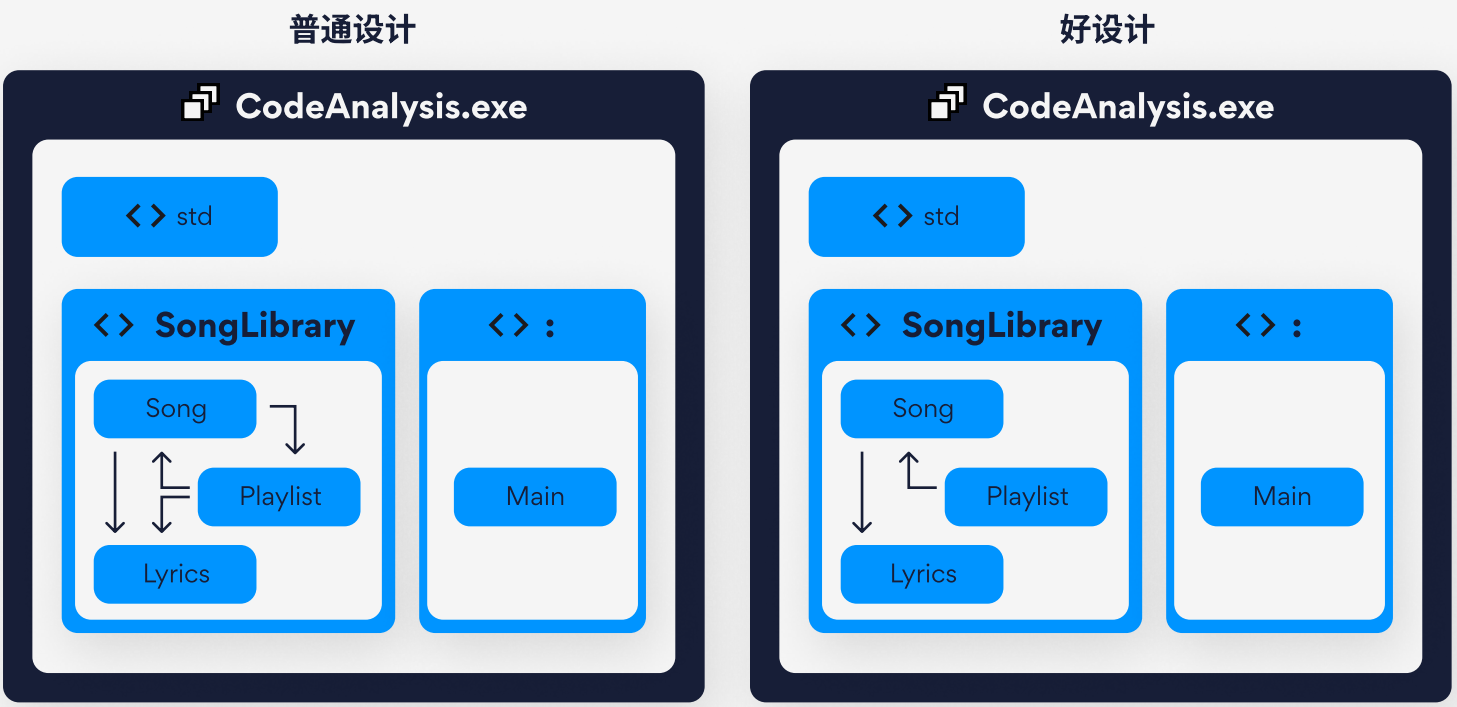
随着内部代码质量的提高，编译时间随之加快。

依赖反转原则，即 SOLID 原则，来减少构建时间。放弃不适合编译单元的头文件。例如 include-what-you-use 这样的工具非常适合从现有代码库中清除这些依赖项。

避免将代码拆分成太多小函数

将代码拆分成多个部分可能会极大地增加调用时间，或者函数调用开销。
开发人员应在手动优化、编译器选项和良好的编码实践之间找到平衡点。这不仅会提高性能，还可以避免引入更多不必要的复杂代码，也不会妨碍编译器优化代码的进程。

像 qipa 这样的编译器选项或者适用于 Mac 的编译器选项可以优化构建进程。



使用预编译头文件

最后，最推荐的一种减少 C++ 编译时间的方法——预编译头文件

预编译头文件就像数字世界中的时间节省者。这些文件以中间形式存储了轻微修改的头文件，加速了本地编译而没有任何不利影响。

预编译头文件是从 C++ 头文件生成经过解析和预处理的二进制文件，当构建进行时，预编译头文件会检查它们是否已经构建过，跳过已经完成的构建，这种方式可以将编译时间缩短高达6倍。

需要注意的是分布式构建也要谨慎使用。虽然预编译头文件可以在本地加速，但在分布式情况下，它们倾向于聚合单元，可能会拖慢进程。因此，请确保密切关注预编译头文件的更改，如果它们经常更新，效率能会降低。

