Accelerate Android Build Times for Faster Time to Market

CloudBees®

Android unites automobiles, cell phones, appliances and a burgeoning number of everyday devices.

However, waiting hours for Android integration build cycles to complete is a massive bottleneck. This is true for device manufacturers who have to validate their entire code base including the Android Open Source Project (OSP) against a large matrix of phones, tablets and appliances before they can release a new version.

For many teams, the time it takes for build and test cycles to complete allows for only one iteration per day, which limits productivity in day-to-day feature development. It also drives up costs through manual analysis and rework and slows time to market in general. Traditional solutions for shortening build times - like more hardware or open source tools – are error-prone and are neither scalable nor affordable.

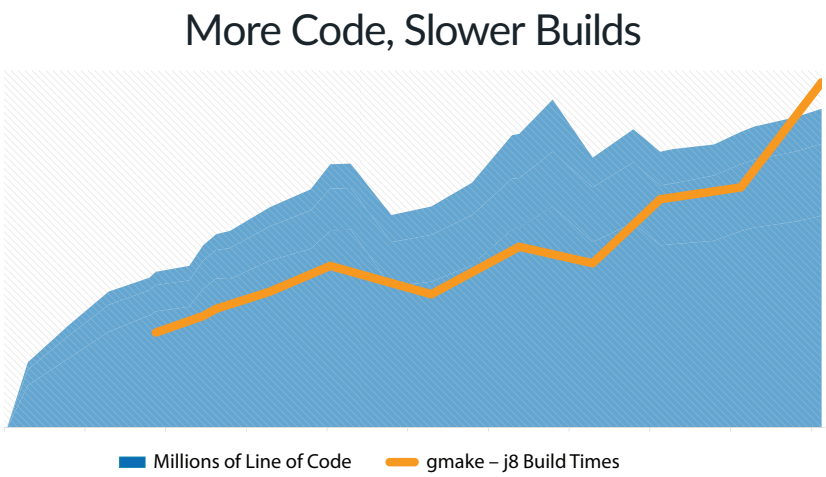
To keep pace with Android innovation and market pressures, what's needed is a new approach that:

- Safely accelerates build and test cycles.
- Leverages available infrastructure for scaling up on demand.
- Increases developer productivity and agile throughput.

This eBook discusses the common challenges for developing with Android OSP and how CloudBees Accelerator is the only cure for slow builds and long test cycles.

Android Market Pressures

Since 2015, Google has released a major update to Android every year, and has publicly committed to maintaining this yearly cadence. Android 9.0 (Android Pie) was released on August 6, 2018 and included, among other things, new functionality like digital well-being, adaptive batteries and displays, interactive content slices and intuitive navigation. Google's own Pixel devices had these new features available immediately. Meanwhile, device makers like Sony, Samsung, LG, Motorola, GM, Chrysler, Magic Leap, Acer and HP are under pressure to release updated versions of their products as soon as possible in order to retain market share — the first device manufacturer to get to market carries a big advantage as end users expect access to these features quickly.

