lzrt Performance by Lukasz Bruun (2009) - www.lukasz.dk

1 Job Benchmark

The flow in lzrt is made up of several kernels which each perform a specific job (transforming triangles, building the kd-tree,..). On the Cell processor each of these kernels is a SPU program, which the PPU loads when necessary. If the entire screen was to be drawn by one thread, the SPU kernels would be uploaded the minimum number of times to the SPE. However, lzrt is a multithreaded ray tracing application and to balance the workload a threadpool of ray tracing jobs is created. Each job is responsible for drawing a part of the screen, the more jobs, the smaller part of the screen each job will draw. By having many jobs we minimize the difference between the workload for each job and by using a threadpool, each thread can start a new job, when it has finished it's previous one. What we want is to have all the threads finish their jobs about the same time. The problem with this approach on the Cell is that there is a bit of an overhead of loading the SPU kernels and for each time we load a new ray tracing job, we upload all the kernels again. So the more jobs we create, the more tiems we have upload SPU kernels to the SPEs. The effect on performance this has is quite severe. On figure 1 is shown the performance of one thread running a different number of ray tracing jobs on the x86 2.2 Ghz, Cell PowerPC, and a Cell SPU. The thing to notice is that both the x86 and PowerPC performance is not affect by more jobs, however the SPU performance drops. The conclusion is, if you are making a high performance Cell application, the best results are achieved if you only upload your SPU program once and signal it to start, instead of uploading it each time you want to start it.

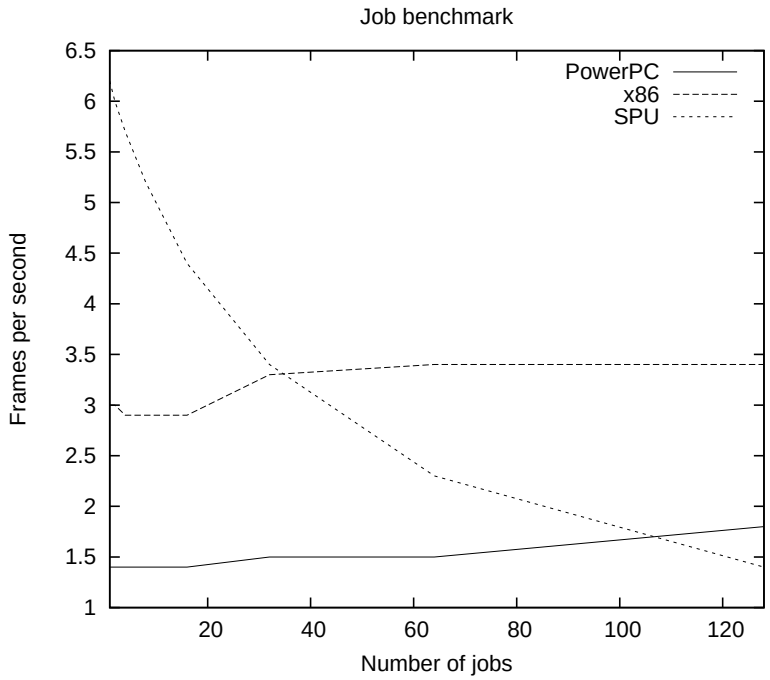

Figure 1: Job Benchmark

2 Triangle Intersection Benchmark

The SPUs on the Cell processor are good at performance computational intensive workloads. In lzrt one of such workloads which need to be performed is intersection test between rays and triangles. To measure the performance of a SPU and compare it to a 2.2 GHz x86 CPU and the Cell PowerPC, I tested intersection against a number triangles in a 512x512 screen resolution (262144 rays). The results are shown in figure 2. The figure shows that for a small number of triangles, the performance is for the SPU is quite superior, but more equal to the two other processors, as the number of triangles increases. The reason for this is unconfirmed, but the way that lzrt transfers and processes the triangles on the SPU could be be a cause, combined with the fact that the each SPU only has 256 KB of memory and therefor alot more transfering of data is necessary as the number triangles increase.

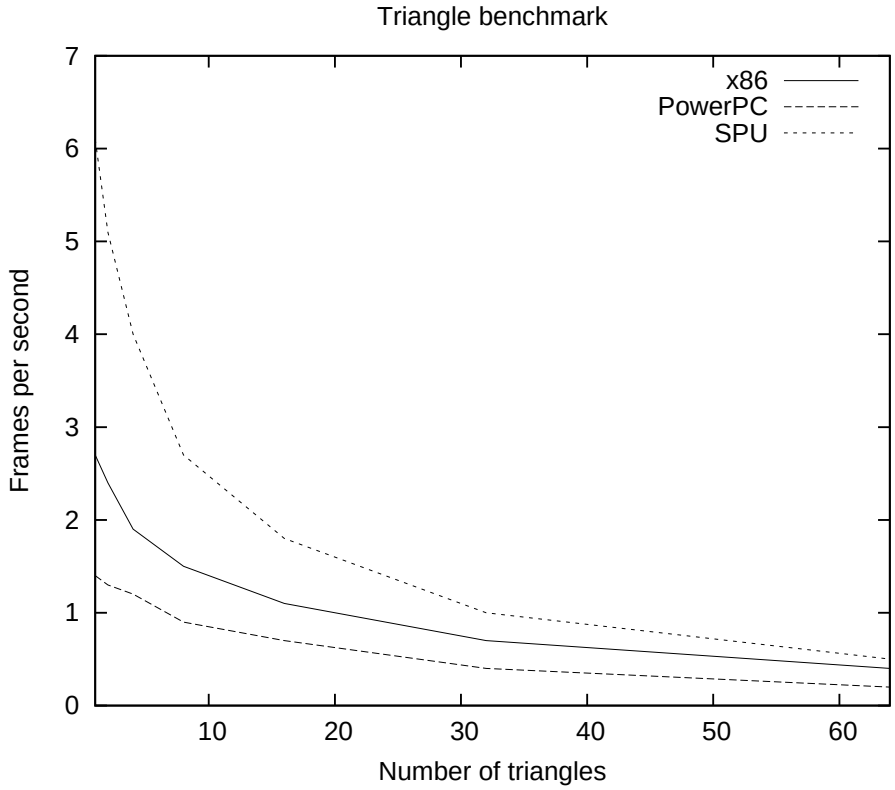

Figure 2: Triangle Benchmark Triangle