INCREDIBUILD

inVRsion

inVRsion optimizes Unreal Engine compilation speed and AWS costs

Industry
| Game Development

Process
| C++

Results
| Compilation time reduced from 4 hours to hours

INVRSION

inVRsion is an Italian-based startup, providing virtual reality SaaS for the retail industry. The company's proprietary SaaS-based technology is used by retailers in the consumer-packaged-goods (CPG) sector to simulate retail spaces, showrooms, products, and shopping experiences. These virtual reality services support B2B activities such as trade marketing negotiations, shopper research, and training, as well as B2C activities like virtual reality e-commerce. inVRsion's SaaS-based solution ShelfZone® is used by a number of leading global retail companies and trusted by players like Accenture, Nestlé, Diesel, and PepsiCo.

The Challenge

"Our SaaS uses a huge amount of automation and scripting to get all the layout and product-placement data for the virtual stores. Customers had to wait a huge amount of time for the script to finish their job," says Michele Antolini (PhD), CTO at inVRsion.

These scripts automate the conversion of the customer's store design into a full-fledged Unreal Engine project. The automated, data-driven creation of such projects is the part that requires very heavy AWS compute resources (in particular, the compilation of the C++ code, the shader compilation, and lights baking) nd are the parts of the automation that are compute-intensive and heavily parallelized among multiple cores.

Since inVRsion's automated process relies on vast automation built on top of Unreal Editor, GPU-powered machines are needed. Without a way to distribute among multiple machines, the high number of CPUs to handle the workload, inVRsion chose using the hefty, GPU-powered g3.8xlarge machine on an on-demand basis.

Lengthy reservation time – inVRsion script starts with trying to reserve a g3.8xlarge. If no instances of such type are available, a smaller machine (g3.4xlarge) is reserved. Since the 32-core machines were scarce in the region, this option sometimes caused customers to wait longer for the process to finish with half of the computational resources available. "Normally g3.8xlarge machines would be available again within 2-3 hours. During the COVID-19 lockdown, it has taken as much as 3 days to allocate a single EC2 instance of this type. That means that a customer would have to wait for a consistent amount of time just for the service to begin processing at a reasonable speed," said Antolini.

Sluggish execution – Executing all these C++ and Unreal Engine workloads on a single instance maxed out the memory and CPU usage, resulting in as much as a 146-minutes long execution per simulation. On a g3.4xlarge (when a downgrade is needed due to instance availability), the average execution time ramps up to 194 minutes (+30%). This impacted both cost and revenue since it limits the number of iterations a user can run on a working day.

High AWS cost – Given the long execution and the expensive EC2 type, each simulation had a high price tag.

The Solution

First things first: Eliminating the bottleneck

The inVRsion team was eager to identify a way to get the bottleneck out of their way, namely the g3.8xlarge machine, which was too expensive—sometimes unavailable—and still too slow for the workloads in hand.

Since Unreal Engine's GPU requirements only apply to the process initiation rather than the actual compilation, it was necessary to break up the architecture from one-machine-does-it-all into several machines, where the GPU is used to kick the process off, but the actual execution runs on a GPU-less machine.

How do you initiate a process on one machine and execute on another?

Or—better yet—distribute the processing to several machines in parallel?

The team, who had used Incredibuild to accelerate code builds and CI pipelines since 2014, had decided to use the same technology to circumvent their GPU bottleneck.

Over one weekend of installation and experimenting with various EC2 setups, Incredibuild was ready to distribute both the simulation's C++ code builds—as well as the Unreal Engine shader compilation—off the g3 machine and onto other "helper" machines.

Now, that the GPU was only needed to fire off the process, the "initiating" machine was downgraded to g3.4xlarge, which had no availability issues and was 50% cheaper than the previous machine type.

The bottleneck has been eliminated successfully.

Incredibuild was crucial for the parallelization of code and shader compilation. Adding an extra machine to help with the lights baking process (SWARM Agent for Lightmass CPU processing) was necessary in order to compensate for the smaller number of CPUs available in g3.4xlarge instance. The advantage would have been eaten up otherwise by a slower lights-backing process.

Monolithic job – out, parallelization – in

Next, the team had built the helper machines, to be used for the workload distribution. They figured that using multiple c5.xl machines simultaneously would offer better availability and cost-effectiveness than one large supercomputer. Plus, all extra instances are automatically spun up and down on-demand, thus eliminating unnecessary costs.

Incredibuild's native integration with Unreal Engine and the C++ build tools for seamlessly distributing compute processes to multiple machines in a parallel manner was the next step. Given that the cheaper machines Incredibuild could employ had double the number of CPUs compared to the initial setup, the inVRsion team had finally reached the results they were looking for both in terms of the near-zero reservation time, processing speed, and cost reduction.

Preparing to scale up without linear cost increase

Incredibuild makes it much simpler to scale up without increasing IT efforts, using its native integration to AWS from an IT management point of view. "Once Incredibuild is installed, you won't even notice that it's working in the background. It is helping you in a transparent manner, there is really no need to manage it," says Antolini

Using Incredibuild's automatic EC2 spin up/down mechanism and ability to share instances across projects ensures that no CPU power will go to waste. Furthermore, the usage of spot instances with Incredibuild in an automated manner will further decrease costs and improve ROI.

The Results

After performing the tests and ensuring that all of the above mentioned issues were successfully addressed, the new setup—powered by Incredibuild—moved to production and is now operating autonomously as part of the company's SaaS infrastructure.

And the results are impressive:

Cost per simulation – before Incredibuild: € 9,72. After Incredibuild: € 5,05. Improvement: 48% cost reduction.

Compilation Time – before Incredibuild: 3:40 hours. After Incredibuild: 2:04 hours. Improvement: 43% faster execution.

EC2 Reservation time – before Incredibuild: 2-3 hours. After Incredibuild: Immediate. Improvement: 100% wait time elimination.

Incredibuild is an AWS Advanced Technology Partner

Incredibuild has achieved AWS Advanced Technology Partner status with Amazon Web Services (AWS) through its AWS Partner Network (APN) program. AWS has recognized Incredibuild for its seamless integration, workload processing acceleration, and cost optimization for companies using Incredibuild and AWS.

"Once Incredibuild is installed, you won't even notice that it's working in the background. It is helping you in a transparent manner; there is really no need to manage it."

"During the COVID-19 lockdown, it has taken as much as 3 days to allocate a single EC2 instance of that type. This means that a customer would have to wait an immense amount of time just for the service to begin processing."

Michele Antolini
CTO

The Bottom Line

After performing the tests, and ensuring that all of the abovementioned issues have been successfully addressed, the new setup, powered by Incredibuild has moved to production and is now operating autonomously as part of the company's SaaS infrastructure.

	Without Incredibuild	With Incredibuild
Compilation Time	4 hours	2 hours
EC2 Reservation time	3 hours	0