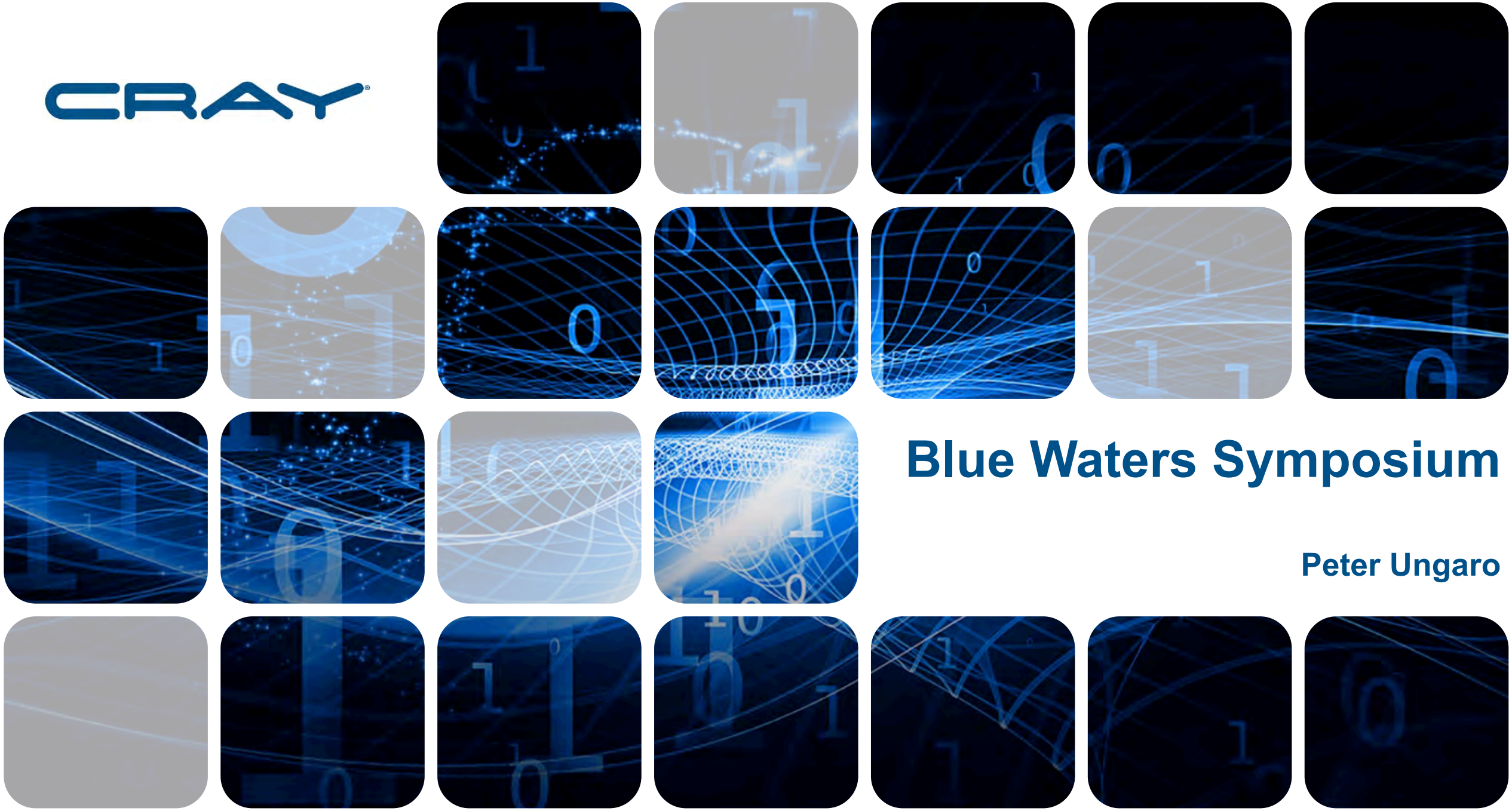


Blue Waters Symposium

Peter Ungaro

Change
Same

3 THINGS THAT HAVEN'T CHANGED

01 Our Commitment to Innovation

Partnership with Our Customers 02

03 Unwavering Customer Support

"The future is seldom the same as the past"

