Scalable Resource Composition in a Flat World

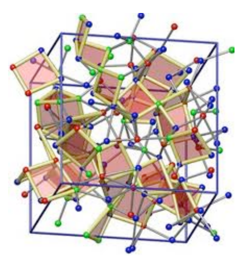
Sudhakar Yalamanchili

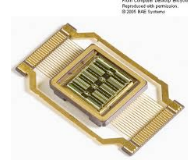
Computer Architecture and Systems Laboratory
Center for Experimental Research in Computer Systems
School of Electrical and Computer Engineering
Georgia Institute of Technology

Sponsors: National Science Foundation, NVIDIA, LogicBlox Inc.

System Diversity

Amazon EC2 GPU Instances

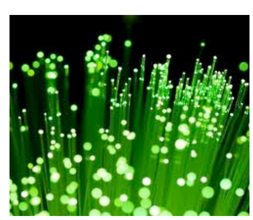




Phase Change Memory

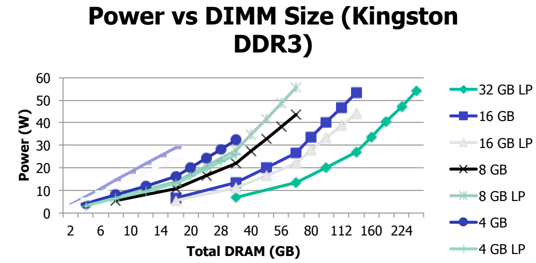
Technology Diversity is mainstream

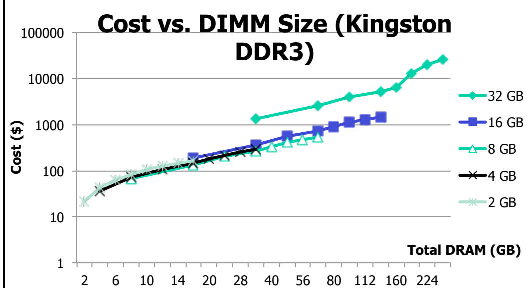
Cray Blue Waters



Photonics

Composing Memory



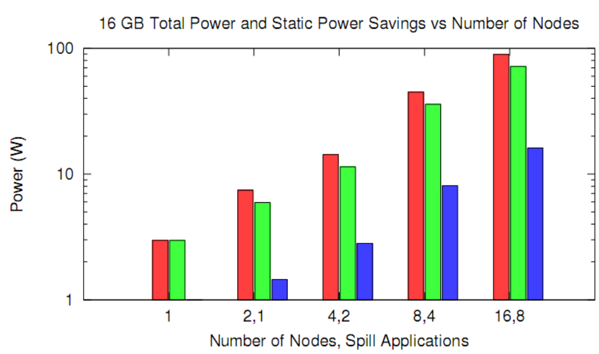


- While 8 GB DIMMs are most cost-effective, larger DIMM sizes are most power-efficient

Resource Sharing

Statistics generated for Supermicro X9DRL-3F Motherboard (8 DIMM slots)

A Simple Model of Savings[2]



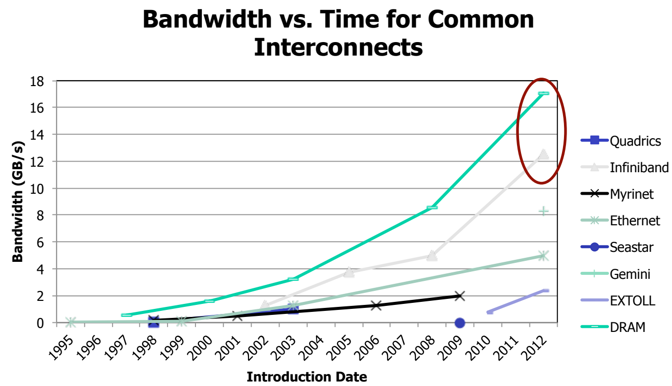
(16GB + Sharing) vs. 20GB MICRON DDR2

- 1.5 to 16 Watts per node on average
- Savings for a 10,000 core data center would be 3,540 Watts[1]
- Synthetic footprint traces from server applications

[1] *HP Power Advisor utility: a tool for estimating power requirements for HP ProLiant server systems, 2009,* http://h20000.www2.hp.com/bc/docs/support/SupportManual/c01861599/c01861599.pdf

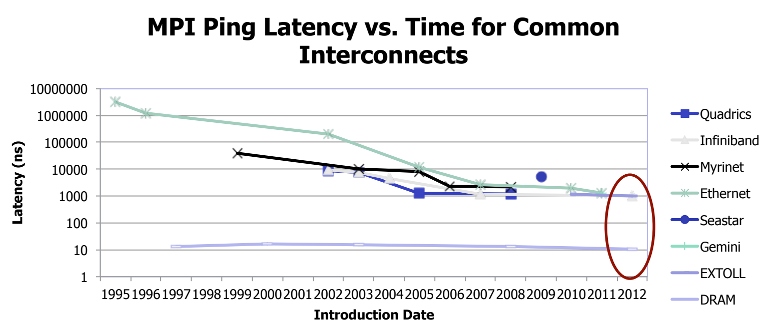
[2] J. Young and S. Yalamanchili, "Dynamic Partitioned Global Address Spaces for Power Efficient DRAM Virtualization," *IEEE Workshop on Work in Progress in Green Computing, August, 2010*

