GPU Computing with Dirac

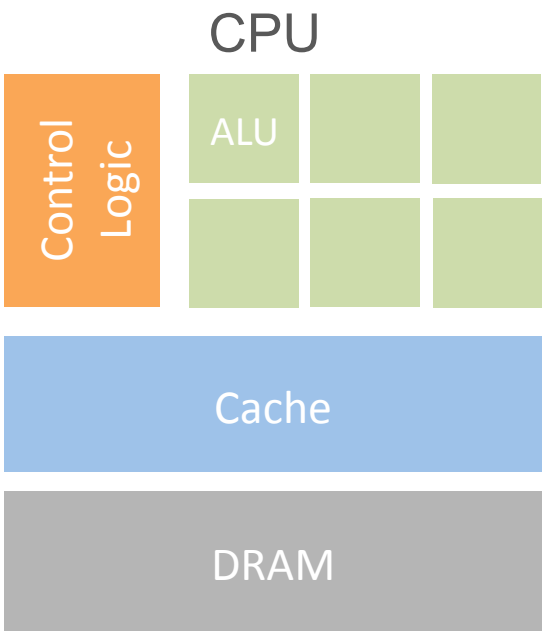
Hemant Shukla

U.S. DEPARTMENT OF ENERGY | Office of Science

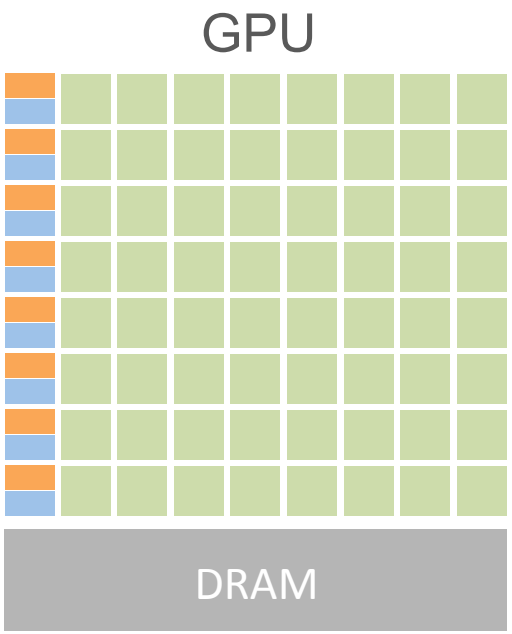
NERSC National Energy Research Scientific Computing Center

BERKELEY LAB Lawrence Berkeley National Laboratory

Architectural Differences



Less than 20 cores
1-2 threads per core
Latency is hidden by large cache



512 cores
10s to 100s of threads per core
Latency is hidden by fast context switching

Programming Models

CUDA (Compute Unified Device Architecture)

OpenCL

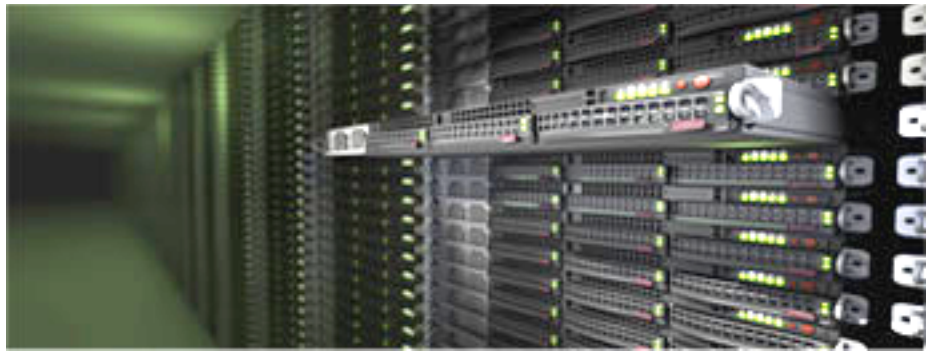
Microsoft's DirectCompute

Third party wrappers are also available for **Python, Perl, Fortran, Java, Ruby, Lua, MATLAB, IDL,** and **Mathematica**

