

# Chapter 8

## Task–Based Crowd Simulation for Heterogeneous Architectures

**Hugo Perez**

*Barcelona Supercomputing Center, Spain*

**Isaac Rudomin**

*Barcelona Supercomputing Center, Spain*

**Benjamin Hernandez**

*Oak Ridge National Laboratory, USA*

**Eduard Ayguade**

*Barcelona Supercomputing Center, Spain*

### ABSTRACT

*Industry trends in the coming years imply the availability of cluster computing with hundreds to thousands of cores per chip, as well as the use of accelerators. Programming presents a challenge due to this heterogeneous architecture; thus, using novel programming models that facilitate this process is necessary. In this chapter, the case of simulation and visualization of crowds is presented. The authors analyze and compare the use of two programming models: OmpSs and CUDA. OmpSs allows to take advantage of all the resources available per node by combining the CPU and GPU while automatically taking care of memory management, scheduling, communications and synchronization. Experimental results obtained from Fermi, Kepler and Maxwell GPU architectures are presented, and the different modes used for visualizing the results are described, as well.*

### INTRODUCTION

Nowadays there are 502 cities with over one million inhabitants, 74 with more than 5 million and 29 megacities with populations above 10 million; in total, there are around 3.9 billion human beings living in urban areas. As a result, it is important to develop computer models that describe the behavior of populations. Crowd simulations allow safe application of the scientific method and may aid in the analysis of certain subsets of the crowd phenomena, such as disease propagation, building evacuation during emergencies, traffic modeling or social evolution. Prediction before, during and after daily crowd events may reduce risks and associated logistic costs. Crowd simulation is usually expressed at macroscopic and microscopic levels (Pelechano, 2008). Macroscopic modeling describes global interactions

DOI: 10.4018/978-1-5225-0287-6.ch008

## ***Task-Based Crowd Simulation for Heterogeneous Architectures***

with the environment and the crowd itself. On the other hand, microscopic level exposes the interactions between individuals within a group; each agent should be processed individually to simulate a crowd. We are interested in modeling crowds at the microscopic level using Agent-Based Models (ABM). An ABM simulates the actions and interactions of autonomous agents; it promotes the progressive assembly of agents into groups resulting in global-scale behaviors (Bonabeau, 2002). In particular, this research focuses on real-time large-scale crowd simulation and in-situ visualization for heterogeneous architectures, considering the requirements that exascale computing will demand.

Real-time performance is crucial for applications such as risk mitigation or disaster control and offers the next advantages:

- Results are available immediately, minimizing time-to-solution.
- Simulation can be restarted almost instantaneously when unexpected outcomes are obtained, saving time and resources.
- Enables simulation steering allowing users to interact with the simulation through parameters.

In-situ visualization consists of visualizing results on the same cluster where simulation is running (Ahrens, J, 2010). This conveys two advantages for this research: relevant results are stored before simulation finishes, and data movement from processors through different memory hierarchies and storage is reduced or avoided.

Crowd simulation and visualization at megacity scale requires significant computational power and memory resources commonly available on High-Performance Computing platforms (HPC). According to Top500<sup>1</sup> Supercomputer Sites list, the use of heterogeneous architectures combining multi-core processors and accelerators is becoming a common practice. Examples to accelerators include Intel Xeon Phi and the NVIDIA Tesla GPU. In our purposes, the GPU has two advantages: accelerates computation, and efficient rendering for in-situ visualization.

Current HPC clusters have hundreds of thousands and even millions of processors. The community is in a transitioning stage between the petascale and the exascale era. Industry trends indicate that exascale clusters will have thousands of millions of cores (Moreland, 2013). The technical challenges for crowd simulation and visualization are intertwined with challenges from other areas including communication, programming models, and hardware architecture.

The rest of the chapter is organized as follows. The following section introduces the programming models, behavior models and current platforms commonly used in crowd simulation. The algorithm, implementation and results obtained from our case study are discussed in the Crowd Simulation section. After that, different crowd visualizations are described in the Visualization section. Finally, the chapter concludes.

## **BACKGROUND**

### **Programming Models**

A long time ago (2005), while still in Mexico, Our group (led by Dr. Rudomin) started to explore moving our crowd simulation from the CPU to the GPU. This because even then, simulation performance was significantly increased in comparison with desktop CPUs, allowing for simulation of larger crowds with

24 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage:

[www.igi-global.com/chapter/task-based-crowd-simulation-for-heterogeneous-architectures/159045](http://www.igi-global.com/chapter/task-based-crowd-simulation-for-heterogeneous-architectures/159045)

## Related Content

---

### Two Approaches of Workflow Scheduling with QoS in the Grid

Fangpeng Dong and Selim G. Akl (2009). *Quantitative Quality of Service for Grid Computing: Applications for Heterogeneity, Large-Scale Distribution, and Dynamic Environments* (pp. 1-27).

[www.irma-international.org/chapter/two-approaches-workflow-scheduling-qos/28268](http://www.irma-international.org/chapter/two-approaches-workflow-scheduling-qos/28268)

### A Traitor Tracking Method Towards Deep Learning Models in Cloud Environments

Yu Zhang, Linfeng Wei, Hailiang Li, Hexin Cai and Yingxin Wu (2022). *International Journal of Grid and High Performance Computing* (pp. 1-17).

[www.irma-international.org/article/a-traitor-tracking-method-towards-deep-learning-models-in-cloud-environments/301588](http://www.irma-international.org/article/a-traitor-tracking-method-towards-deep-learning-models-in-cloud-environments/301588)

### Time Restraint Load Balancing in the Cloud Environment

Nikita Malhotra, Sanjay Tyagi and Monika Singh (2022). *International Journal of Grid and High Performance Computing* (pp. 1-11).

[www.irma-international.org/article/time-restraint-load-balancing-in-the-cloud-environment/301592](http://www.irma-international.org/article/time-restraint-load-balancing-in-the-cloud-environment/301592)

### Near Sets in Assessing Conflict Dynamics within a Perceptual System Framework

Sheela Ramanna and James F. Peters (2010). *Novel Developments in Granular Computing: Applications for Advanced Human Reasoning and Soft Computation* (pp. 179-195).

[www.irma-international.org/chapter/near-sets-assessing-conflict-dynamics/44704](http://www.irma-international.org/chapter/near-sets-assessing-conflict-dynamics/44704)

### Serendipity Reloaded: Fair Loading in Event-Based Messaging

Daniel Cutting and Aaron Quigley (2010). *Principles and Applications of Distributed Event-Based Systems* (pp. 90-118).

[www.irma-international.org/chapter/serendipity-reloaded-fair-loading-event/44397](http://www.irma-international.org/chapter/serendipity-reloaded-fair-loading-event/44397)