BigGPU

A Distributed PTX Compilation and Execution System on Hybrid CPU/GPU Clusters

HPDS Lab, K.U.A.S. Electrical Engineering

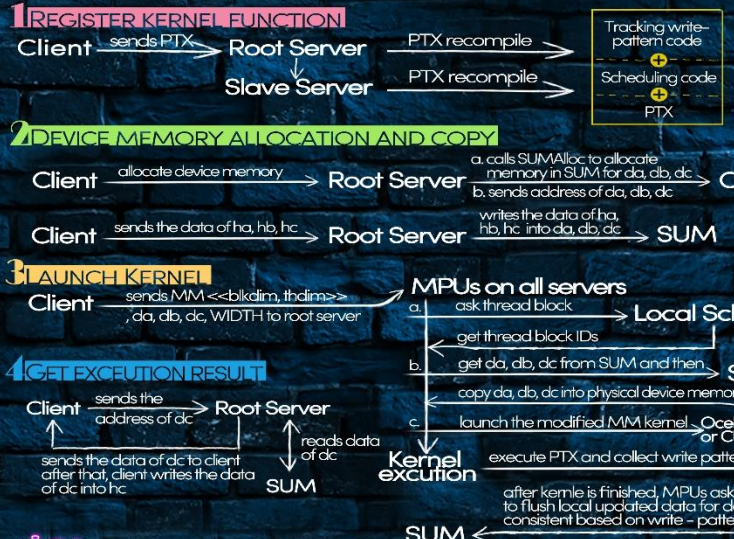
Introduction

BigGPU is a distributed PTX compilation and execution system. From the view point of users, BigGPU can be regarded as a single CUDA-compatible GPU with lots of virtual stream processors and a virtual large device memory space which physically are emulated by integrating CPUs and GPUs distributed over networks.

BigGPU can support

- Single GPU programming for hybrid CPU/GPU clusters
- Execution of large-scale kernel functions
- Dynamic load balance among heterogeneous processors

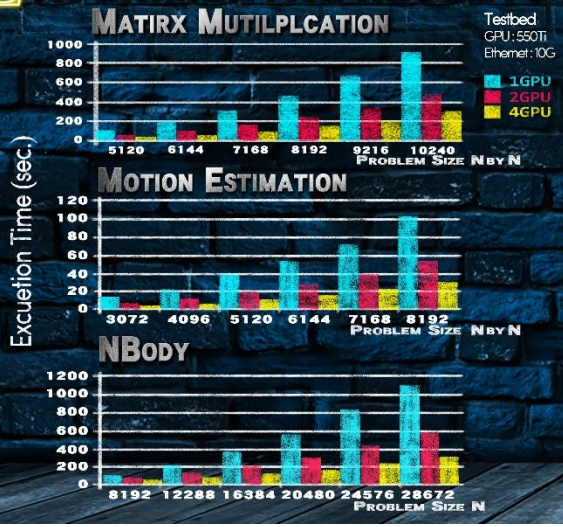
Framework



Matrix Multiplication

```
1 __global__ void MM(int *A,int *B, int *C,int, N){
    int i=blkDim.x*blkIdx.x+thIdx.x;
    int r=i/N,c=i%N;
    for(int n=0; n<N; ++n)
      C[r*N+c]+=A[r*N+n]*B[n*N+c];
  }
2 static void mkInts(int **h,int **d){
    *h=(int*)malloc(65536*4);
    cudaMalloc(d,65536*4);
    cudeMemcpy(d,h,HtoD);
  }
  int main(){
    int *ha,*hb,*hc,*da,*db,*dc;
    mkInts(&ha,&da);
    mkInts(&hb,&db);
    mkInts(&hc,&dc);
3   MM<<<256,256>>>(da,db,dc,256);
4   cudeMemcpy(dc,hc,65536*4,DtoH);
    free(ha);free(hb);free(hc);
  }
```

Experimental Results



Contribution

- Simplify the CUDA programming because of large-scale kernels.
- Reduce the programming complexity of GPU or hybrid CPU/GPU clusters because of no need of hybrid MPI/CUDA or MPI/OpenMP programming.
- Helpful for moving problems to hybrid CPU/GPU clusters in order for saving energy and obtaining high performance.

- Software Unified Memory(SUM) : a software distributed shared memory based on eager-released consistency.
- MPU : Multiple Processor Unit
- In BigGPU a device can be a CPU or GPU.

References

[1] Gregory Diamos, Andrew Kerr, Sudhakar Yalamanchili, Nathan Clark, "Ocelot: a dynamic optimization framework for bulk-synchronous applications in heterogeneous systems", PACT 2010, pp. 353-364.
[2] CUDA Programming Guides, http://docs.nvidia.com/cuda/#programming-guides.
[3] Parallel Thread Execution ISA , http://docs.nvidia.com/cuda/parallel-thread-execution.
[4] Tyng-Yeu Liang, Chun-Yi Wu, Jyh-Biau Chang, Ce-Kuen Shieh, "Teamster-G : A Grid-enabled Software DSM System", CCGrid 2005, pp. 905-912.

sponsor: National Science Council(Project No.NSC-102-2221-E-151-028)

Hung-Fu Li | Yu-Jie Lin | Bi-Shing Chen | Tyng-Yeu Liang