Libspe2 SPU Optimized Bullet Physics SDK

Bullet provides a parallel collision detector and parallel constraint solver in Extras/BulletMultiThreaded. This was originally being developed for Playstation 3, in collaboration with Playstation 3 game developers.
Playstation 3 licensed developers can request a SPURS SPU optimized version through Sony Playstation 3 Devnet.

A threading interface abstracts the SPURS task scheduler, and allow other threading implementations. Apart from Playstation 3 proprietary SPURS task scheduling support, the public open source version of Bullet contains Win32 Threads support for Windows and XBox 360 platform ,as well as Libspe2 support for IBM CellBlade and Playstation 3 Linux. From Bullet 2.63 onwards it is possible to compile Bullet Physics SDK with specific optimizations running on Cell SPUs through the Libspe2 SPU scheduler. Most of the collision detection can run on one or multiple SPUs in parallel.

Steps how to compile for IBM Cell SDK 2.1 for Playstation 3 Linux using the included Makefiles:

0) Download Bullet 2.63 or later, unzip to a folder and let's call this BULLET_ROOT
1) Change directory to $(BULLET_ROOT)/src/ibmsdk
run make
This should create the 3 core Bullet PPU libraries (bulletmath.a, bulletcollision.a and bulletdynamics.a), and places them in $(BULLET_ROOT)/lib/ibmsdk
2) Change directory to $(BULLET_ROOT)/Extras/BulletMultiThreaded
run make all
This should create a PPU library (bulletmultithreaded.a) placed in $(BULLET_ROOT)/lib/ibmsdk.
It also creates a SPU executable (spuCollision.elf) placed in
$(BULLET_ROOT)/Extras/BulletMultiThreaded/out
3) Change directory to $(BULLET_ROOT)/Demos/OpenGL/ibmsdk
run make
This creates OpenGL support functionality (bulletopenglsupport.a), placed in
$(BULLET_ROOT)/lib/ibmsdk
Change directory to $(BULLET_ROOT)Demos/CcdPhysicsDemo/ibmsdk
run make
This creates CcdPhysicsDemo, a PPU executable
4) Test if all works so far, by running the PPU-only demo, while still in directory
$(BULLET_ROOT)/Demos/CcdPhysicsDemo/ibmsdk :
./CcdPhysicsDemo
This should run the demo fine, showing a stack of cylinders. Press 'q' to quit the demo
5) Now enable SPU support by enabling #define USE_PARALLEL_DISPATCHER 1 in
CcdPhysicsDemo.cpp, located in $(BULLET_ROOT)/Demos/CcdPhysicsDemo
6) copy spuCollision.elf in the current folder, next to the CcdPhysicsDemo executable :
cp $(BULLET_ROOT)/Extras/BulletMultiThreaded/out/spuCollision.elf
$(BULLET_ROOT)/Demos/CcdPhysicsDemo/ibmsdk
7) run the CcdPhysicsDemo, again from the same folder as step 4
./CcdPhysicsDemo
8) To enable debugging DMA get/put operations (for mis-alignment, wrong sizes etc), you can enable DMA Debugging:
Around line 42 of file $(BULLET_ROOT)/Extras/BulletMultiThreaded/SpuFakeDma.h, enable #define DEBUG_DMA

Several option to choose the maximum outstanding number of threads/tasks, packet sizes will be documented.
Feedback is welcome through the Bullet Physics forums at http://bulletphysics.com
Thanks,
Erwin Coumans