

Unreal Engine's New Chaos Physics System Screams With In-Depth Intel CPU Optimizations

Introduction

A “Hollywood quality” physics simulation and destruction system was unveiled at the 2019 Game Developers Conference (GDC). In its State of Unreal* presentation, Epic Games* premiered the Chaos physics system, developed with a big helping hand from Intel engineers. Based on the virtual reality game *Robo Recall**, the Chaos demo showcased the system's new fracturing tools, dynamic strain evaluation, and improved physics parameters. The action-packed short was a huge hit.



Figure 1. Showcasing realistic destruction, the Chaos physics system met an enthusiastic reception at GDC 2019. (Image source: [State of Unreal](#))

“We worked very closely with Epic’s physics team to make this happen,” said Jeff Rous, senior developer relations engineer on the Intel Graphics and Multicore Engineering team.

Kim Libreri, chief technology officer at Epic, acknowledged Intel’s “massive help” in optimizing the code. “Real-time rendering,” he declared, “is getting pretty close to being able to recreate the look of the real world.” Regarding “interactions with objects in the virtual world,” he said, it is time to do more than “simple rigid body collisions, rag dolls, and occasionally some cloth.” The new physics system dramatically advances special effects and physical simulations, adding fresh dimensions to gameplay.

Revel in New Levels of Chaos

“The next generation of physics will be an absolute game-changer,” Libreri said of the new Chaos physics system. “We’ve assembled the world’s greatest physics experts from games, movies, and even academia, and we aim to blow your socks off.”

The following features are to appear in the early-release version of Unreal Engine* 4.23:

- In-engine fracturing tools
- Dynamic strain evaluation
- Interactive simulation caching
- Scalability using single setup
- Physics parameters fields
- [Niagara](#) integration
- Next-generation collisions
- Materials support

Libreri added that the team wants to allow users to harness the full power of their workstations. Technical artists should be able to create “super-high-density simulations” that can be played back at run-time on modest hardware.

A Closer Look at Intel Optimization

Libreri invited GDC attendees to join a subsequent technical session with “our friends from Intel that helped us with the optimization and [made] sure that we were getting everything out of their processors.” Jeff Rous was part of that talk, which covered the internals of the new system and how Intel helped with the multicore-friendly code. Hundreds of developers played the demo at the Epic GDC booth, gleefully blasting concrete columns into gigantic piles of realistic rubble.



Figure 2. The new Chaos physics engine makes smoke, dust, and rubble appear much more realistic. (Image source: [State of Unreal](#))

Rous revealed that multiple aspects of the demo benefited from Intel optimization; specifically, code changes in low-level solvers, data structures, and thread parallelism.

INTEL ISPC Is the Key

Rous put in long hours alongside Epic engineers to hit the GDC deadline, but said the real hero in this story is the [Intel® Implicit SPMD Program Compiler \(Intel® INTEL ISPC\)](#) technology that takes full advantage of multiple cores. INTEL ISPC, he explained, is a compiler for a variant of the C programming language, with extensions for ‘single instruction, multiple data’ (SIMD) programming. SIMD enables developers to increase processor throughput by performing multiple computations with a single instruction.

Rous pointed out that CPUs, since the early 2000s, have had vector units that programmers use to speed up their code. “SIMD gives you the ability to do eight floating point adds in one instruction,” he explained. “One of the problems with C++ is that it’s hard to get the compiler to vectorize anything. The usual solution is compiler intrinsic [built-in functions] that you have to update with new instruction sets.”

That’s where INTEL ISPC comes in. “INTEL ISPC looks like a shader language that runs on the CPU, and gives you implicit parallelism,” Rous continued. Like a shader language, INTEL ISPC assumes SIMD by default and has an execution model similar to GPU shaders.

A big benefit is that programmers can write INTEL ISPC once and compile to many vectorized instruction sets, such as SSE4, AVX, and AVX2. “If you’re compiling for multiple instruction sets, an auto-detect mechanism chooses the right function, based on the user’s CPU,” Rous said. “No more having to update your intrinsics when a new instruction set lands.” Instead, developers can just recompile.

The goal is to make INTEL ISPC usage as seamless as possible. When the Chaos physics system ships later this year in Unreal Engine 4.23, developers will be able to make use of INTEL ISPC in their own games. The workflow is simple: include the INTEL ISPC module in the build, then add INTEL ISPC files to the project. The INTEL ISPC inclusion generates C++ headers as part of the build process. The Unreal build tool handles the rest, including compiling for several instruction sets and linking those object files into the final executable.

Significant Speedups Ahead

INTEL ISPC frequently provides significant speedup on CPUs with 4-wide vector SSE units. CPUs with 8-wide AVX vector units yield even better results, without the challenge of writing intrinsics code. Parallelization across multiple cores is also supported by INTEL ISPC. This makes it possible to write code whose performance improves according to the number of cores and the vector unit size.

Using INTEL ISPC, a program appears to be a regular serial program, but the execution model results in program instances executing in parallel on the hardware. (See the [INTEL ISPC documentation](#) for more details and examples that illustrate this concept.)

Performance-critical areas in the Chaos physics system benefit greatly from INTEL ISPC. For example, when doing physics intersection testing, multiple interactions can be tested simultaneously, using SIMD unlocked by INTEL ISPC. Another big benefit is that programmers can write INTEL ISPC once and compile to many vectorized instruction sets, such as SSE4, AVX, and AVX2. If you’re compiling for multiple instruction sets, an auto-detect mechanism chooses the right function based on your CPU. No more having to update your intrinsics when a new instruction set lands; just recompile!



Figure 3. The Chaos physics system delivers an interactive environment that is crucial for deeper immersion. (Image source: [State of Unreal](#))

Enormous Opportunities for Developers

Unreal Engine is among the world's premier AAA game engines, powering many of the most popular titles. The next release of Unreal Engine 4 will allow all developers to use INTEL ISPC in their games. Get yourself ready now: browse the links below, study the documentation, and take full advantage of this crucial technology in your next project.

Resources

Intel SPMD Program Compiler [download](#)

Junkins, Stephen: [What Can We Learn from the Intel SPMD Program Compiler?](#)

Kennedy, Jon: [Use the Intel SPMD Program Compiler for CPU Vectorization in Games](#)

Unreal Engine: [Chaos High-Performance Physics and Destruction System Full-length Demo](#)

Unreal Engine: [State of Unreal](#) (56:15 for Chaos)