Bandwidth Trends



- DRAM to interconnect bandwidth ratio has been steadily dropping

Latency Trends

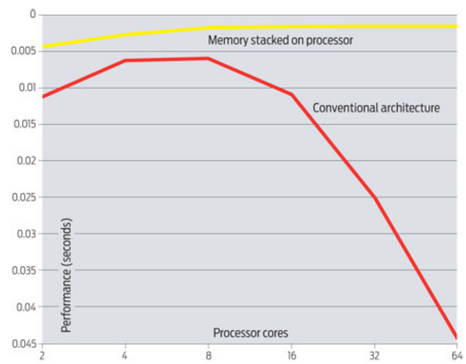


- MPI latency has steadily approached DRAM read latency
  - Hardware switching times in the low hundreds of nanoseconds.
- Note progress in photonics

Extend the reach of a socket

The Memory Wall

"Multicore Is Bad News For Supercomputers"
IEEE Spectrum 2008

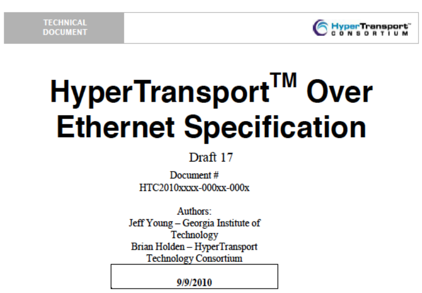


- Data intensive applications
- Memory bandwidth demand is scaling faster than memory interface capacity

"You can buy bandwidth but you cannot bribe God"
- unknown

Convert Network Bandwidth into Memory Bandwidth

Impact on Clustering


TECHNICAL DOCUMENT

HyperTransport Consortium

HyperTransport™ Over Ethernet Specification

Draft 17

Document #
HTC2010xxxx-000xx-000x

Authors:
Jeff Young – Georgia Institute of Technology
Brian Holden – HyperTransport Technology Consortium

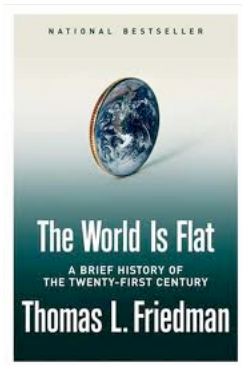
9/9/2010

- Combine commodity interconnects and memory systems
- Need flexible hardware level composition of resources
- This is an old idea whose time has come?

Some Examples

- Lim, et al. - Memory Blades for disaggregated memory
- Tolentino, Cameron – Memory Miser OS level support
- Lefurgy, et al. – DRAM server power and DRAM consolidation
- RDMA - Liang '05 low-level implementation for page swapping
- Memscale – UoH, UPC
- Feng et. Al – Green Supercomputing

Flattening Cluster Hierarchies



Observation I:
Everyone is getting closer and we need better sharing but…..

Post Dennard Performance Scaling

$$Perf\left(\frac{ops}{s}\right) = Power(W) \times \underbrace{Efficiency\left(\frac{ops}{joule}\right)}$$

Dally, Keynote IITC 2012

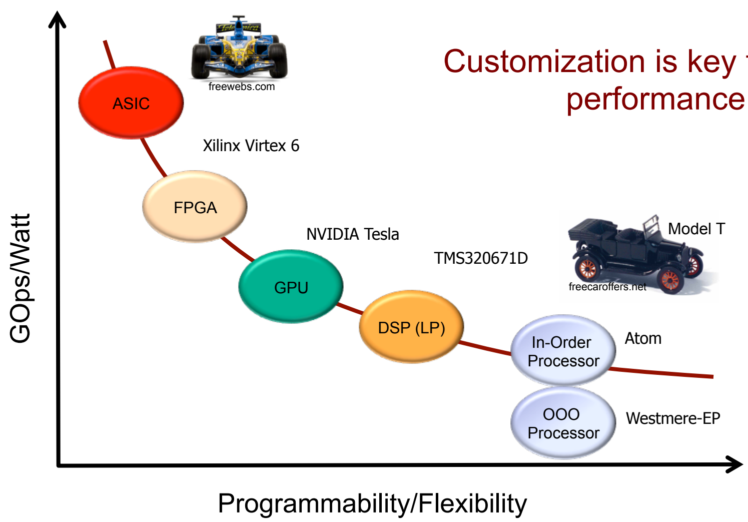
$$\underbrace{\text{Operator\_cost}} + \underbrace{\text{Data\_movement\_cost}}$$

Three operands x 64 bits/operand

Specialization → heterogeneity and asymmetry

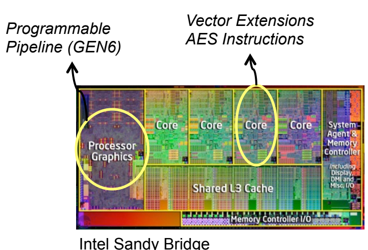
$$Energy = \#\,bits \times dist-mm \times energy-bit-mm$$

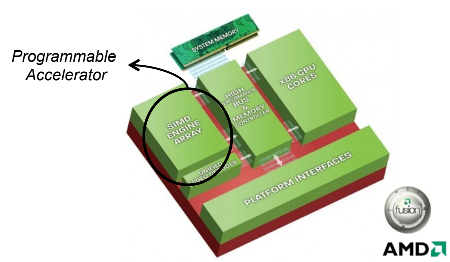
Hardware Power-Performance Tradeoffs



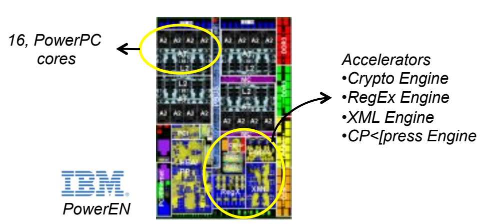
Customization is key to power performance!