升级编译器是另外一项提升构建速度的方式。现代编译器不断改进，在优化和代码生成方面变得更加高效并提升编译速度。实践证明，从旧编译器切换到新编译器 可以明显缩短 C++ 编译时间。

AI 能优化构建时间吗？

虽然减少构建时间是一个难题，但未来可能会有一些突破性的解决方案。然而，可能阻碍这些解决方案的一个主要趋势是人工智能的发展 (AI)。据估计，AI 辅助开发实际上会减慢构建时间，导致更多的瓶颈，因为它使用了更多的代码，并将其更快地移入管道中。目前，92%的美国开发人员已经在工作内外使用AI编码工具，这确实是一个值得关注的问题。

策略三：优化硬件和资源配置

构建速度更快的一个简单方法是拥有更好的硬件，或者以更智能的方式利用现有的硬件和资源。具体实践建议如下：

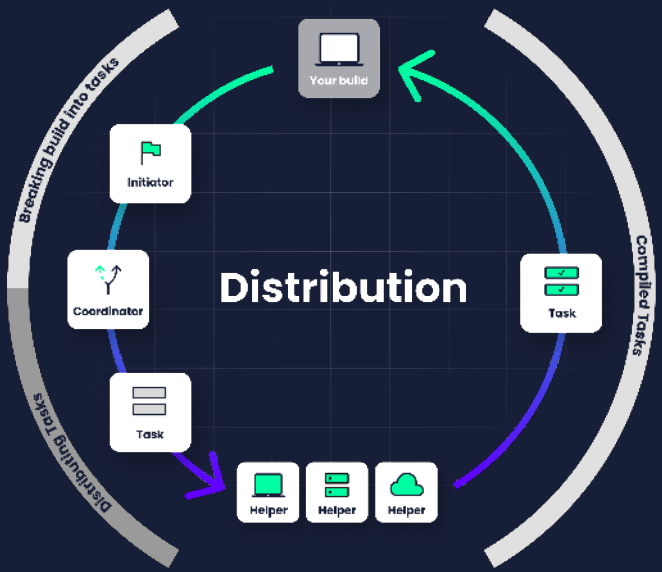
- 获得更多硬件-这是获得更快构建速度的最快方法，添加额外的内存、CPU、硬盘和服务器。但这也带来了一些问题：成本昂贵，且扩展性差。为了避免耗尽空间、资金和资源，开发团队需要找到经济效应的边际点，才能确保在不增加额外支出的基础上获得最优效益。

- 启用更多的云资源 - 与购买硬件类似，在预算允许的情况下，开发团队只需支付一定的经济成本就可以轻松获得更多的资源，需要注意的是购买云资源不止是支出经济成本，还需要花费时间和精力来管理他们，很多时候开发团队会忘记一些实例而造成资源浪费。

- 充分利用已有资源 - 与其他两种选择不同，总有全新的解决方案来重复利用现有的资源，利用分布式构建、当前构建的基础设施以及 Pipeline 最大化使用现有硬件所贡献的资源。针对云资源，可以选择更具性价比的机制和方案来管理和分配使用。

分布式构建是什么？

分布式构建将任务拆分成多个部分，同时并行运行较小的子任务和组件。这样可以减少每次构建所需的时间以及每个用户所需的资源，利用更多现有的资源，而不是要求增加更多硬件。除此之外，还可以通过云构建来实现分布式构建 ：
这是我们之前制作的一个示例



构建对开发人员来说任重道远，但充满希望

一个公认的事实是：开发出优质的软件产品需要时间，这在短期内是无法改变的。

但我们可以寻求各类方法来优化开发进程，尽可能地缩短构建时间。
通过使用本文介绍的三种关键策略：构建缓存、编写优质代码以及优化硬件和资源配置来最大化的缩减过长的构建时间，无需付出额外的费用、时间或压力。

在实际开发实践中有效实施这三种策略，营造更敏捷的开发环境，提高整体项目交付效率。

关于 Incredibuild

Incredibuild 是全球领先的开发加速平台。Incredibuild 虚拟化分布式处理和拥有专利的构建缓存技术帮助开发团队的在本地和云端更快地构建，降低开发成本，使开发人员能够更多地迭代，更快地修复错误，更短时间内发布优秀的产品。Incredibuild 还帮助开发团队充分利用当前在硬件方面的投资，优化云资源，以获取更高的投入产出效益。使用 Incredibuild 不需要更改现有的工具或流程，它是一个轻量级、无代码的解决方案，大多数团队可以在几小时内运行起来。Incredibuild 设计用于加速分布式开发团队，即使是那些在家工作且带宽较低的团队——从第一个 Sprint 开始

了解更多 Incredibuild 如何工作的信息 incredibuild.cn

INCREDIBUILD