Data-Intensive Science Using GPUs

Alex Szalay, JHU
Data in HPC Simulations

• HPC is an instrument in its own right
• Largest simulations approach petabytes
  – from supernovae to turbulence, biology and brain modeling
• Pressure for public access to the best and latest through interactive numerical laboratories
• Creates new challenges in
  – How to move the petabytes of data (high speed networking)
  – How to look at it (render on top of the data, drive remotely)
  – How to interface (smart sensors, immersive analysis)
  – How to analyze (value added services, analytics, …)
  – Architectures (supercomputers, DB servers, ??)
Visualizing Petabytes

• Needs to be done where the data is...
• It is easier to send a HD 3D video stream to the user than all the data
  – *Interactive visualizations driven remotely*
• Visualizations are becoming IO limited: precompute octree and prefetch to SSDs
• It is possible to build individual servers with extreme data rates (5GBps per server… see Data-Scope)
• Prototype on turbulence simulation already works: data streaming directly from DB to GPU
• N-body simulations next
Immersive Turbulence

“… the last unsolved problem of classical physics…” Feynman

• Understand the nature of turbulence
  – Consecutive snapshots of a large simulation of turbulence: now 30 Terabytes
  – Treat it as an experiment, play with the database!
  – Shoot test particles (sensors) from your laptop into the simulation, like in the movie Twister
  – Now: 70TB MHD simulation

• New paradigm for analyzing simulations!

with C. Meneveau, S. Chen (Mech. E), G. Eyink (Applied Math), R. Burns (CS), K. Kanov, E. Perlman (CS), E. Vishniac
Sample code (fortran 90)

do it = 1,15,1
   print *, 'time = ', time
   time = time - deltat
   CALL getvelocity(authkey, dataset1, time, Lagrangian6thOrder, PCHIPInterpolation, 4*n, points, dataout)
   do i = 1,4*n
      do k = 1,3
         points(k,i) = points(k,i) - dataout(k,i)*deltat
      end do
   end do
   if (it.eq.5.or.it.eq.10.or.it.eq.15) then
      do i = 1,4*n
         write(10,*) points(1,i),points(2,i),points(3,i)
      end do
      write(10,*) points(1,1),points(2,1),points(3,1)
   endif
   write(10,*) ''
end do
endif

Not possible during DNS

advect backwards in time!

minus
Integrated Visualization

- Experiment on GPU integration with databases
- Kai Buerger, R. Westermann (TUM, Munich)
- Turbulence data in database, 100 snapshots stored on SSD array for fast access
- Data stored in $8^3$ array datatype in DB, organized along a space filling curve (z-Index)
- Query fetches cubes in arbitrary order to GPU
- Cube copied into proper location on GPU
- Rendering using DirectX-10 engine
Streaming Visualization of Turbulence

Kai Buerger, Technische Universitat Munich, 24 million particles
Architectural Challenges

• How to build a system good for the analysis?
• Where should data be stored
  – *Not at the supercomputers (too expensive storage)*
  – *Computations and visualizations must be on top of the data*
  – *Need high bandwidth to source of data*
• Databases are a good model, but are they scalable?
  – *Google (Dremel, Tenzing, Spanner: exascale SQL)*
  – *Need to be augmented with value-added services*
• Makes no sense to build master servers, scale out
  – *Cosmology simulations are not hard to partition*
  – *Use fast, cheap storage, GPUs for some of the compute*
  – *Consider a layer of large memory systems*
JHU Data-Scope

- Funded by NSF MRI to build a new ‘instrument’ to look at data
- Goal: ~100 servers for $1M + about $200K switches+racks
- Two-tier: performance (P) and storage (S)
- Mix of regular HDD and SSDs + GPUs
- Large (5PB) + cheap + fast (400+GBps), but...
  ...a special purpose instrument