Consolidation on Chip

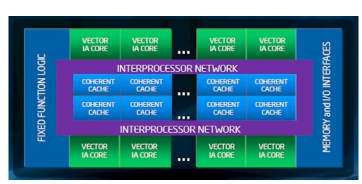




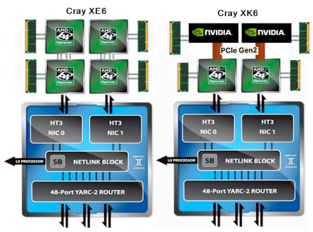
Multiple Models of Computation
Multi-ISA



Intel Knights Corner

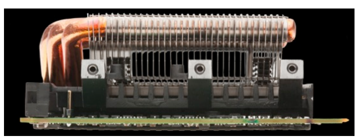


Consolidation in a System



perisofparallel.blogspot.com

So its not just memory that needs to be shared!



NVIDIA Tesla

amazon webservices™

Post Dennard Performance Scaling

$$Perf\left(\frac{ops}{s}\right) = Power(W) \times \underbrace{Efficiency\left(\frac{ops}{joule}\right)}$$

Dally, Keynote IITC 2012

$$\underbrace{Operator\_cost} + \underbrace{Data\_movement\_cost}$$

Specialization → heterogeneity and asymmetry

Three operands x 64 bits/operand

$$Energy = \#bits \times dist - mm \times energy - bit - mm$$

Scaling: Key Driver is Energy/Power

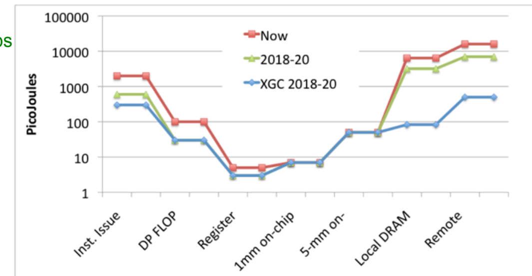
Embedded Platforms

Goal: 1-100 GOps

Big Science: To Exascale

Goal: 20MW/Exaflop

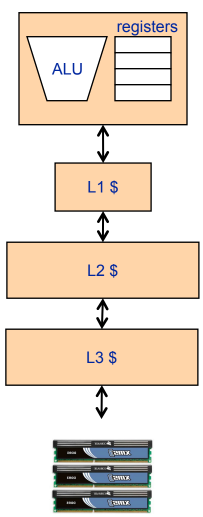
Cost of Data Movement



Courtesy: Sandia National Labs :R. Murphy).

- Sustain performance scaling through massive concurrency
  - New execution models
- Data movement becomes more expensive than computation

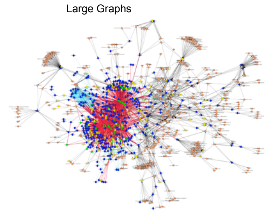
Optimizing Locality



Observation II:
You can hide latency, but you cannot hide energy!

A Data Rich World

Large Graphs



Mixed Modalities and levels of parallelism

Irregular, Unstructured Computations and Data

facebook

Pharma

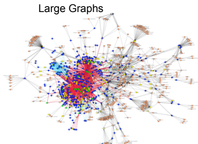
conventioninsider.com

Images from math.nist.gov, blog.thefuturescompany.com,melihsozdinler.blogspot.com

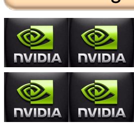
Trend analysis

System Model

Large Graphs



Programming Models	*Domain Specific Languages*
Data Movement Optimizations	*Compiler and Run-Time Support*
System Abstractions e.g. GAS, Virtual DIMMs, etc	*Cluster Wide Hardware Consolidation*



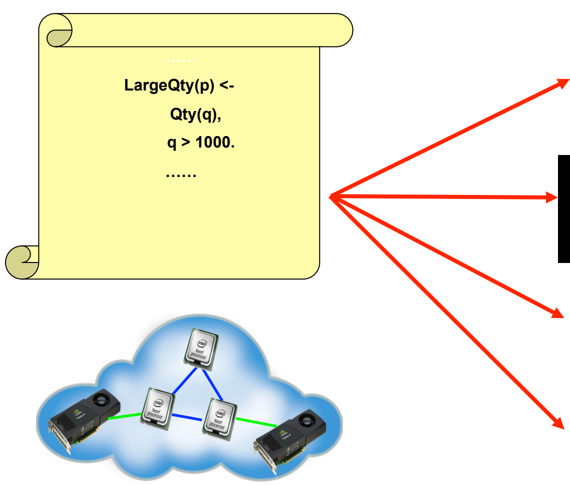
Hardware Customization

Application: Data Warehousing

- On-line and off-line analysis
  - Retail analysis
  - Forecasting
  - Pricing
  - ……
- Combination of data queries and computational kernels
- Current applications process 1 to 50 TBs of data [1]
- Potential to change a companies business model!