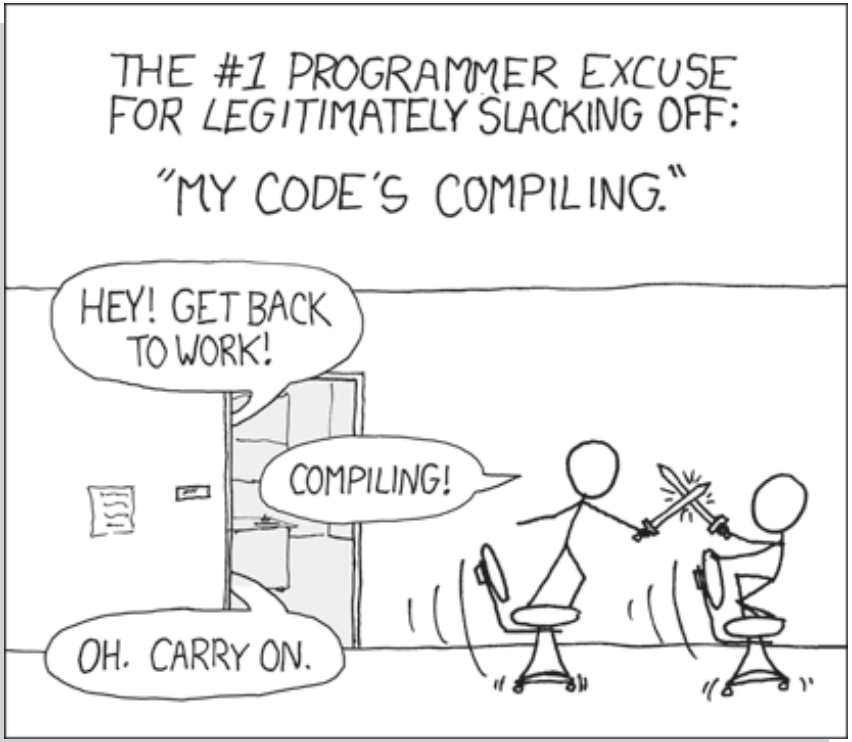


Build Times Increasing

Adding to the pressure to release new software faster, the Android operating system is huge and growing with each new release (Android Pie has 95 million lines of code - nearly a 2x increase from 2015 to 2018). Build durations have been increasing along with that growth. What used to take 30 minutes, now takes an hour or more – and that's before any customizations like kernel changes, branding and applications are added into the build. Device manufacturers regularly report AOSP, kernel and application build times running to five hours or more. If there is ever a problem with the build, as there inevitably will be, developers have to spend valuable time debugging the build itself, rather than writing new features or fixing code.

What Can Be Done?

Waiting for builds to complete is not a new problem so there has been plenty of time for a number of solutions to have been developed. The most obvious one is to provide more and faster hardware to developers so they can stay focused on their work (and stay off the ping-pong table). Coupled with the hardware is a myriad of open source tools available for managing builds, caching results and balancing workloads. Or, perhaps there's a different way to approach the problem.



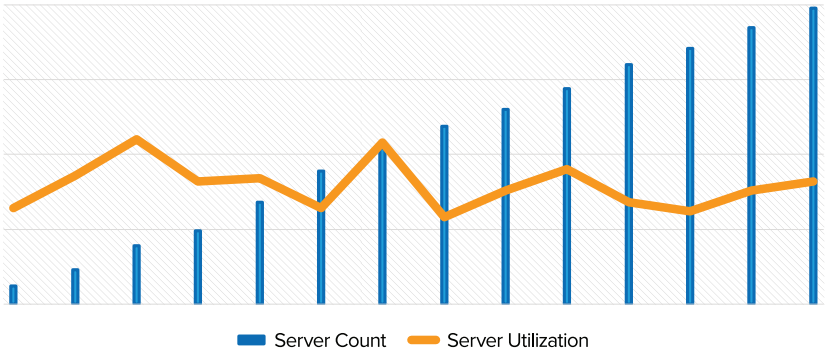
Source: XKCD

More Hardware Doesn't Solve the Problem

If developers are starved for computing resources to perform builds at acceptable speeds, then an obvious solution is to just buy them more computing power: equip all developers with their own 32- or 64-core servers or workstations.

That sounds reasonable, except that can cost in excess of $20,000 for each developer. That's unsustainable for any budget.

Build and Test Infrastructure Footprint



A second option is to buy several dedicated build servers shared by the team, then let developers work out a time-sharing plan for those systems (you know, like our grandparents used to do in the 1960s). Problem solved!

Except for a few things:

1. Build times vary wildly depending on the load on shared servers.
2. Build failures caused by missing dependencies frequently occur when using multi-threading on larger machines.
3. Surprisingly, the more hardware that is available, the lower the total utilization.

Challenges of Using Open Source Tools

The proliferation of open source tools means there is probably a tool somewhere that will help with any software development issue, and build acceleration is no exception. In addition to GNU Make, Ninja, ccache and distcc are just a few open source tools available to execute and accelerate builds. They are readily available, have large followings and are great for individual developers to manage simple builds.

ccache

"...we removed ccache from our builders (and master), since it was causing more problems than it was saving in time."

(Source: Comment excerpt from the Android-building mailing list, October, 9, 2018)

However, there are problems with performance, reliability, resource utilization and visibility:

1. Many tools are complicated and require deep expertise to be configured for optimal performance.
2. They offer no assistance in managing dependencies, parallelization and load balancing in large builds, leading to unreliability at scale.
3. There are no native diagnostic tools for fixing broken builds, further wasting valuable developer time.

An Alternative for Build Acceleration: CloudBees Accelerator

A better way to accelerate builds would be to give developers:

Blazing Speed

A platform that intelligently distributes workloads across available infrastructure to lower build times to minutes, instead of hours.

Rock-Solid Reliability

A platform that eliminates the waste of broken builds by automatically optimizing builds and their dependencies.

Hands-Free Scalability

A platform that automatically manages workloads for multiple teams, data centers and clouds.

Built-in Optimization

A platform that intelligently reuses unchanged artifacts and automatically knows to use the best available resources.