Seymour Cray
June 4, 1995

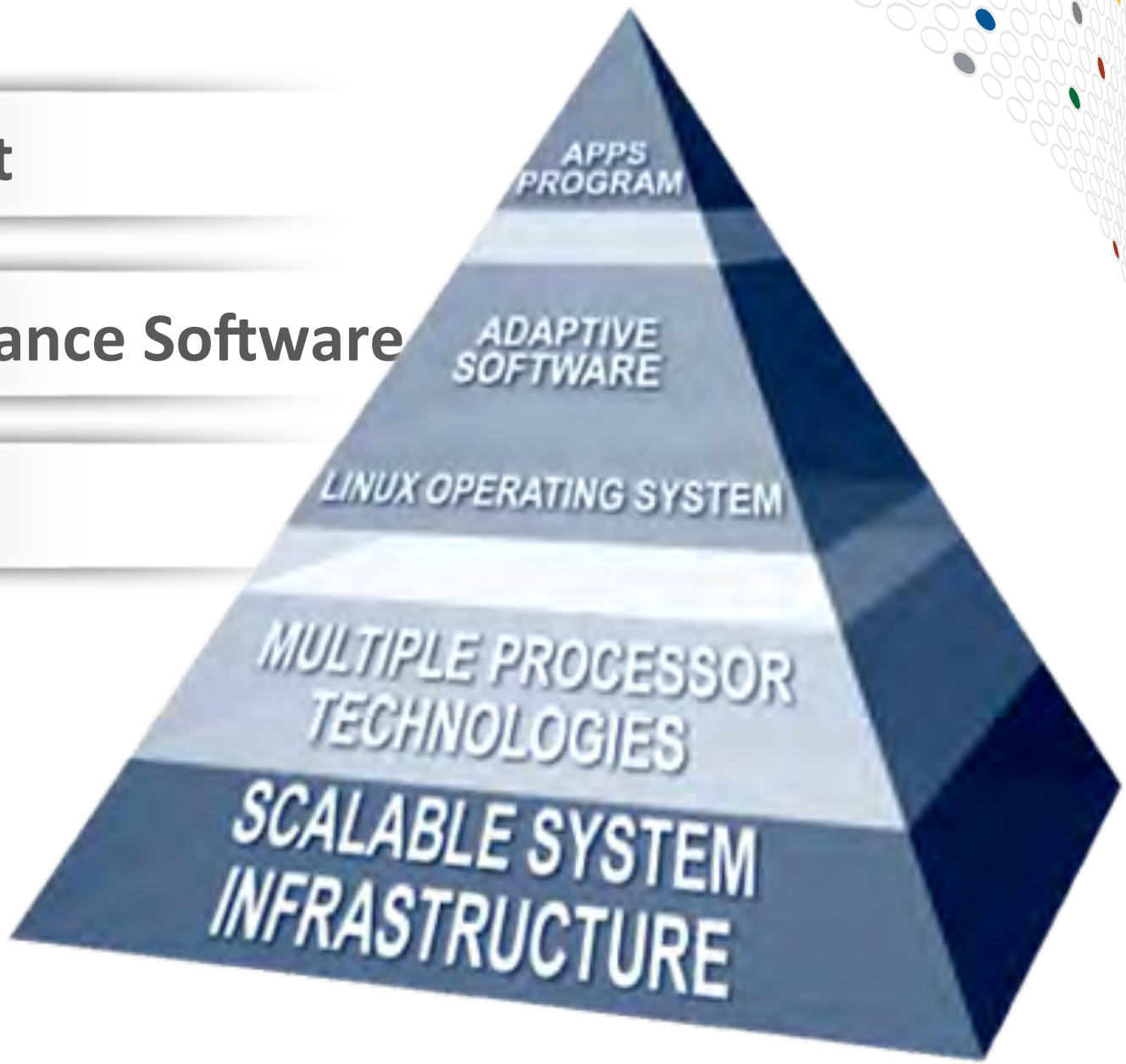
CRAY
THE SUPERCOMPUTER COMPANY

Commitment To Innovation…

System Interconnect

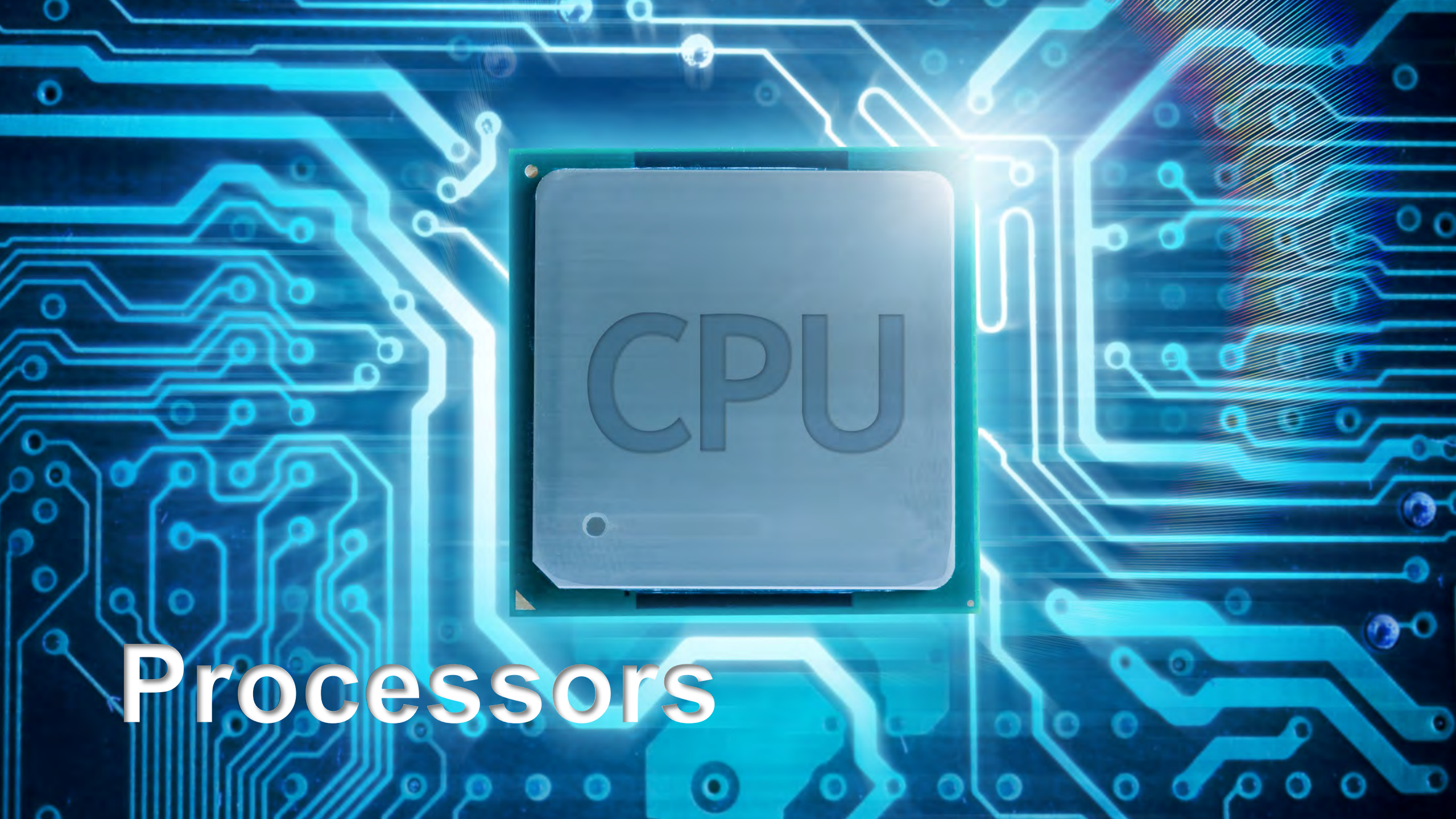
Systems Management & Performance Software

Packaging



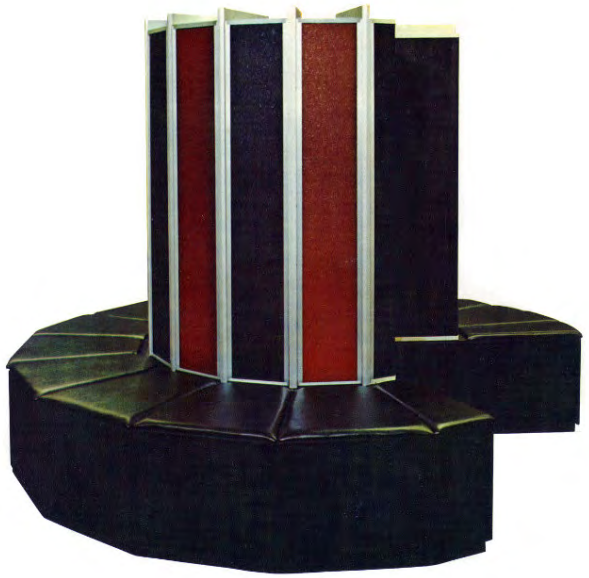
So What Has Changed?