[1] Independent Oracle Users Group. A New Dimension to Data Warehousing: 2011 IOUG Data Warehousing Survey.

Databases: *Not* a Traditional Domain of GPUs



Relational Computations Over Massive Data Sets

Database Applications on GPUs

- The good
  - Lots of potential data parallelism
  - If data fits in GPU mem, 2x—27x speedup has been shown
- The bad
  - Very large data set (will not even fit in host memory)
  - I/O bound (GPU has no disk)
  - PCI data transfer takes 15–90% of the total time*
- The Ugly
  - Irregular/unstructured accesses to data

Order	Price	Discount
0	10	10%
1	20	20%
2	10	15%
3	51	14%
4	33	13%
5	22	10%
……	……	……

* B. He, M. Lu, K. Yang, R. Fang, N. K. Govindaraju, Q. Luo, and P. V. Sander. Relational query co-processing on graphics processors. In TODS, 2009.

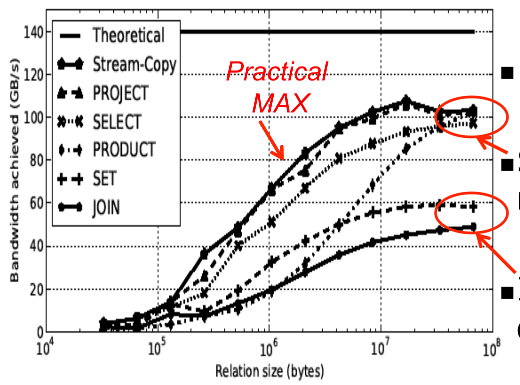
Research Thrusts

- I: Optimized implementations of primitives
  - Relational algebra
  - Data management within the GPU memory hierarchy
- II: In-core processing
  - Cluster wide memory aggregation techniques
  - Change the ratio of host memory size to accelerator memory size
- III: Data movement optimizations
  - Between hosts and (local or remote) accelerators
  - Within an accelerator

I. Relational Algebra Primitives on GPUs

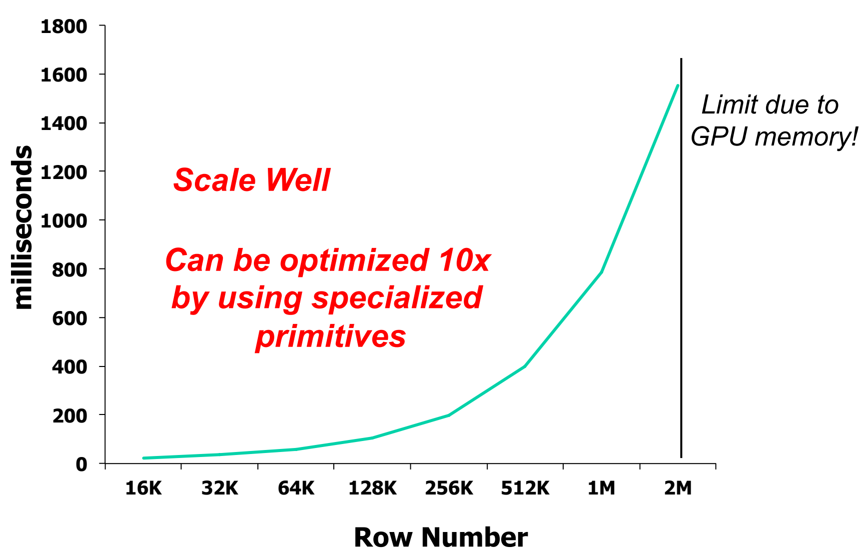
Raw Performance (C2050)

Fastest in GPU

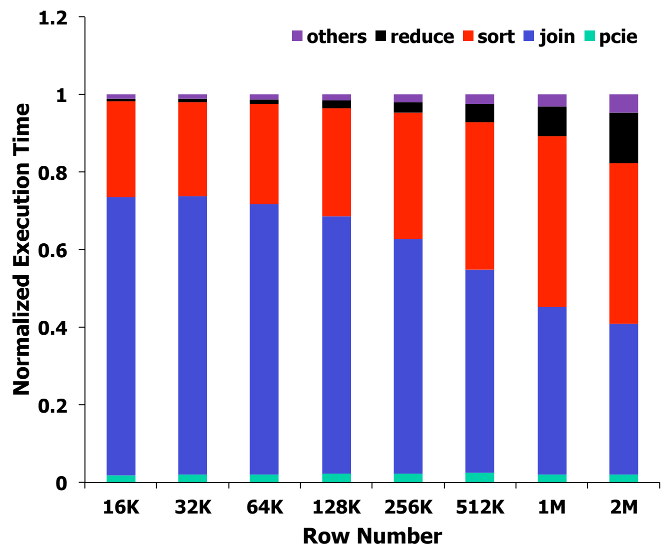


- Multi-stage algorithm (under review)
- Push to memory-bound
- Simple primitives are close to maximum performance
- Improved primitives under development

TPC-H Query 1-Overall Performance



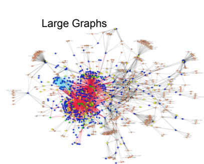
TPC-H Query 1-Breakdown Normalized Performance