Deep Diagnostics

Analytics that enable developers and resource managers optimize, debug and build infrastructure at scale.

The Better Way

CloudBees Accelerator is the acceleration platform that intelligently and automatically parallelizes software tasks across clusters of physical or cloud CPUs to dramatically lower build times. This gives Android device manufacturers the ability to speed up any number of concurrent builds so they can deliver better software, get more productivity from developers and quickly respond to market requirements.

CloudBees Accelerator Benefits:

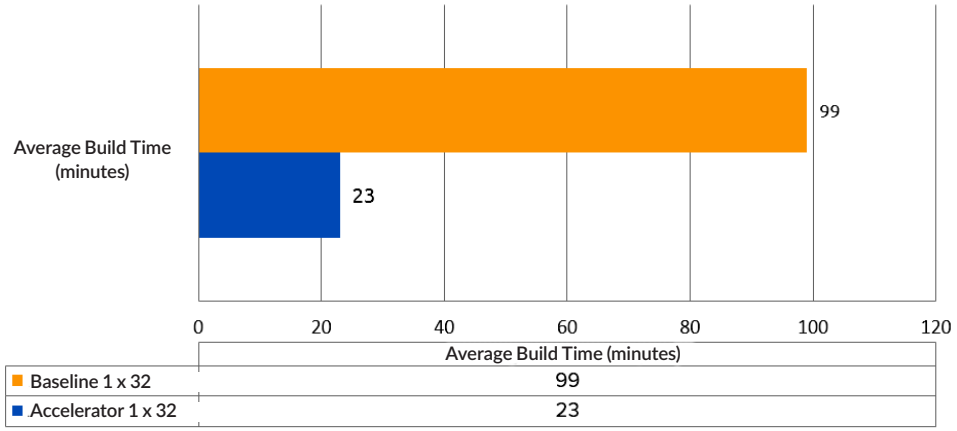
- **Unprecedented build acceleration:** Up to 20x build speedups.
- **Unmatched reliability:** Guaranteed builds eliminate costly rework and debugging time.
- **Unlimited scalability:** Intelligently manages build clouds across multiple teams, data centers and clouds.
- **Resource optimization:** Eliminates rework, reduces dedicated build hardware as much as 75%, greatly increases existing infrastructure utilization and easily bursts to the cloud.
- **Unparalleled insight:** Visibility into single-build performance and resource utilization to pinpoint problems and identify bottlenecks in the organization.

Proof Point: Zebra

Zebra manufactures Android-based devices for barcode printing, mobile computing, data capture and location services. According to their website:

> *"Zebra empowers those on the front line in retail, healthcare, transportation and logistics, manufacturing and other industries to achieve a performance edge – an edge that translates to delighted customers, good patient outcomes and superior business results."*

CloudBees Accelerator shrinks build time by 76 minutes
Increases build throughput by 4x



Challenges

- Concurrent access to saturated build infrastructure severely affected throughput and cycle times.
- Developers wait for up to five hours for Android builds to complete, reducing the number of iterations and slowing time to market.
- Android builds took 99 minutes in the best case scenario.
- Increased build times drove developers to constantly demand more cores and dedicated nodes, resulting in spiraling infrastructure costs.

Results with CloudBees Accelerator

- 4x build throughput increase on identical infrastructure setup.
- Build time variance dropped from two to as much as five hours, to 10-23 minutes.
- Average build times reduced by 80%, from 99 minutes to 23 minutes.

Summary

The stakes for Android device manufacturers are high. Time to market, developer productivity and resource optimization are key to success in such a competitive space. Agile development isn't agile and continuous delivery isn't continuous if builds are time-consuming. CloudBees Accelerator is the only solution for predictably lowering build times by 20x or more – and on 75% less hardware.

About

CloudBees is the industry's leading DevOps technology platform delivering the world's first end-to-end continuous software delivery management system. CloudBees enables developers to focus on what they do best: Build stuff that matters—while providing peace of mind to management with powerful risk mitigation, compliance and governance tools.

Used by 50% of the Fortune 500, CloudBees is helping thousands of companies harness the power of continuous everything and gets them on the fastest path from great idea, to great software, to amazing customer experiences, to being a business that changes lives.

Visit CloudBees at www.cloudbees.com.

CloudBees Core is built on top of Jenkins, an independent community project. Read more about Jenkins at: www.cloudbees.com/jenkins/about

0120v01

CloudBees, Inc.
4 North Second Street | Suite 1270
San Jose, CA 95113
United States
www.cloudbees.com
info@cloudbees.com