### Final configuration

<table>
<thead>
<tr>
<th></th>
<th>1P</th>
<th>1S</th>
<th>All P</th>
<th>All S</th>
<th>Full</th>
</tr>
</thead>
<tbody>
<tr>
<td>servers</td>
<td>1</td>
<td>1</td>
<td>90</td>
<td>6</td>
<td>102</td>
</tr>
<tr>
<td>rack units</td>
<td>4</td>
<td>34</td>
<td>360</td>
<td>204</td>
<td>564</td>
</tr>
<tr>
<td>capacity</td>
<td>24</td>
<td>720</td>
<td>2160</td>
<td>4320</td>
<td>6480 TB</td>
</tr>
<tr>
<td>price</td>
<td>8.8</td>
<td>57</td>
<td>8.8</td>
<td>57</td>
<td>792 $K</td>
</tr>
<tr>
<td>power</td>
<td>1.4</td>
<td>10</td>
<td>126</td>
<td>60</td>
<td>186 kW</td>
</tr>
<tr>
<td>GPU</td>
<td>1.35</td>
<td>0</td>
<td>121.5</td>
<td>0</td>
<td>122 TF</td>
</tr>
<tr>
<td>seq IO</td>
<td>5.3</td>
<td>3.8</td>
<td>477</td>
<td>23</td>
<td>500 GBps</td>
</tr>
<tr>
<td>IOPS</td>
<td>240</td>
<td>54</td>
<td>21600</td>
<td>324</td>
<td>21924 kIOPS</td>
</tr>
<tr>
<td>netwk bw</td>
<td>10</td>
<td>20</td>
<td>900</td>
<td>240</td>
<td>1140 Gbps</td>
</tr>
</tbody>
</table>

Amdahl Number 1.38
JHU Jetson Cluster
Cosmology Simulations

• Millennium DB is the poster child/ success story
  – 600 registered users, 17.3M queries, 287B rows
  http://gavo.mpa-garching.mpg.de/Millennium/
  – Dec 2012 Workshop at MPA: 3 days, 50 people
• Data size and scalability
  – PB data sizes, trillion particles of dark matter
  – Where is the data stored, how does it get there
• Value added services
  – Localized (SED, SAM, SF history, posterior re-simulations)
  – Rendering (viz, lensing, DM annihilation, light cones)
  – Global analytics (FFT, correlations of subsets, covariances)
• Data representations
  – Particles vs hydro grid
  – Particle tracking in DM data
  – Aggregates, uncertainty quantification
Crossing the PB Boundary

- Via Lactea-II (20TB) as prototype, then Silver River (50B particles) as production (15M CPU hours)
- 800+ hi-rez snapshots (2.6PB) => 800TB in DB
- Users can insert test particles (dwarf galaxies) into system and follow trajectories in pre-computed simulation
- Users interact remotely with a PB in ‘real time’
  Madau, Rockosi, Szalay, Wyse, Silk, Kuhlen, Lemson, Westermann, Blakeley

- INDRA (512 1Gpc box with 1G particles, 1.1PB)
Dark Matter Annihilation

• Data from the Via Lactea II Simulation (400M particles)
• Computing the dark matter annihilation
  – *simulate the Fermi satellite looking for Dark Matter*
• Original code by M. Kuhlen runs in **8 hours** for a single image
• New GPU based code runs in **24 sec**, Point Sprites, Open GL shader language. [Lin Yang (Forrest), grad student at JHU]
• Interactive service (design your own cross-section)
• Approach would apply very well to gravitational lensing and image generation (virtual telescope)
Interactive Web Service

THE MILKY WAY IN DARK MATTER
Changing the Cross Section

Annihilation (No Correction)
Annihilation (1/ν Correction)
Annihilation (1/ν² Correction)
Decay Map (No Correction)
Multi-Epoch Blind Deconvolution

Tamas Budavari, Matthias Lee
Sequence Alignment on GPUs

- Richard Wilton, Ben Langmed, Steve Salzberg, Alex Szalay, Sarah Wheelan, Tamas Budavari
Summary

• Amazing progress in the last 5 years
• New challenges emerging:
  – Petabytes of data, trillions of particles
  – Increasingly sophisticated value added services
  – Need a coherent strategy to go to the next level
• It is not just about storage, but how to integrate access, computation and visualization
• Petabyte-scale streaming problems, ideal for GPUs
• Bridging the gap between data server and supercomputer
  – Easy to add GPUs to data servers!!
• Democratizing the use of large simulations