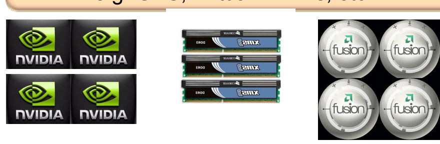
Spends most time in JOIN and SORT

Solutions (TBD) use Radix Sort

System Model

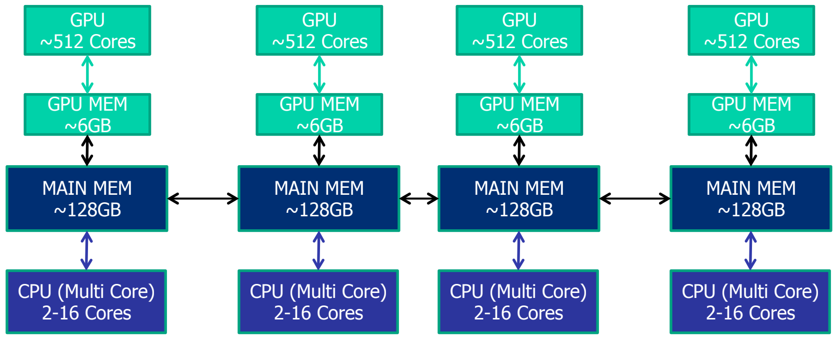


Programming Models	*Domain Specific Languages*
Data Movement Optimizations	*Compiler and Run-Time Support*
System Abstractions e.g. GAS, Virtual DIMMs, etc	*Cluster Wide Hardware Consolidation*



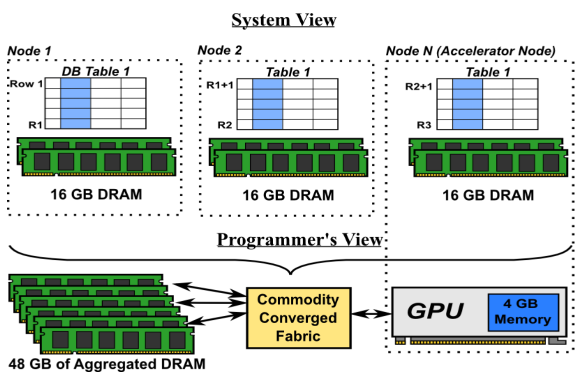
Hardware Customization

II. In-Core Processing



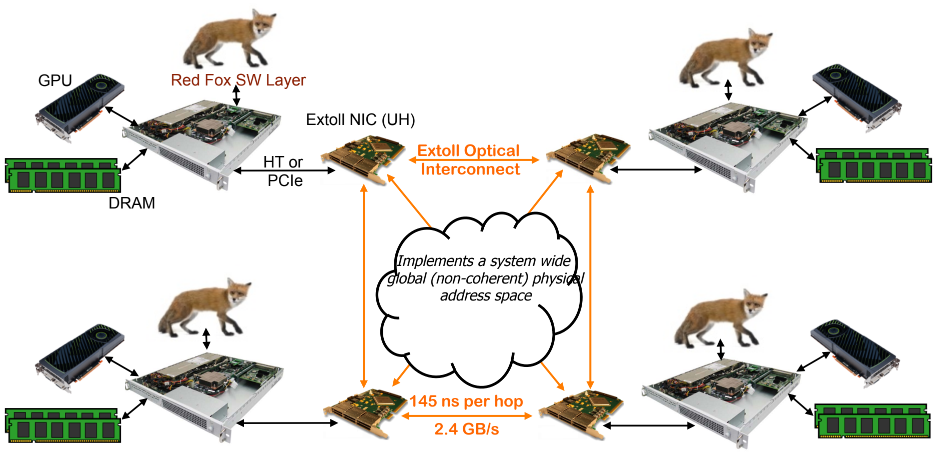
- Cluster-based memory aggregation
- Hardware support for global non-coherent, physical address space system
- Change the ratio of host-memory : GPU-memory

Oncilla: Fabrics for Accelerator Clouds



- Goal: Efficient memory aggregation for accelerators in data centers
- Solution: Use Global Address Spaces (GAS) and commodity fabrics (HT, QPI, PCIe, 10GE, IB)
  - Support in-core databases using software from Red Fox project

Oncilla Infrastructure

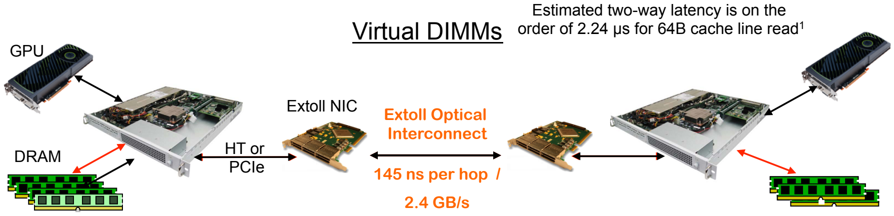


- Low-latency, commodity hardware (Extoll) for efficient memory and GPU aggregation and Red Fox SW layer supports DB queries on remote nodes
- Collaboration with University of Heidelberg (UH), Polytechnic University of Valencia, AIC Inc., LogicBlox Inc.