Compilers from **PGI, RCC, HMPP, Copperhead, OpenACC**

DIRAC

Hardware



44 Nodes

1 Tesla C 2050 (Fermi) GPU
2 Nehalem Quad Core
24 GB of DDR3

4 Nodes

1 Tesla C 1060 (Fermi) GPU
2 Nehalem Quad Core
24 GB of DDR3

1 Node

4 Tesla C 1060 GPU
2 Nehalem Quad Core
24 GB DDR3

1 Node

4 Tesla C 2050 GPU
2 Nehalem Quad Core
24 GB DDR3

Interconnect

4X QDR InfiniBand
(32 Gb/s)

DIRAC ACCOUNT

Request an account of Dirac

http://www.nersc.gov/users/computational-systems/dirac/getting-started/request-for-dirac-account/

For questions contact NERSC Consultants at *consult@nersc.gov*

DIRAC Access

CARVER IBM iDataPlex

Dirac nodes reside on Carver

DIRAC

SINGLE NODE ACCESS

```
> ssh -l <username>@carver.nersc.gov
> qsub -I -V -q dirac_int -l nodes=1:ppn=8:fermi
qsub: waiting for job 1822582.cvrsvc09-ib to start
qsub: job 1822582.cvrsvc09-ib ready

[dirac28] >
```

1. Log into Carver
2. Start interactive node on Dirac
3. Dirac node prompt

First CUDA Program

```
#include <stdio.h>
int main()
{
        int noOfDevices;
        /* get no. of device */
        cudaGetDeviceCount (&noOfDevices);

        cudaDeviceProp  prop;
        for (int i = 0; i < noOfDevices; i++)
        {
                /*get device properties */
                cudaGetDeviceProperties (&prop, i );

                printf ("Device Name:\t %s\n",  prop.name);
                printf ("Total global memory:\t %ld\n”, prop.totalGlobalMem);
                printf (”No. of SMs:\t %d\n”, prop.multiProcessorCount);
                printf ("Shared memory / SM:\t %ld\n", prop.sharedMemPerBlock);
                printf("Registers / SM:\t %d\n", prop.regsPerBlock);

        }
        return 1;
}
```

whatDevice.cu

First CUDA Program

Compilation (on Carver)

```
> module load cuda
        OR
> modlue load cuda/<version>

> nvcc whatDevice.cu –o whatDevice
```

Log into Dirac and run the code

```
> qsub -I -V -q dirac_int -l nodes=1:ppn=8:fermi

[dirac27] > ./whatDevice
```

Output

```
Device Name:            Tesla C2050
Total global memory:    2817982464
No. of SMs:             14
Shared memory / SM:     49152
Registers / SM:         32768
```

CUDA + OpenMP + MPI

```
> gcc –fopenmp someCode.cu –o someCode      [include omp.h]

> nvcc –Xcompiler -fopenmp -lgomp someCode.cu –o someCode
```

```
> mpicc -c main.c -o main.o
> nvcc -c kernel.cu -o kernel.o

> mpicc main.o kernel.o -L/usr/local/cuda/lib64 -lcudart
```

Matlab

2D FFT Example

```
>> N = 5000;
>> a = rand (N,N);
>> b = gpuArray(a);

>> tic; r1 = fft2(a); toc;
Elapsed time is 1.135059 seconds.

>> tic; gather (r1); toc;
Elapsed time is 0.015606 seconds.

>> tic; r2 = fft2(b); toc;
Elapsed time is 0.423533 seconds.

>> tic; gather (r2); toc;
Elapsed time is 0.253305 seconds.
```

Running Batch Jobs

submit.job

```
#PBS -q dirac_reg
#PBS -l nodes=16:ppn=8:fermi
#PBS -l walltime=00:10:00
#PBS -A gpgpu
#PBS -N my_job
#PBS -e my_job.$PBS_JOBID.err
#PBS -o my_job.$PBS_JOBID.out
#PBS -V

cd $PBS_O_WORKDIR
mpirun -np 128 ./my_executable
```

```
> qsub submit.job
```

