INCREDIBUILD

Epic Games

Distributed Computing and Game Development: How Epic Games uses the power of many to turbo-charge the build process

Industry

| Game Development

Process

| C++

Results

| Reduced UDK 32-bit Windows build from 3 mins to 1.6 mins

Epic Games' Unreal® Engine is the technological driving force behind many of today's leading video games. This article explains how Epic integrated Incredibuild's distributed computing solution in Unreal Engine to great success, benefiting the company and others who've licensed the technology.

The Challenge

Unreal Engine contains more than two million lines of code, all of which can be compiled in various configurations for numerous supported platforms. Programmer time lost to lengthy compile times adds up fast, as does developer time lost to broken code. The challenge is to improve iteration time for programmers and automated test systems verifying code.

Consider that a 32-bit Windows build of the Unreal Development Kit (UDK) in release mode takes just under three minutes to compile on a blazing-fast, ultra-high-end 8-core/16-thread machine. That might not sound so bad, but when one factors in the vast range of platforms and configurations it has to serve, not to mention the fact that countless programmers around the world are working on Unreal Engine 3-licensed games at any given time, even minor increases in build time can have a multiplier effect, resulting in undue strain on manpower and budgets.

Without Incredibuild, Epic's only option to meaningfully speed up compile times was to increase the number of cores in the systems on which code was compiled. If some files took too long to compile – because of a lack of available cores, for example -classic serialization issues would arise and productivity would suffer. This alternative ultimately didn't allow Epic to speed up build processes nearly as much as they can with Incredibuild, which effectively provides access to hundreds of cores across their entire network of machines.

The Solution

Epic needed a solution that would not only speed up local compilations for individual platforms and configurations, but also parallelize the process of building across multiple platforms and configurations simultaneously, leveraging the shared nature of the Unreal Engine's multi-platform code base. It was important that the solution benefit all Unreal Engine 3 licensees as well. Incredibuild rose to the challenge. With Incredibuild, rather than relying on additional cores running on individual units, Epic instantly gained access to hundreds of cores working across its entire network of machines.

The Incredibuild integration was smooth, and the impact was immediate.

Today Epic uses Incredibuild directly from its build system, UnrealBuildTool (UBT), for all of the platforms that compile on the PC. Developers begin by creating a dependency graph in UBT and generating the Incredibuild control file with the command lines needed to compile the build. Incredibuild takes over from there, using that XML file to distribute compiling across hundreds of systems throughout the studio.

Benefits

Incredibuild has had a direct impact on the amount of time programmers spend waiting for builds, enabling them to massively parallelize compiling

across more than 200 multi-core systems across the studio.

With Incredibuild, the aforementioned 32-bit Windows build time of UDK in release mode is reduced to 1:37 minutes. The real benefit? Epic can simultaneously compile games on all platforms and configurations and complete that process in record time due to the near-perfect parallelization possible with Incredibuild.

Epic's UBT also incorporates intelligent source-merging technology, which can dramatically improve build times when performing full rebuilds of the engine. However, for the best compile performance during rapid iteration on a single source file, it may actually be beneficial to disable this feature temporarily. In either case, Incredibuild accelerates the build process substantially.

Summary

Using Incredibuild, Epic is able to accelerate its build process dramatically, speed up testing and allow developers to focus on development instead of waiting around for builds to compile. When taking into account the large number of UE licensees across the globe, this approach literally saves man-years – not just days or hours – over the course of a project, simply by implementing Incredibuild directly in the engine.

UDK 32-bit Windows build

Without Incredibuild: 3 min

With Incredibuild: 1.6 min

---

incredibuild.com