Some Candidate Systems Concepts

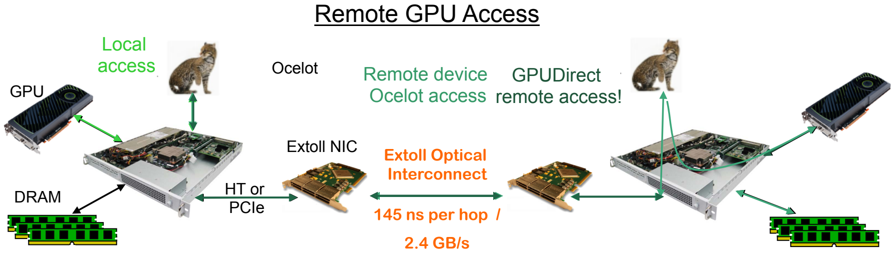
Virtual DIMMs

Estimated two-way latency is on the order of 2.24 μs for 64B cache line read[1]

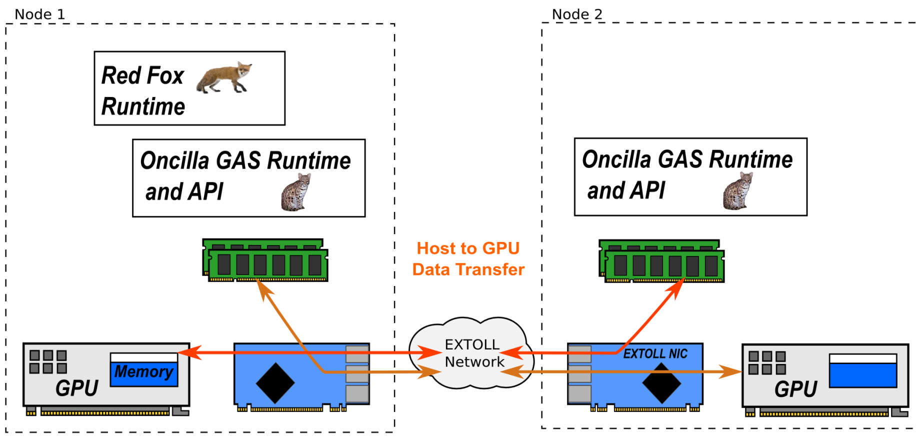


1) *Young, J., Yalamanchili, S., Dynamic Partitioned Global Address Spaces for Power-Efficient DRAM Virtualization, WIPGC at IGCC, 2010*

Remote GPU Access

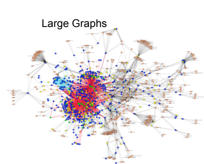


Oncilla



- The Oncilla GAS runtime simplifies memory management and enables low-latency scheduling
- Combines best features of NVIDIA's UVA and EXTOLL hardware provides a simplified API to scheduler to describe available compute and memory resources
  - This API will be used to ... data movement and query management can be used to optimize data warehousing applications with remote accelerators.

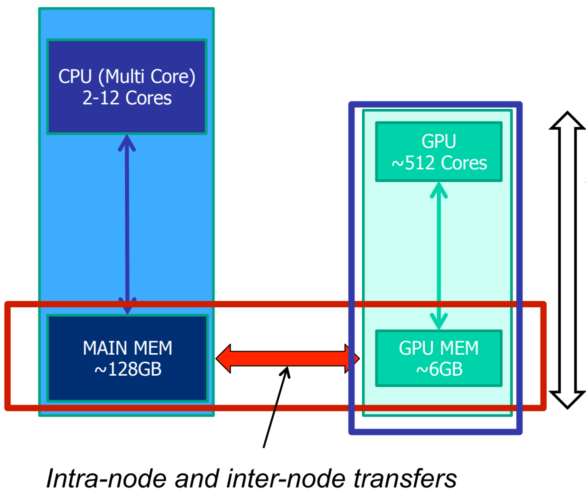
System Model



Programming Models	*Domain Specific Languages*
Data Movement Optimizations	*Compiler and Run-Time Support*
System Abstractions e.g. GAS, Virtual DIMMs, etc	*Cluster Wide Hardware Consolidation*