Programming Models on Dirac

CUDA

OpenCL

Matlab

HMPP

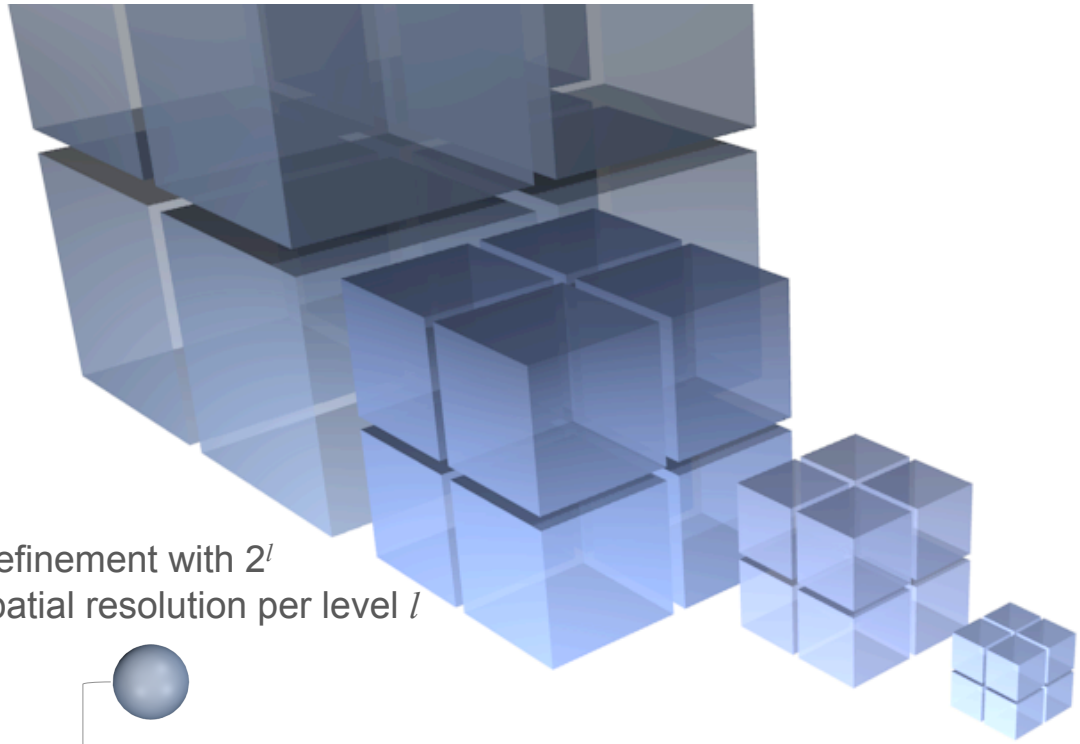
MINT

RCC

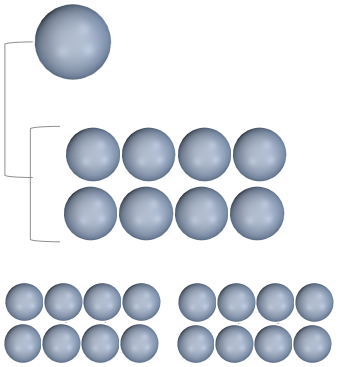
PGI (CUDA-FORTRAN)