- It has been 40 years since the first Cray-1 Supercomputer shipped
- Optimal design has always been dependent on underlying technologies
  - Processors & Memories
  - Storage
  - Interconnects
- Shifts in these technologies can (and will) have large impacts on how systems look and how they are programmed
- Workloads and Users are also changing…

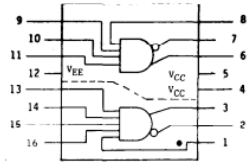


Processors

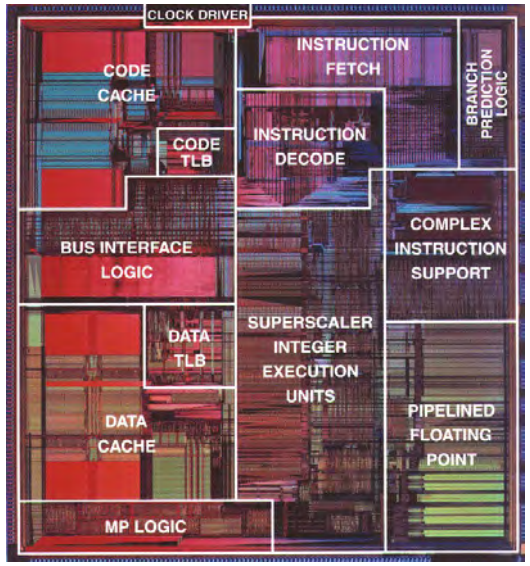
Technology Drives Architecture



Cray 1, 1976

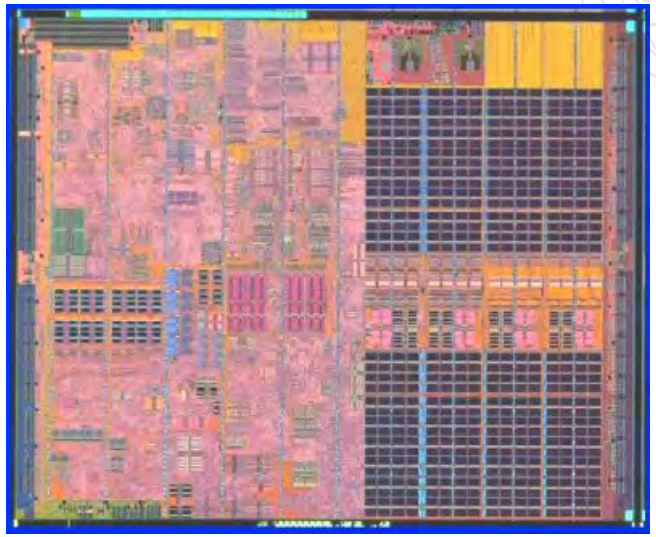


- ECL 5/4 NAND gate ICs (95%)
- 75K gates. (3400 PCBs!)
- RISC design
- Vector ISA
- Memory latency 11 clocks



Intel Pentium, 1993

- CMOS VLSI IC
- 3M transistors
- CISC design
- Scalar ISA
- Deep pipelines, complex predictions



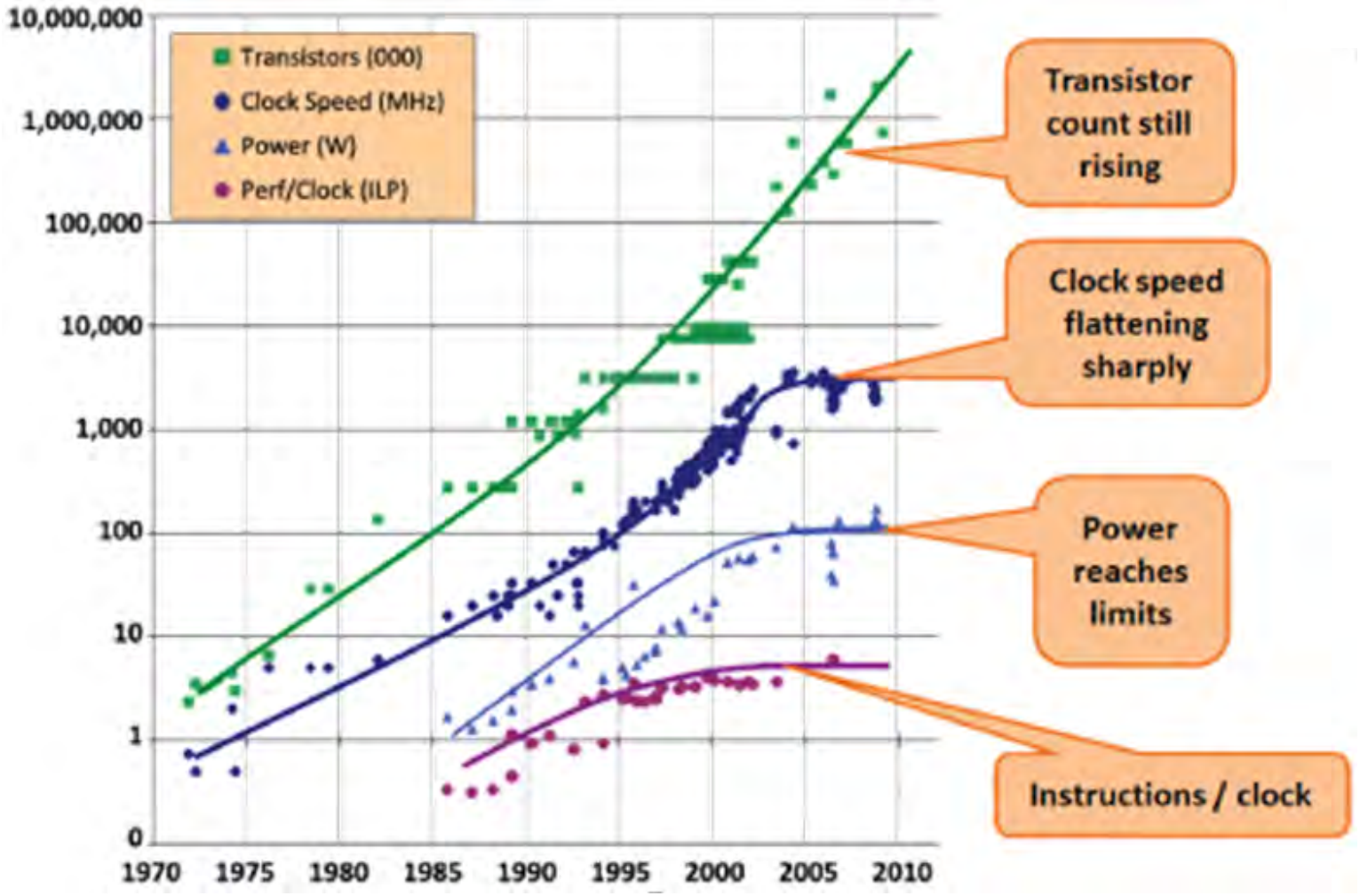
Intel Pentium 4 Cedar Mill, 2006

- 184M transistors!
- *Very* CISC design
- 31-stage pipeline
- 3.6 GHz in 65nm
- Last of its breed….

