

Stress testing with 20000 Entities: AEROPVS010

In vsTASKER, the "Stress testing with 20,000 Entities" project demonstrates how the simulation engine handles massive-scale scenarios while maintaining real-time performance.

The core challenge of simulating 20,000 entities—such as a large drone swarm, civilian traffic, or battlefield units—is avoiding the exponential computing slowdown caused by thousands of sensors interacting simultaneously.

Key Technical Achievements

- **Real-Time Performance:** The engine successfully detects and tracks **20,000 entities while maintaining a steady 30Hz frame rate.**
- **Combinatorial Optimization:** It prevents the simulation from crashing or lagging due to "combinatorial explosion" (where every entity checks its distance from every other entity).
- **Massive Sensor Detection:** The test showcases high-density data processing across multiple simultaneous radar and sensor inputs.

Optimization Techniques Used

To run 20,000 entities without system failure, Virtualsim framework utilizes specific optimization strategies:

- **Proximity Meshing:** Divides the simulation space into a localized grid mesh. Entities only calculate interactions with nearby entities instead of the entire map.
- **Sensor Filtering:** Discards irrelevant data early in the pipeline before it consumes heavy CPU processing cycles.
- **Efficient Code Generation:** Optimizes runtime code execution via its C++ architecture to allow faster-than-real-time capabilities.

Are you looking to **replicate this stress test**, or do you need help **optimizing a specific scenario** of your own in vsTASKER?

The "**Stress testing with 20000 Entities**" experiment is a benchmark showcase designed by **VirtualSim** to demonstrate the high-performance capabilities of its simulation engine, vsTASKER.

The primary goal of the stress test is to prove that the software can handle massive sensor detection grids without suffering from architectural bottlenecks or dramatic drops in execution speed.

Core Technical Details

- **Objective:** Simultaneously tracking and detecting **20,000 distinct entities** in a single scenario.
- **Performance Benchmark:** Maintaining a strict, stable **30Hz real-time frame rate** (30 ticks/updates per second) throughout the execution.
- **Primary Mechanics:** The scenario utilizes **multiple simulated sensors (such as radars)** operating concurrently.
- **How vsTASKER Achieves This Scale**

Simulating 20,000 entities interacting with sensors usually causes a "combinatorial explosion" because every sensor tries to calculate its distance and line-of-sight to every single object ($O(N^2)$ complexity). vsTASKER bypasses this performance wall using specific architecture:

Optimization Filters: It applies native spatial partitioning and proximity meshes to discard far-away entities immediately. This means the radar sensors only run heavy math on entities within their actual field of view and range.

C++ Code Generation: Unlike interpreted simulation engines, vsTASKER automatically generates optimized C++ code. The scenario compiles directly into native binaries, allowing it to leverage full hardware processing speeds.

Component Integration: The simulation logic can be linked with 3D visualization tools like OpenSceneGraph (OSG) or tactical displays while running the heavy background computations seamlessly.

Practical Applications

This benchmark is used to validate vsTASKER for large-scale military and civil defense simulations, including:

Massive drone swarms and dynamic shielding.

- **Air Traffic Control (ATC)** tracking global, live flight streams (e.g., streaming ADS-B data).

- **Electronic warfare** environments involving dense radar cross-sections.
-

Are you looking to replicate this specific **stress test setup**, or are you designing a custom **large-scale simulation** and want to know how to optimize its performance?

Stress testing with 20000 Entities

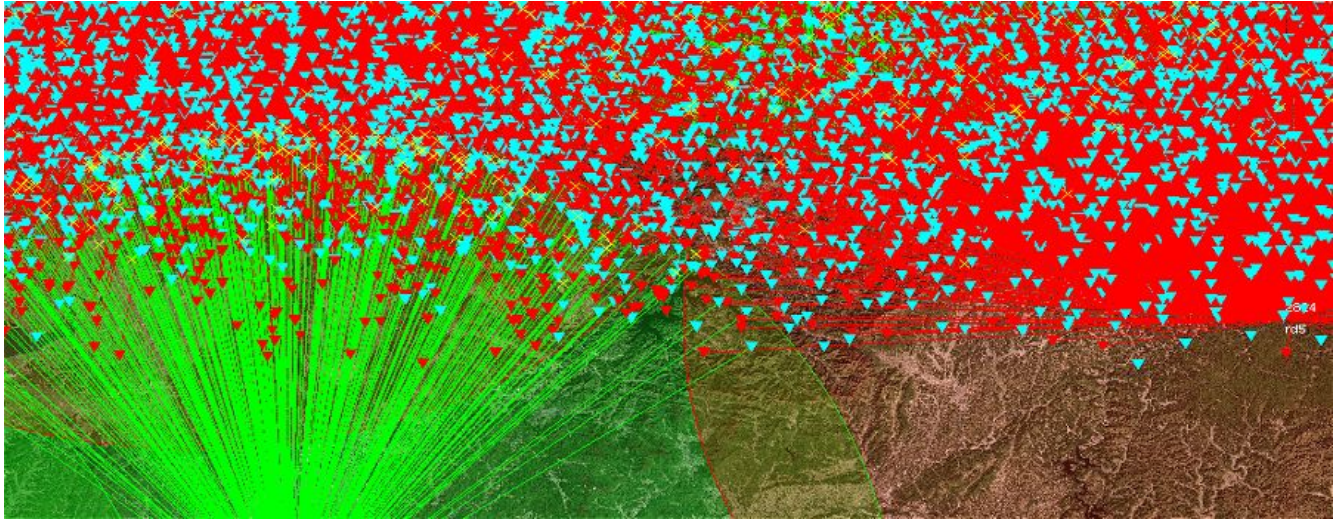
by VirtualSim

Customer requirements go sometime beyond what engineers consider enough in their product design. Stress testing is good as it fire-proves the software without state of mind, while the programmer go often too gently with his code.

Here, we need to be able to run around 30hz base frequency. 10 sensors of 10 km visibility range shall detect specific targets and destroy them. The detection has to take less than 1 second and the firing process must obeys to a specific selection rule with a random reload time delay.

For this stress test, a control panel has been designed with the vsTASKER HMI builder. On the map, 5 sensors detect green intruders and 5 others detect red. On the right, each of the sensors displays on a timeline the number of detections and hits. 20000 intruders are generated with a random position and speed.

- 20.000 flying intruders
- 10 sensors + AAA
- vsTASKER 7 with dedicated 10Hz HMI
- constraints: ~30Hz base rate, 1 second max for detection



OS & HARDWARE SUPPORT

INDUSTRY APPLICATION

Note: ** Specifications are copied from manufacturer data. Please contact before buying.