OpenACC

Adaptive Mesh Refinement



Refinement with 2^l
spatial resolution per level l



Data stored in Octree data structure

2D Patch

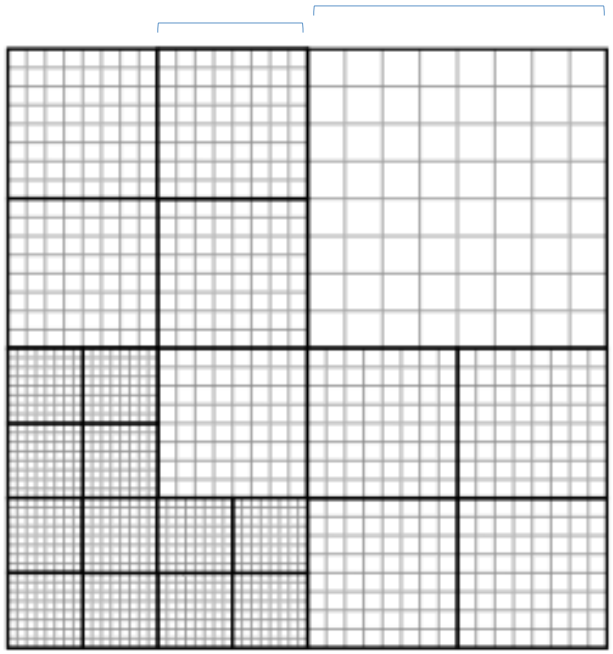


Figure - Hsi-Yu Schive *et al.*, 2010

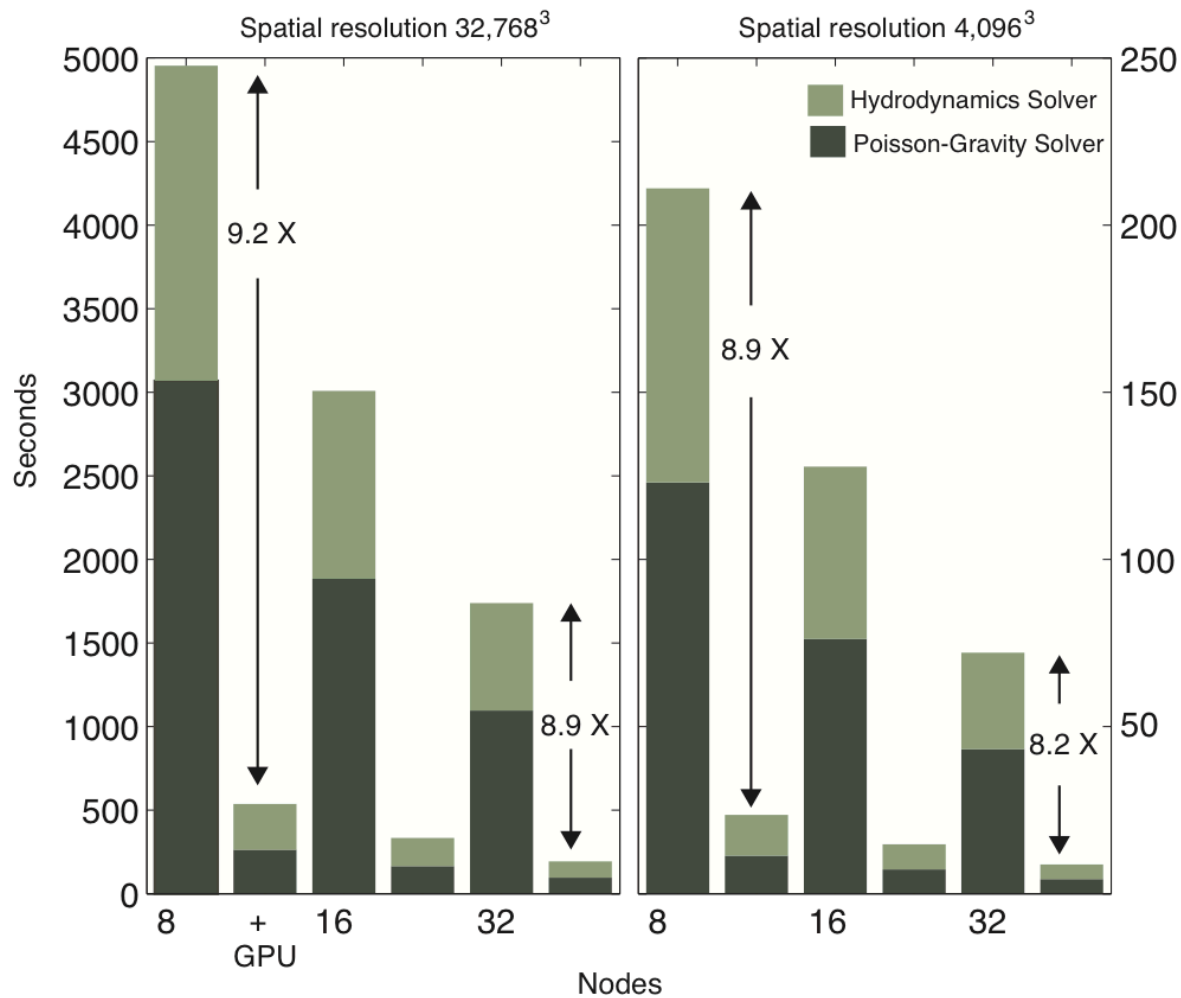
8^3 cells per patch

Identical spatial geometry (same kernel)

Uniform and individual time-steps

Results

Large scale Cosmological Simulations with GAMER



Hemant Shukla,
Hsi-Yu Schive
et al. SC 2011

For More Information

http://www.nersc.gov/users/computational-systems/dirac/