Data Intensive Computing on Heterogeneous Platforms

Norm Rubin
Fellow
GPG graphics products group
AMD
HPEC 2009
What might we see in future platforms?

Multi-core implies reprogramming so all kinds of new architectures are possible

The traditional view of more performance per year, based on clock changes is over

New approaches will change our view of computing platforms

Compute is about to change radically
The drivers for compute

1. New more natural user interfaces
2. Search (e.g., finding things in images) Massive data
3. Search over time changing data (e.g., live video) Massive data
4. Machine learning (starting with simple models) Massive data
5. Standard compute tasks

Data intensive covers 2, 3, and 4
Massive Data and machine learning

Data-driven models become tractable and usable
Less need for analytical models
Less need for heuristics
Real-time connectivity enables continuous model refinement
Poor model is an acceptable starting point

Classification accuracy improves over time
Simple Models

Google demonstrated the value of applying massive amounts of computation to language translation in the 2005 NIST machine translation competition.

They won all four categories of the competition translating Arabic to English and Chinese to English.

Purely statistical approach

multilingual United Nations documents comprising over 200 billion words, as well as English-language documents comprising over one trillion words.

No one in their machine translation group knew either Chinese or Arabic.

How Technology May Soon "Read" Your Mind

Establish the correspondence between a simple cognitive state (such as the thought of a hammer) and the underlying brain activity.

Use machine learning techniques to identify the neural pattern of brain activity underlying various thought processes. MRI scanner generates data on the brain activity. A database of responses of individuals can then be created.

Using fMRI Brain Activation to Identify Cognitive States Associated with Perception of Tools and Dwellings
New ways to program graphics

A domain expert (the programmer), 3 cameras, silhouette extraction and a large data base of human motions

Use 3D movement as a programming language

Three Video Streams

Silhouettes

010...010101
Yaw Feature Vector

111...010101
Body Configuration Feature Vector

Yaw Hash Tables

010...1011
010...1100
111...0100
0 Degree Yaw

110...1011
110...1110
111...0100
Selected Yaw

111...1111
111...1100
111...0101
350 Degree Yaw

Animated Sequence

Human Motion Database
How will GPU/CPU change to meet big data?

Current machine models are two separate devices
GPU – a graphics thing, maybe also a data parallel accelerator, good for lots of data, small programs
CPU – a serial processor, maybe capable of task parallelism, good for limited data, big programs

CPU strength is single thread performance
(latency machine)
GPU strength is massive thread performance
(throughput machine)
GPU compared with CPU

**GPU**
- Great float
- Great bandwidth
- High performance without fixed isa, lots of room for innovation, without breaking code different to program
- But
- Limited scratchpad memory in place of a cache

**CPU**
- Great control flow
- Lots of existing code
- Locked into an legacy isa
- Very good at reuse, real caches
- But
- Limited ability to scale with cores
Some limitations of today's GPU

general cross thread communication
task switching, task parallel problems
nested parallelism
dynamic parallelism
very expensive to move data between GPU/CPU
programmer controlled small scratch pad memory
The 3C view of the future, (c-cubed?)

We are in a time of architectural change much like the switch from mainframes to microprocessors.

abundant content, connections, compute

Slash dot http://science.slashdot.org/article.pl?sid=09/08/20/1233258

NASA Probe Blasts 461 Gigabytes of Moon Data Daily

On its current space scouting mission, NASA’s Lunar Reconnaissance Orbiter (LRO) is using a pumped up communications device to deliver 461 gigabytes of data and images per day, at a rate of up to 100 Mbps. As the first high data rate K-band transmitter to fly on a NASA spacecraft, the 13-inch-long tube, called a Traveling Wave Tube Amplifier, is making it possible for NASA scientists to receive massive amounts of images and data about the moon's surface and environment... It kills me that the moon has better bandwidth than my house.
Software Challenge

Most of the successful parallel applications seem to have dedicated languages (DirectX®11/map-reduce/sawzsall) for limited domains.

Small programs can build interesting applications, Programmers are not super experts in the hardware Programs survive machine generations

Can we replicate this success in other domains?
Can we get performance with high level languages?