Now Constrained by *Power*

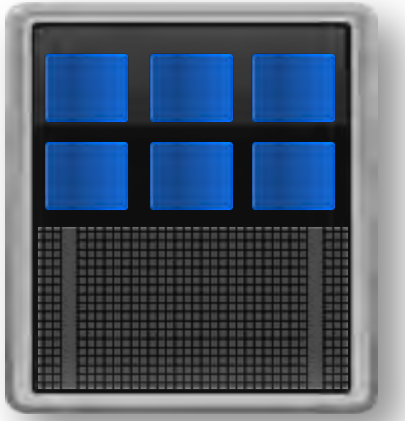
Performance/watt no longer keeping up with Moore's Law

Single Core Performance Scaling is Over

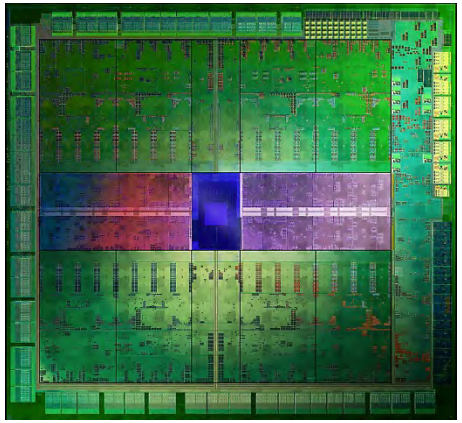


http://www.extremetech.com/computing/116561-the-death-of-cpu-scaling-from-one-core-to-many-and-why-were-still-stuck

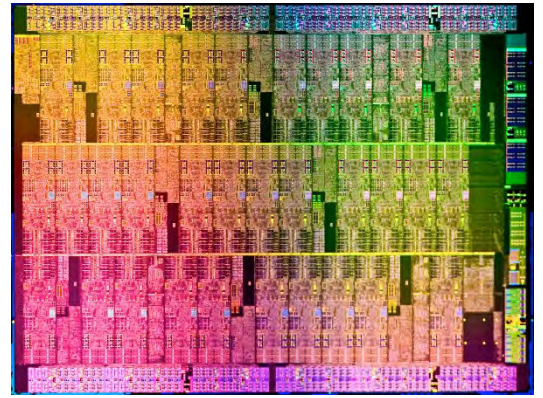
Architectural Response



Multicore
Multiple, simpler processors



GPU computing (NVIDIA Tesla)
Large # of *much* simpler processors

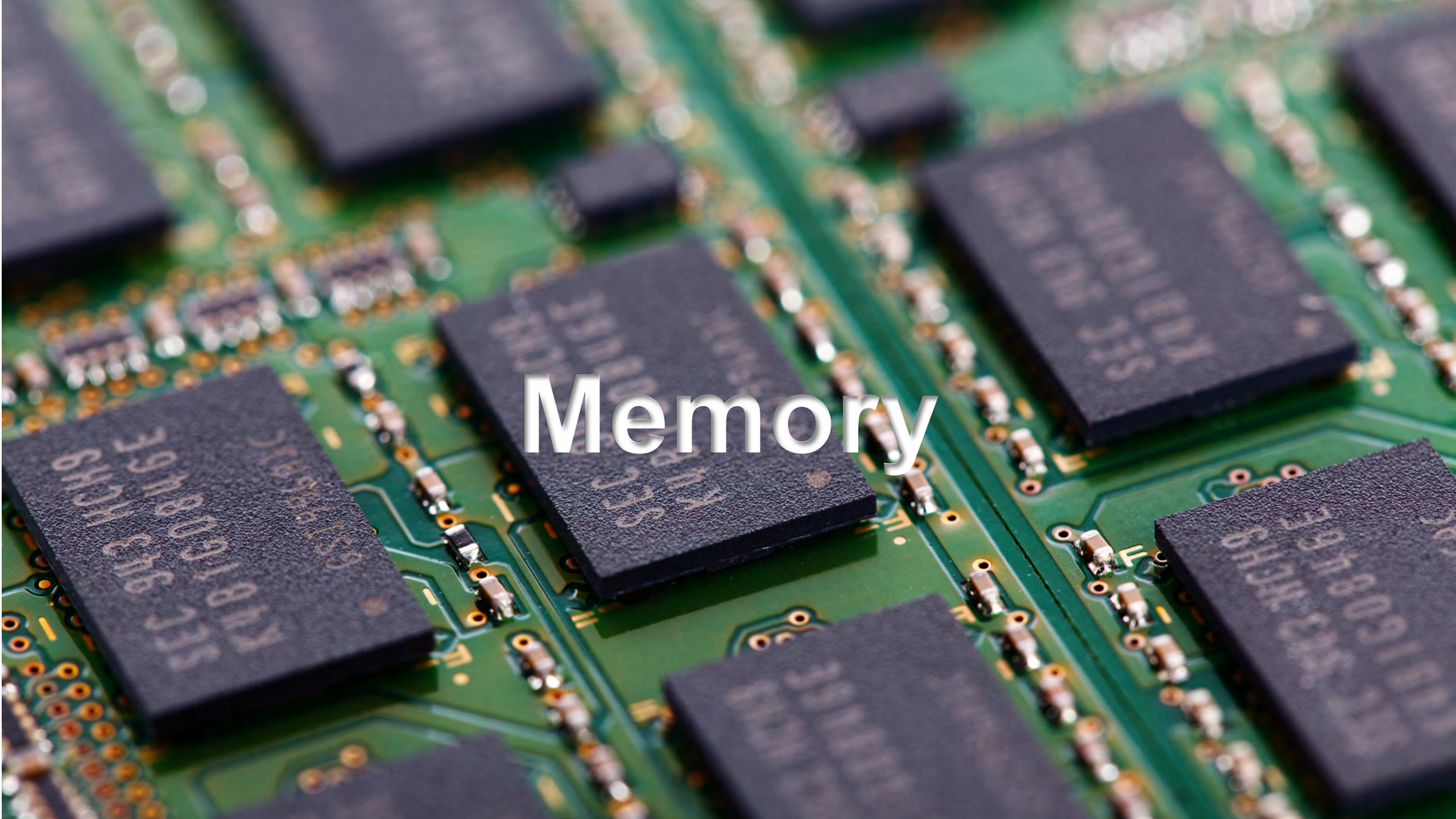


Vector Computing (Intel Xeon Phi)
Parallelism with low complexity & control overhead

