



Online Game Prototype

May 13th, 2005

Introduction

This demo is an example of a prototypical online game application running on a Cell based server. The game prototype was designed with an emphasis on physically based modeling of rigid body dynamics.

The premise of this project was that next-generation online games will rely heavily on server based physical simulations to add an element of behavioral realism not available in the current state of the art. The team chose rigid body dynamics because of its increasing use in online games.

Our proposed solution leverages the Cell processor's SPEs to perform some of the more computationally intensive processing required for a game server. We use the SPEs to perform a hybrid integration calculation required to compute rigid-body displacements in a multi-body game scene.

Gameplay

The storyline for the game consists of mechanical robots attacking a city inhabited by humans forced to defend themselves by using a variety of weapons to disable the robots. These weapons include handheld rocket launchers, machine guns, and satchel explosives. A robot can be destroyed by directing weapons fire at vulnerability points located near arm, leg, etc. joints. Alternatively, humans can destroy static structures causing a robot to lose balance and fall or be hit by falling debris.

The articulation of robot and human joints resulting in realistic movement depends on solving several technical challenges. First, Collision detection was needed to determine when moving bodies intersecting with other moving bodies as well as static bodies such as walkways, buildings, terrain, etc. Collision detection was implemented with a 2 phase approach using broad phase to quickly eliminate bodies that could not physically collide with each other during any given frame update and narrow phase to specifically determine if pairs of bodies would intersect with each given frame



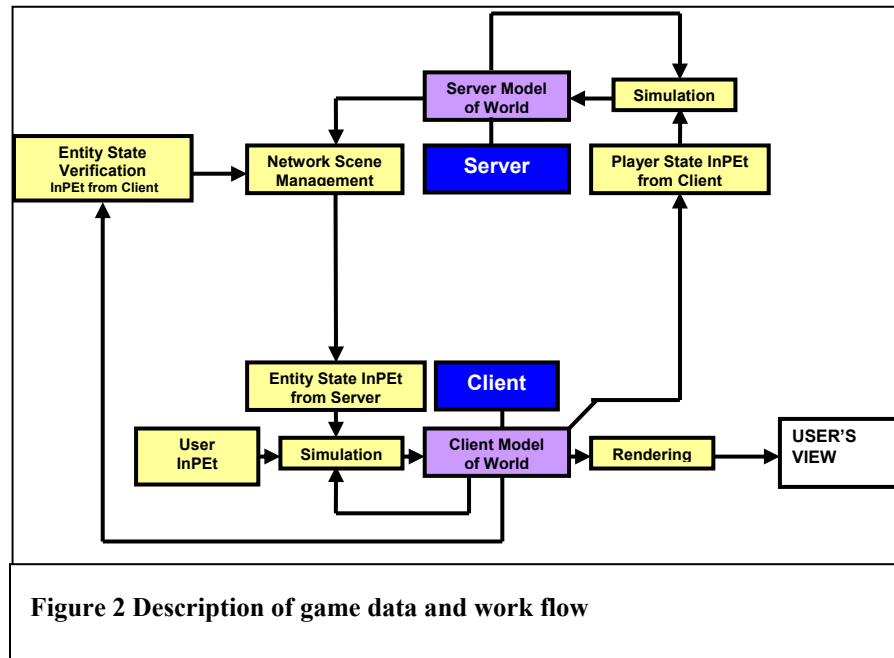
Figure 1 Mech-style game robot

update. The entire process was both integer and floating point intensive. Figure 1 shows a large robot with many articulating joints. Each pair of joints is represented in a database of collision bodies. Bodies that are static are referred to as “sleeping” while moving bodies are referred to as “awake”. The state of a body can change from frame to frame. Robot and player movement and balance were anticipated to be a difficult problem. Specifically, we did not want and moving bodies to look like they were “floating” over terrain or paved surfaces. Also, the articulation of joints should look smooth and coordinated. This would be especially challenging since network bandwidth limitations would require differential updates of bodies to reduce data flow from server to client. We considered

what granularity of updates in timeline would be appropriate to insure that articulating joints did not “drift” and separate over several frames.

Client/server synchronization was another important challenge. Due to network latency time we would need to compensate on the client for player response time that could not tolerate a round-trip communication between client and server. Compensating with the low performance client would require approximations that could result in a desynchronization of world state between the more accurately computed server and client.

Ultimately, aggregate outbound server network bandwidth was limited to 100Mb/s and we needed to consider data movement to client during state updates and packaging of information. Client to server communications was considerably less and more susceptible to higher latency than overall bandwidth.



Performance

We were focusing on how a Cell Processor Based Blade Servers could be used to accelerate gameplay for MMOG. One point to note is that the Cell performance profiling was done on early hardware that was not running at potential maximum clock speed and only had 6 (as opposed to 8) functioning SPEs.

During the implementation phase, we had considered double buffering data I/O to/from SPE local store. Profiling determined that numerical integration executing on the SPE had an extremely high compute to DMA time ratio - approx 182, i.e. DMA time (single buffered) is less than 1% of all SPE execution time. This suggested that double buffering input and output buffers would likely reduce overall performance by lowering maximum workload size without providing a speedup.

By streaming small integration work loads to all 8 SPEs and vectorizing the code we were able to achieve an overall integration speedup that was more than 8x that of a single processor solution. Since integration and collision detection accounted for the largest percentage of server side computational load, we concluded that Cell Processor Based Blade Servers could potentially be used to provide significant performance increase for online games that heavily relied on physical simulation.

Authored by:

Bruce D'Amora
 IBM Research
damora@us.ibm.com
 914-784-6514



© IBM Corporation 2005
IBM Corporation
Systems and Technology Group
Route 100
Somers, New York 10589

Produced in the United States of America
May 2005
All Rights Reserved

This document was developed for products and/or services offered in the United States. IBM may not offer the products, features, or services discussed in this document in other countries.

The information may be subject to change without notice. Consult your local IBM business contact for information on the products, features and services available in your area.

All statements regarding IBM future directions and intent are subject to change or withdrawal without notice and represent goals and objectives only.

IBM, the IBM logo, are trademarks or registered trademarks of International Business Machines Corporation in the United States or other countries or both. A full list of U.S. trademarks owned by IBM may be found at:
<http://www.ibm.com/legal/copytrade.shtml>.

Other company, product, and service names may be trademarks or service marks of others.

IBM hardware products are manufactured from new parts, or new and used parts. Regardless, our warranty terms apply.

Photographs show engineering and design models. Changes may be incorporated in production models. Copying or downloading the images contained in this document is expressly prohibited without the written consent of IBM

This equipment is subject to FCC rules. It will comply with the appropriate FCC rules before final delivery to the buyer.

Information concerning non-IBM products was obtained from the suppliers of these products or other Public sources. Questions on the capabilities of the non-IBM products should be addressed with those suppliers.

All performance information was determined in a controlled environment. Actual results may vary. Performance information is provided "AS IS" and no warranties or guarantees are expressed or implied by IBM

The IBM home page on the Internet can be found at: <http://www.ibm.com>.