CUDA and OpenCL are good languages in the hands of experts but they are still too low level.

We need high level programming models that express computations over data.

One possible model is “MapReduce”

Map part selects interesting data
Reduce part combines the interesting data.
MapReduce

Map: for each input in parallel
  - if the input is interesting – output a key and a value, values might be compound structures

Sort based on the keys

Reduce: in parallel for each key
  - for each value with the same key, combine
Observations on MapReduce

Developers only write a small part of the program, rest of the code comes from libraries
No race conditions are possible
No error reporting (just keep going)
Can view the program as serial (per input)
No developer knows the number of processors
Not like pthreads

Hard part is reductions, but it appears that there are only a few that are common, so maybe they can be prebuilt
Notice nested parallelism in MapReduce

For each key (in parallel)
    do a parallel reduction (also parallel)

One key may have lots of values, a second key might have a few (work stealing? Dynamic parallelism?)
K means clustering (representative algorithm)

Given N objects each with A attributes and K possible cluster centers

Map:
for each object find the distance to each center
Classify the object into a cluster, based on min distance
Key is the cluster, data is the object

Reduce: for all objects in the same cluster,
find the new cluster centers

Repeat till no object changes clusters
K means demo
K means performance experimental data

CPU 4 - core i7 processor
GPU (2 NV 260 mid range GPU’s)

GPU programmed in CUDA
CPU programming in TBBU
(joint work with Balaji Dhanasekaran, U of Virginia)

AMD has announced a new higher performance GPU card (58xx series), which supports the industry standard language OpenCL.

Performance numbers using OpenCL on AMD hardware
Will be posted at http://developer.amd.com/Pages/default.aspx
# Performance

4 million objects, 2 attributes, 100 clusters
50 iterations

<table>
<thead>
<tr>
<th>Map</th>
<th>Reduce</th>
<th>Time (sec)</th>
<th>4 CPU</th>
<th>1CPU</th>
</tr>
</thead>
<tbody>
<tr>
<td>2GPU</td>
<td>2GPU</td>
<td>.81</td>
<td>40.9</td>
<td>194</td>
</tr>
<tr>
<td>2GPU</td>
<td>4CPU</td>
<td>1.29</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1GPU</td>
<td>1GPU</td>
<td>1.54</td>
<td>21.5</td>
<td>102</td>
</tr>
<tr>
<td>1GPU</td>
<td>4CPU</td>
<td>1.71</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4CPU</td>
<td>4CPU</td>
<td>33.11</td>
<td>1</td>
<td>4.75</td>
</tr>
<tr>
<td>1CPU</td>
<td>1CPU</td>
<td>157.34</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Measured performance for other sizes shows similar results, better for larger attributes, objects.
K-means

Do most of the work on the GPU, but do some of the reductions on the CPU if you have enough idle cores. Preliminary numbers suggest GPU can solve this very well.

Can we develop software that automatically adjusts for this, or do programmers have to write variant algorithms?
Disclaimer and Attribution

DISCLAIMER

The information presented in this document is for informational purposes only and may contain technical inaccuracies, omissions and typographical errors.

AMD MAKES NO REPRESENTATIONS OR WARRANTIES WITH RESPECT TO THE CONTENTS HEREOF AND ASSUMES NO RESPONSIBILITY FOR ANY INACCURACIES, ERRORS OR OMISSIONS THAT MAY APPEAR IN THIS INFORMATION.

AMD SPECIFICALLY DISCLAIMS ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE. IN NO EVENT WILL AMD BE LIABLE TO ANY PERSON FOR ANY DIRECT, INDIRECT, SPECIAL OR OTHER CONSEQUENTIAL DAMAGES ARISING FROM THE USE OF ANY INFORMATION CONTAINED HEREIN, EVEN IF AMD IS EXPRESSLY ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

ATTRIBUTION

© 2009 Advanced Micro Devices, Inc. All rights reserved. AMD, the AMD Arrow logo, ATI, the ATI logo, CrossFireX, PowerPlay and Radeon and combinations thereof are trademarks of Advanced Micro Devices, Inc. Other names are for informational purposes only and may be trademarks of their respective owners.
Questions?