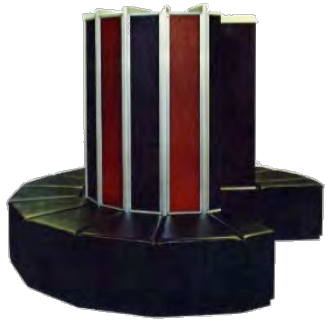
Our Approach to Processor Choice

- **Work closely with multiple processor vendors to understand roadmaps**

- **Evaluate options based on:**
  - Predicted sustained price-performance
  - Programming models and productivity
  - Physical enablement (power, cooling, packaging)
  - Customer feedback
- **Work with processor partners to impact the designs**
  - Feedback on features and node architecture for performance and programmability
  - Optimize software stack for that processor
  - Cray interconnect enablement
- **Build a great system around whatever processors we use**


Memory

Memories Through the Decades



1980s

Small, Fast Memories

- Out-of-core programming
- Single Core
- Vectors

1990s

Large, DRAM-based memory, local memory

- Convert to in-core algorithms
- Still single-core
- Vectors

2000s

Distributed DDR Memory - caches

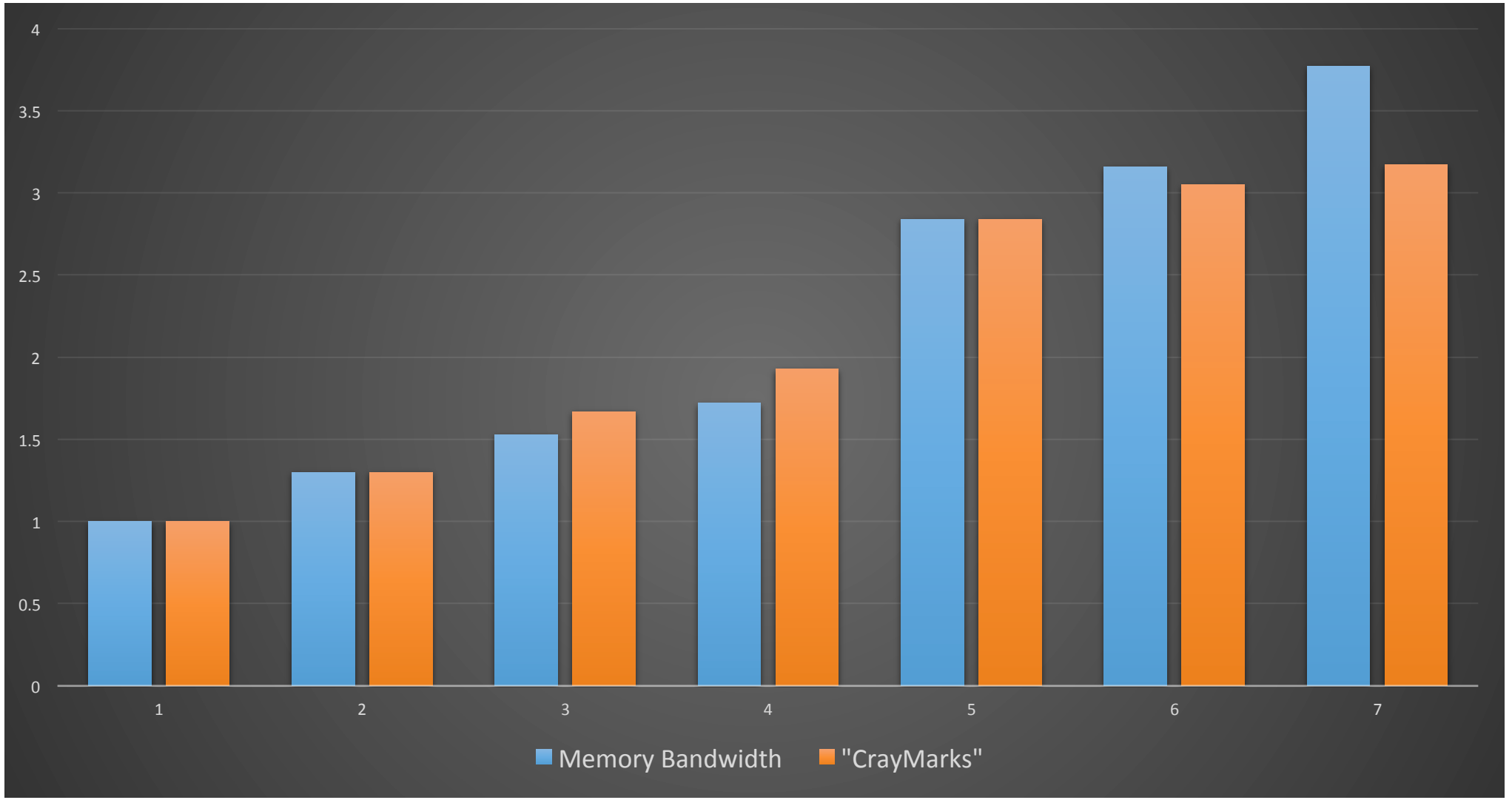
- Big Disruption! Convert to MPI
- Vectorization not so important