Hardware Customization

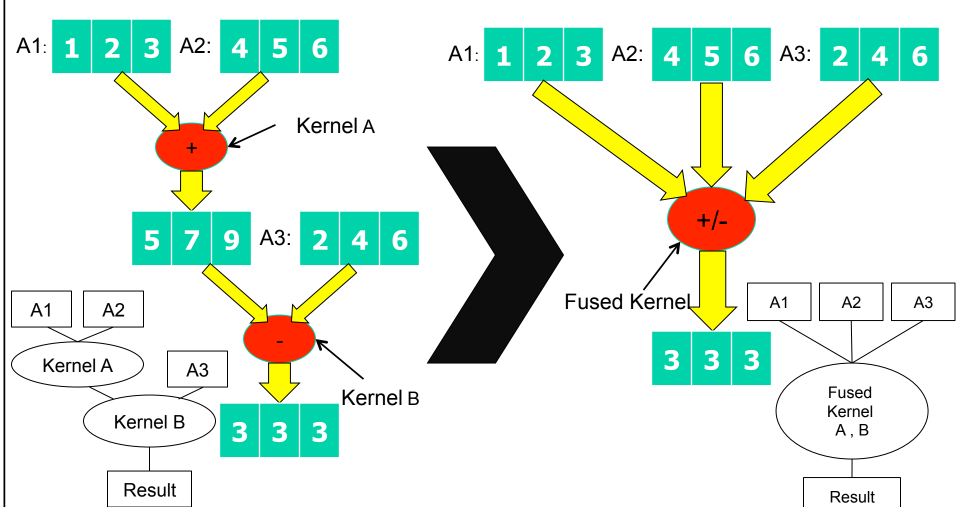
III. Data Movement Optimizations



Kernel Fusion – Aggregate computation to reuse data

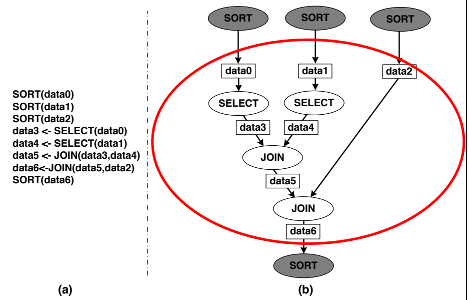
Kernel Fission – Overlap computation with data transfer

Kernel Fusion

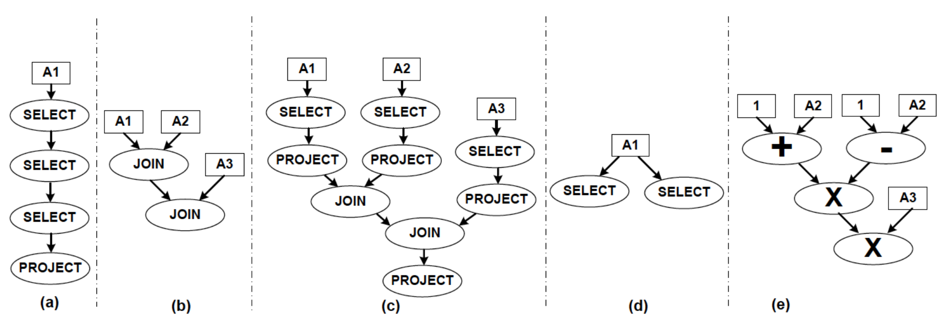


Kernel Fusion Benefits

- Smaller Data Footprint
  - Reduction in Memory Accesses
  - Temporal Data Locality
  - Reduction in Traffic
  - Larger Input Data
- Larger Optimization Scope
  - Common Computation Elimination
  - Improved Compiler Optimization Benefits

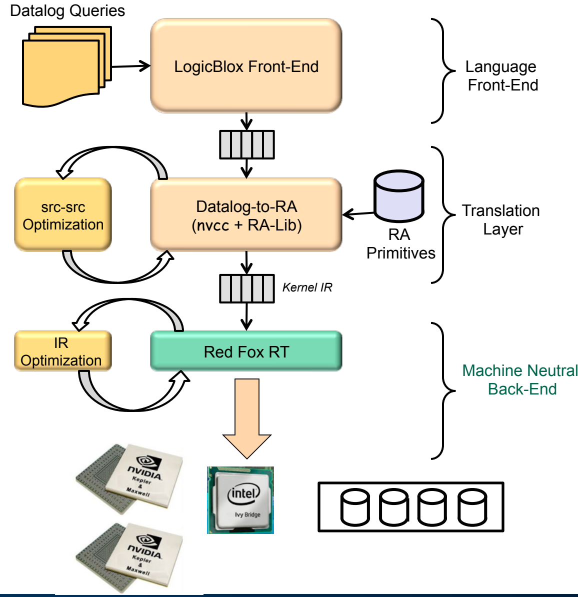


Common RA Combinations of TPC-H



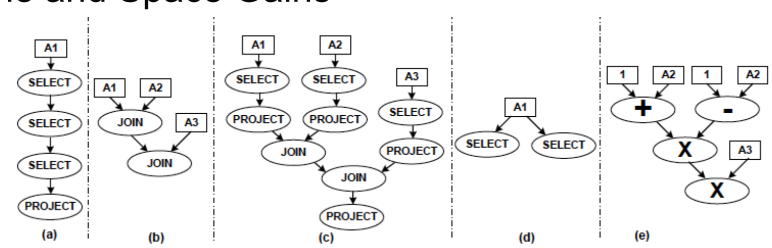
Domain Specific Compilation: Red Fox

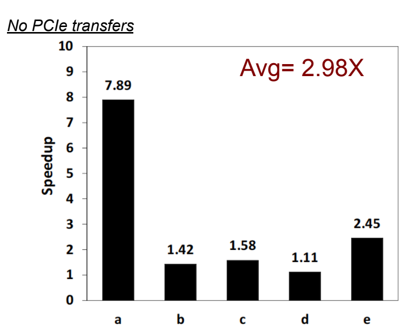
Joint with LogicBlox Inc.

