

# 将 Android 车载系统 AOSP 构建速度提升5倍

在本文中，你将了解如何使用共享缓存和分布式计算来加速 AOSP 构建，实现：

- ✓ 更快的构建时间
- ✓ 提升 CI 吞吐量
- ✓ 降低云端成本

行业基准线与使用 Incredibuild 加速后的 Android 车载系统 14 在 32 核机器上的构建时间比



\* 在 32 核机器上，使用 Incredibuild 构建时间加速比行业基准线提升约 5 倍。

# 内容提要

Incredibuild 提供了一款专为大规模 Android 构建而设计的构建加速解决方案。该方案通过无缝集成分布式计算、通用共享缓存和全面的构建可观测性，可将 Android 构建速度提升高达 5 倍。它能够直接与现有 CI 工具集成，有效减少构建等待时间、提升主机吞吐量并优化资源效率，且无需对代码、构建脚本或工具链进行任何更改。本文档阐述了 Incredibuild 如何在本地、混合环境或多团队协作场景中，为基于 AOSP 的构建管道提供持续的性能提升和完整的可视化洞察。

## 等待的时间成本

“开发者等待时间的年度总经济成本为  
3,000,000 美元/每年

CircleCI

虽然 AOSP 的构建过程稳健，但本质非常耗时。一次完整的全量构建可能耗费数小时，并占用大量计算资源。更关键的是，由于复杂的依赖关系，即使是增量构建也常常需要重新编译代码库中的大部分内容。这种“每次改动都需从头构建”的模式给开发团队带来了若干关键痛点：

- **开发效率低下与空闲等待：**工程师需要耗费大量时间等待构建完成，这不仅打乱了他们的工作流程，也延长了反馈周期。这种非生产性的等待时间直接影响了团队士气和工作效率。
- **CI 流水线吞吐量下降：**构建队列成为主要瓶颈，尤其在开发高峰期。每台构建服务器每日只能处理有限数量的构建任务，导致等待时间延长和设备资源扩张。
- **基础设施成本攀升：**长时间的计算密集型构建需要配备高性能、持续运行的构建服务器和大量的云计算资源，从而显著推高了运营成本。

当前模式存在大量重复性工作——每位开发者和 CI 代理都可能反复编译相同的未改动源文件，这种既昂贵又低效的过程在大规模开发中难以持续。

# 解决方案

Incredibuild 平台通过两项核心技术——多工具共享缓存与分布式计算，智能地消除冗余工作并加速剩余的非冗余任务，从而有效应对上述挑战。这一方法将构建过程从一系列孤立、重复的任务转变为协同化、累积式的高效作业。

## 跨技术栈的共享缓存

Incredibuild 性能和资源优化的核心在于其通用共享缓存引擎。该引擎确保：只要已执行任务的输入数据未发生变化，同一任务就无需任何人重新执行，而是直接由共享缓存提供结果。

共享缓存会存储历史任务的输出，并在不同用户、机器和 CI 作业中复用这些结果。一旦某个任务在组织内任何地方执行过，其输出就会被缓存，后续无需重复执行，除非 Incredibuild 自动检测到其输入已发生变更。

在 AOSP 构建过程中会使用多种不同的工具和脚本。Incredibuild 共享缓存的通用性因此显得尤为重要，它能有效加速大量多样化工具的运行。下表列出了 AOSP 构建中受 Incredibuild 共享缓存加速的工具列表：

工具名称	类别
clang	C++ 编译器
clang++	C++ 编译器
clang++	C++ 链接器
rustc	Rust 编译器 & 链接器
yasm	汇编器
javac	Java 编译器
R8	AOSP java 工具
D8	AOSP java 工具
metalava	AOSP java 工具
kotlin-compiler	Kotlin 编译器 (JVM环境)
jarjar	Java 打包工具
Turbine	Java头文件编译器
versioner	AOSP专用工具
aidl	AOSP专用工具
header-abi-dumper	AOSP专用工具
hiddenapi	AOSP专用工具
appcompat.sh script	AOSP专用工具

## 分布式计算技术

与 Incredibuild 共享缓存形成互补的是其分布式计算引擎，该引擎能将普通工作站或云虚拟机转变为强大的构建节点。它允许将编译、链接或执行自定义工具等任何构建任务，实时卸载到网络中空闲的 CPU、CI 运行器或公有云上，从而有效将每台机器转变为拥有数百个核心和数十 GB 内存的高性能计算机。

对于需要额外计算能力的工作负载，Incredibuild 的编排器能够自动调配基于云的辅助机器（包括经济高效的竞价实例）实现无需人工干预的突发容量扩展。这使得企业能够在保持低成本的同时，按需弹性扩展远程构建核心资源。

# 实测结果： 安卓车载系统 14 基准测试

为评估 Incredibuild 对 AOSP 开发流水线的影响，我们对安卓车载系统14的构建进行了基准测试。测试在32核和96核构建机器上模拟真实场景，对比了标准构建方案与采用 Incredibuild 技术加速的同等构建方案。结果显示，构建性能获得显著提升，资源利用率也得到有效改善。

指标	标准 CI	使用 Incredibuild	影 响
AOSP 构建时间	1 小时 7 分钟	14 分钟	构建速度提升约 5 倍
每CI主机每日构建次数	约 10 分钟	约 5分钟	构建速度提升约3.2倍，吞吐量提升3倍
等待时间 (峰值时段)	30-60 分钟	小于7分钟	Significantly reduced developer wait times

安卓车载系统14的全量构建时间从超过一小时缩短至14分钟，效率提升达5倍，每次构建可节省53分钟！即便在高端96核机器上，Incredibuild 仍能实现近三倍的构建速度提升，同时凭借其高效的缓存机制降低总体算力消耗。这种优化极大提升了多核机器的吞吐效率，使得单台机器能够并行执行三个 AOSP 构建任务，而非仅限一个。

# 总结

Incredibuild 提供了一套加速 AOSP 构建管道的完整解决方案。通过将分布式计算、共享缓存与对现有工具链的零修改方案相结合，它可带来以下显著优势：

- **开箱即用的3-5倍构建加速：**无需配置即可实现构建速度大幅提升
- **提升 CI 吞吐量：**单台主机可执行更多构建任务，并显著减少队列等待时间
- **降低云计算成本：**通过共享缓存机制和支持使用经济高效的 Helper 竞价实例实现
- **360度构建运维可视性：**提供图形化构建监控界面，实时可视化构建执行流程，助力快速定位瓶颈与故障；详细的缓存命中/未命中跟踪工具可持续分析并优化缓存效能

Incredibuild 的共享缓存与分布式处理技术均采用通用的底层应用插桩方案，具备高度可扩展性。该技术使 Incredibuild 能够为各类编译器、链接器、构建系统及其他工具提供开箱即用的支持。对于实际 Android 构建中常见的自定义供应商工具或专有预处理/后处理步骤，仅需简单配置即可将其纳入缓存和分发逻辑，从而获得额外的性能提升。

被全球超2000家领军企业信赖