2010s

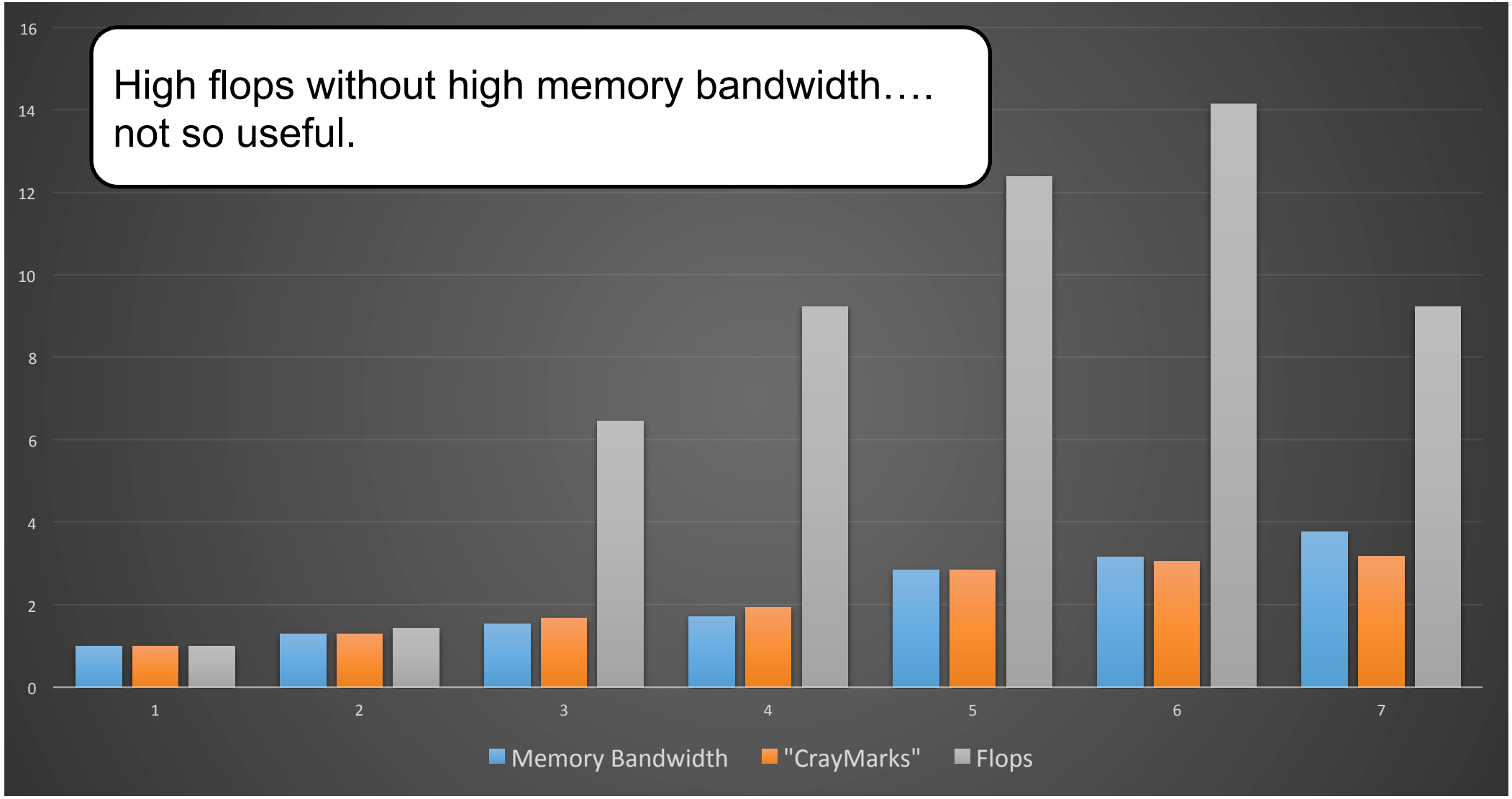
More cores per node- Some “hybrid” nodes with fast memories

- Some conversion to CUDA
- Some hybrid programming
- X86 “Vectors”

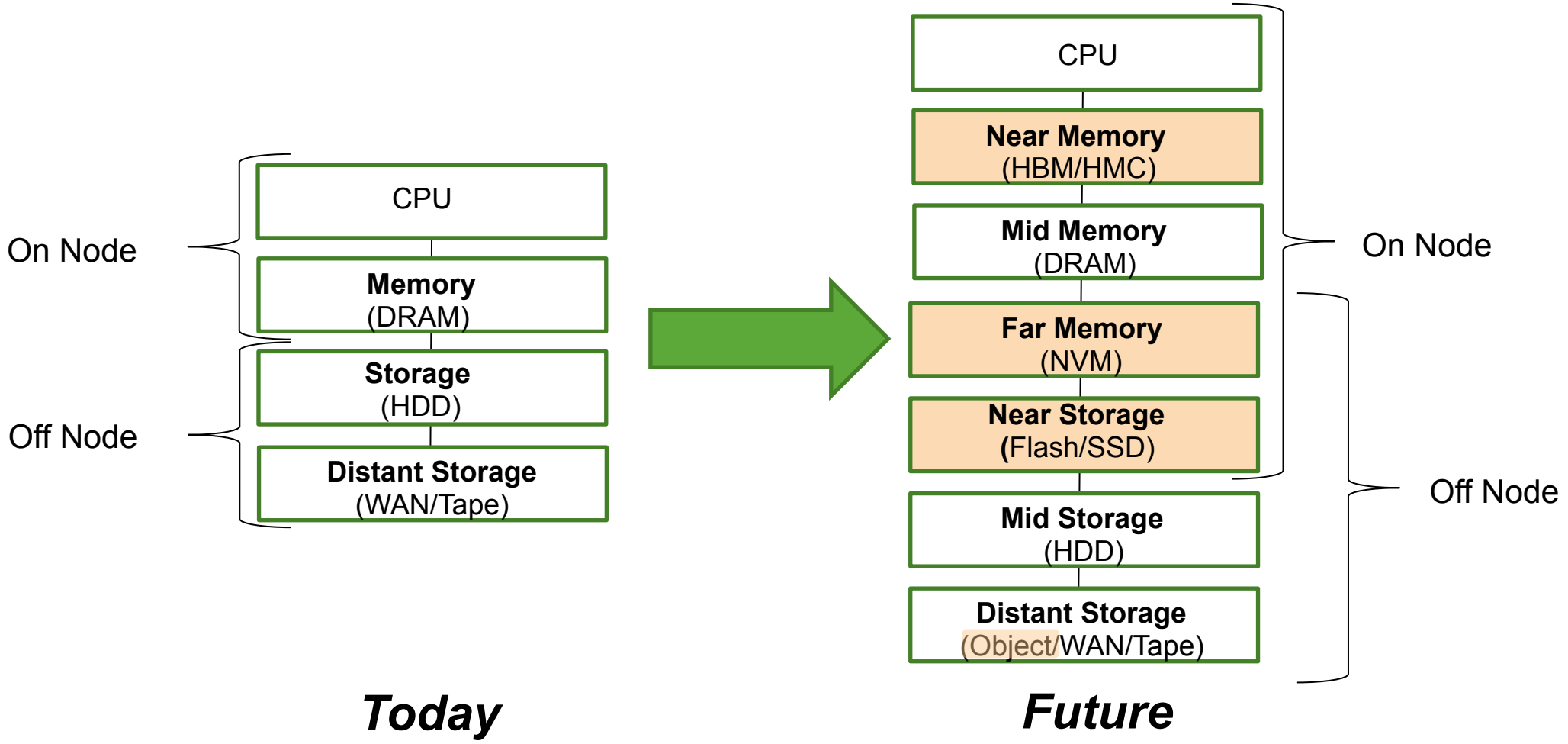
Application Performance Correlates with Bandwidth