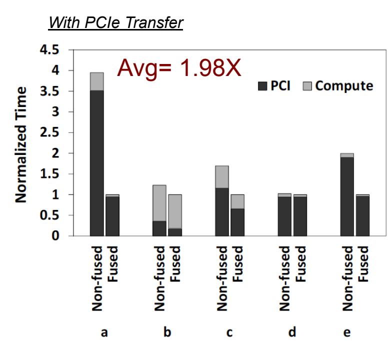


- Targeting Accelerator Clouds for meeting the demands of data warehousing applications
- In-core databases

Time and Space Gains

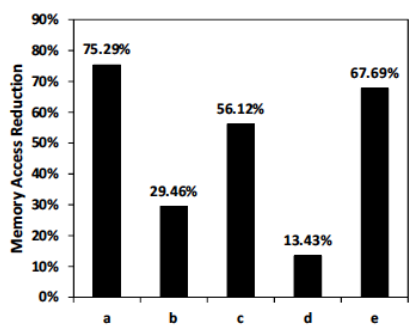




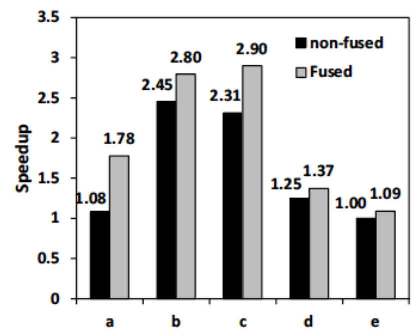


Memory and Optimization Scope

Memory Accesses

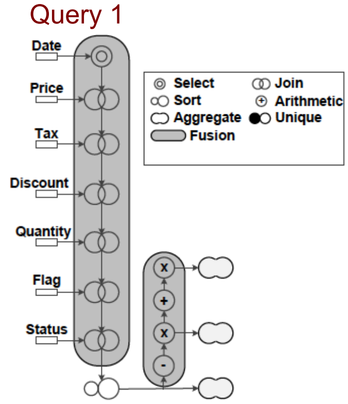


Impact of Optimization Scope – O3 vs. O0



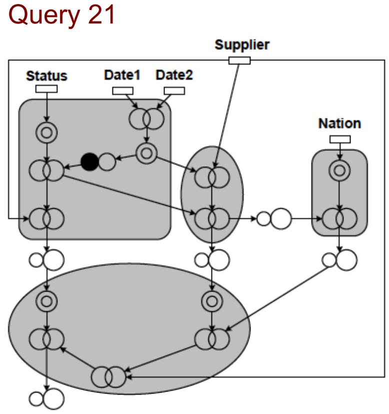
TPC-H Queries

Query 1



(a)

Query 21

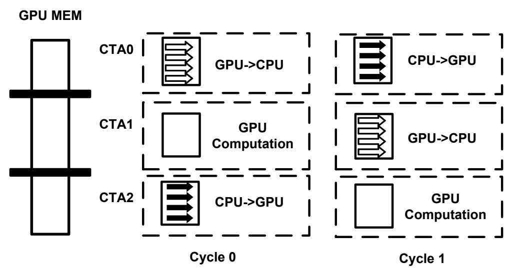


(b)

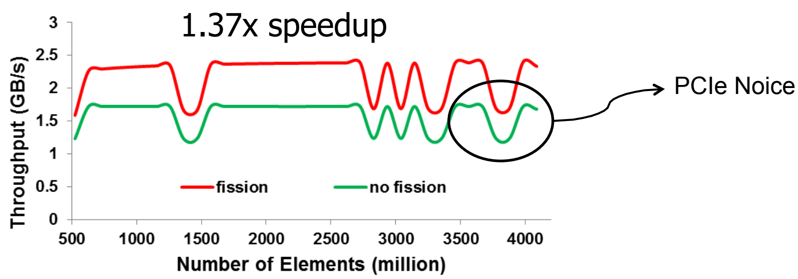
Avg= 1.25X (3.18X w/o SORT and PCIe)

Avg= 1.22X

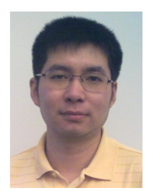
Example of Kernel Fission



Reminiscent of Software Pipelining



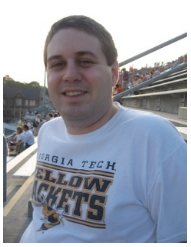
People



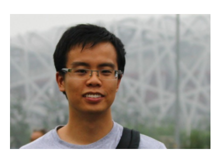
Gregory Diamos
Dynamic optimizations (Harmony, Ocelot, LLVM Bridge)

Andrew Kerr
Program Transformations & Optimizations for Data Parallel Computation (Ocelot, LLVM Bridge, VSIPL)

Haicheng Wu
Dynamic Opytimizations (Ocelot)

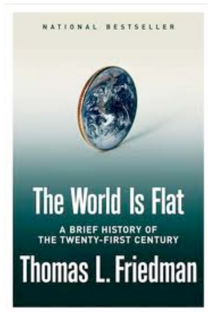


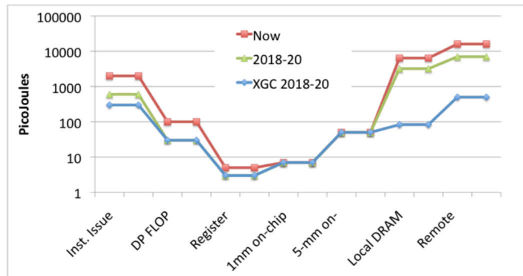
Jeff Young
Integrated Networks-Memory, Oncilla



Si Li
Correctness & Emulation Tools, GP architectures

Summary





- Refactor cluster architectures for
- Flexible hardware composition of resources and
- Migrate to a communication-centric model of algorithms, systems, and optimizations