…Not FLOPS

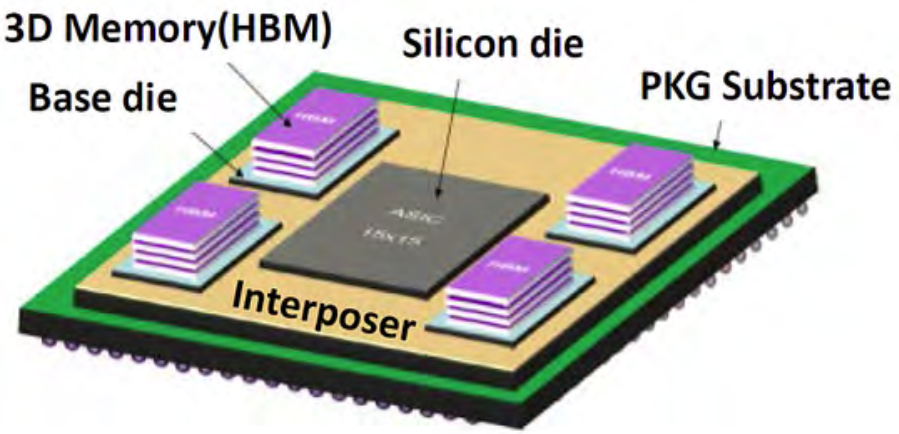

High flops without high memory bandwidth…. not so useful.
16
14
12
10
8
6
4
2
0
1
2
3
4
5
6
7
Memory Bandwidth
"CrayMarks"
Flops

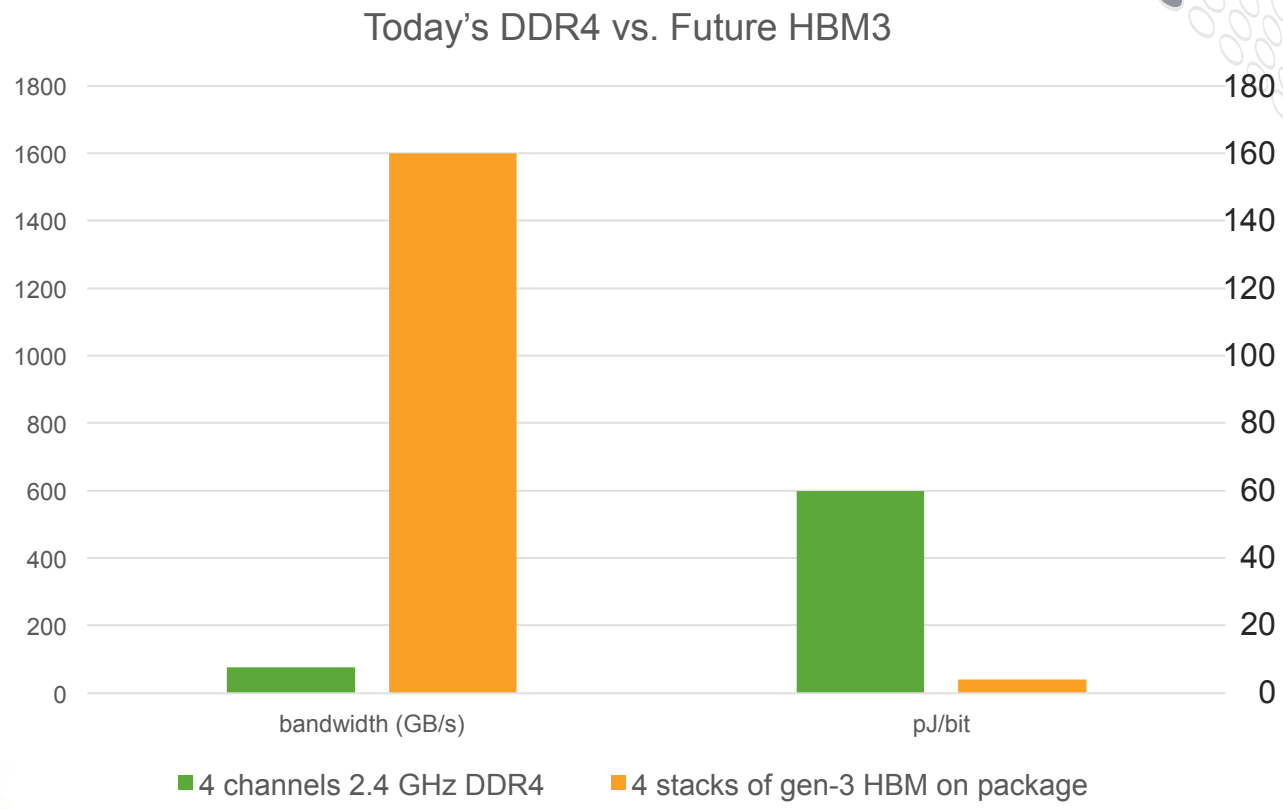
Deeper Memory/Storage Hierarchy



On-Package Memory Can Restore Balance

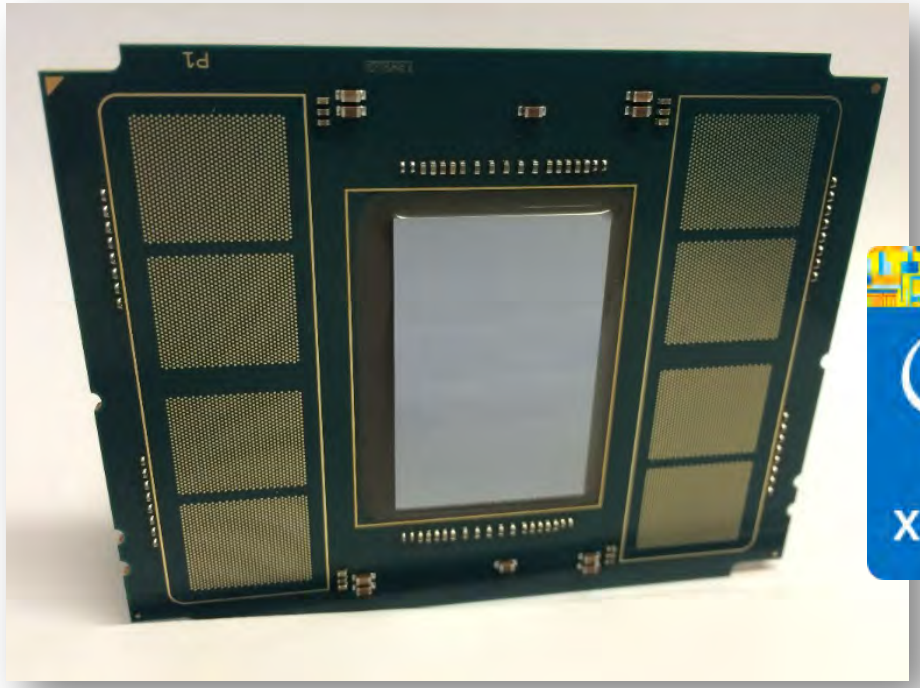
- Standard DDR memory BW has not kept pace with CPUs
- Expect processors to adopt stacked, on-package memory
- HBM:
  - 10x higher BW, 10x less energy/bit
  - Much lower latency
  - Costs ~2x DDR4 per bit
  - JDEC standard with multiple sources



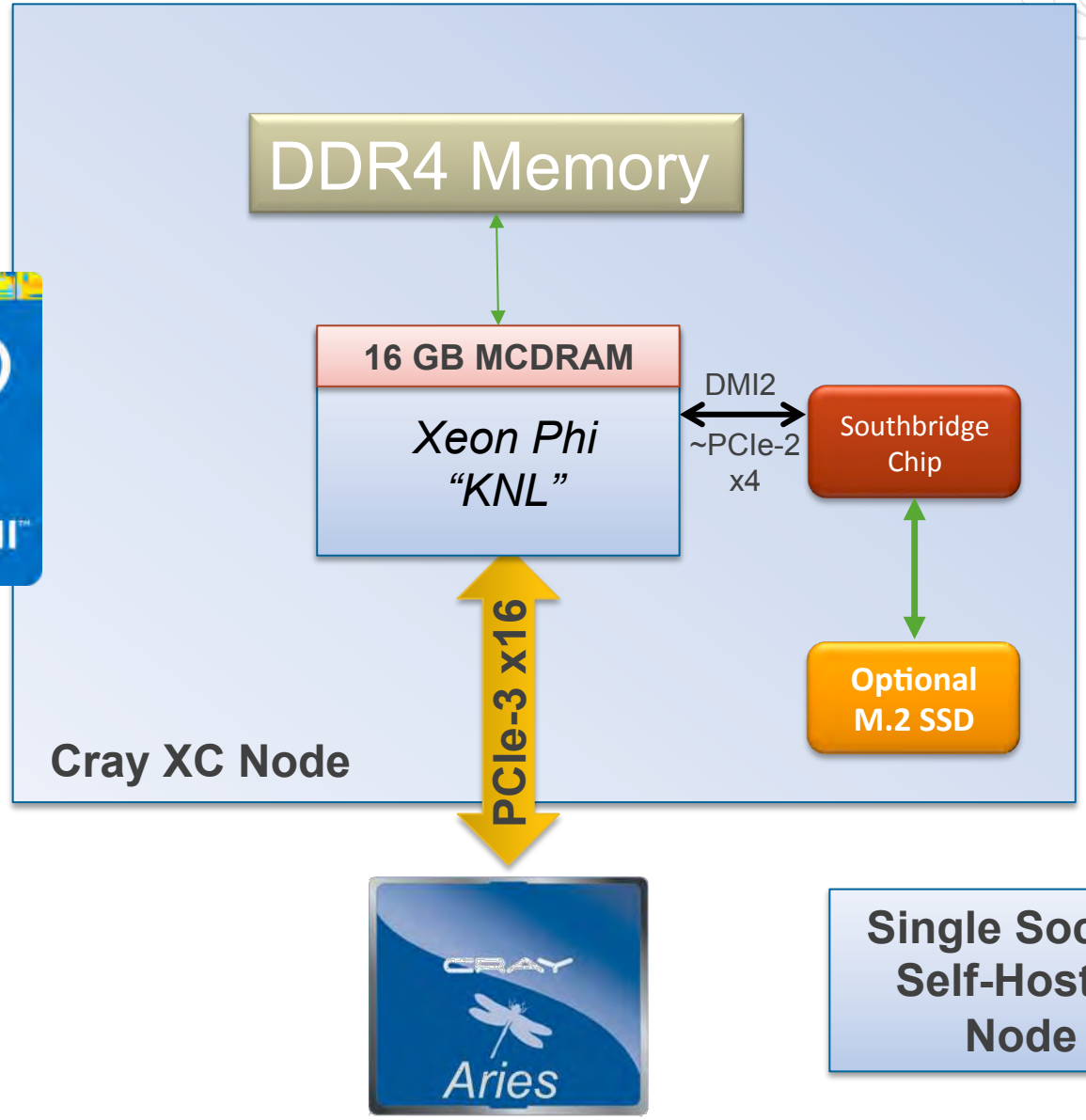


May drive us to smaller, simpler nodes that are balanced with on-package memory

Xeon Phi KNL Node - 2016



- >60 Cores
- >3 Tflops
- High Bandwidth Memory (@ ~5X DRAM BW)
- Flexible Memory Modes



Storage

Storage Technologies

- For many years, rotating storage has been utilized for “high speed” storage

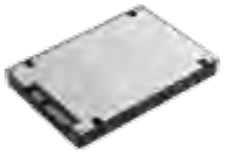
- Striping and parallel file system allowed us to increase bandwidth far beyond a single spindle

lustre

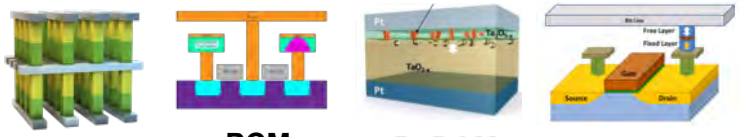
Storage Trends

- **Flash becoming cheap enough to hold primary scratch file system**
  - *Much* better {bandwidth, IOPS}/$ than disk
  - Media access can no longer hide high software overhead ⇒ need leaner software stack

Disk is the new tape.



Flash is the new disk.



New technologies to bridge memory-Flash gap

- **Software defined storage enabling reliable, inexpensive storage**

amazon web services™ S3

Google Cloud Storage

Windows Azure Blob

openstack™

ceph

- **Desire for new ways to organize and interact with data**

First Step: Flash Front-End for Storage

- **Accelerate Application and I/O Performance**
  - 5x the bandwidth of disk at the same cost
- **Flexible Usage Models**
  - Burst buffers for fast check-pointing
  - Local and Shared I/O models
  - Implicit file system caching
  - Accelerate multi-job pipelines
  - No application changes required

CRAY DATAWARP™

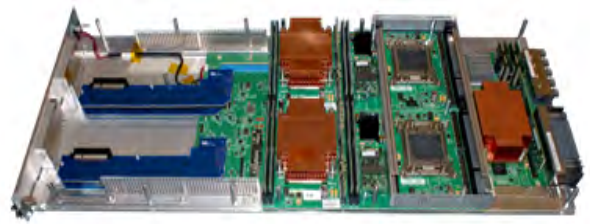
Overcomes the performance gap between compute and disk storage

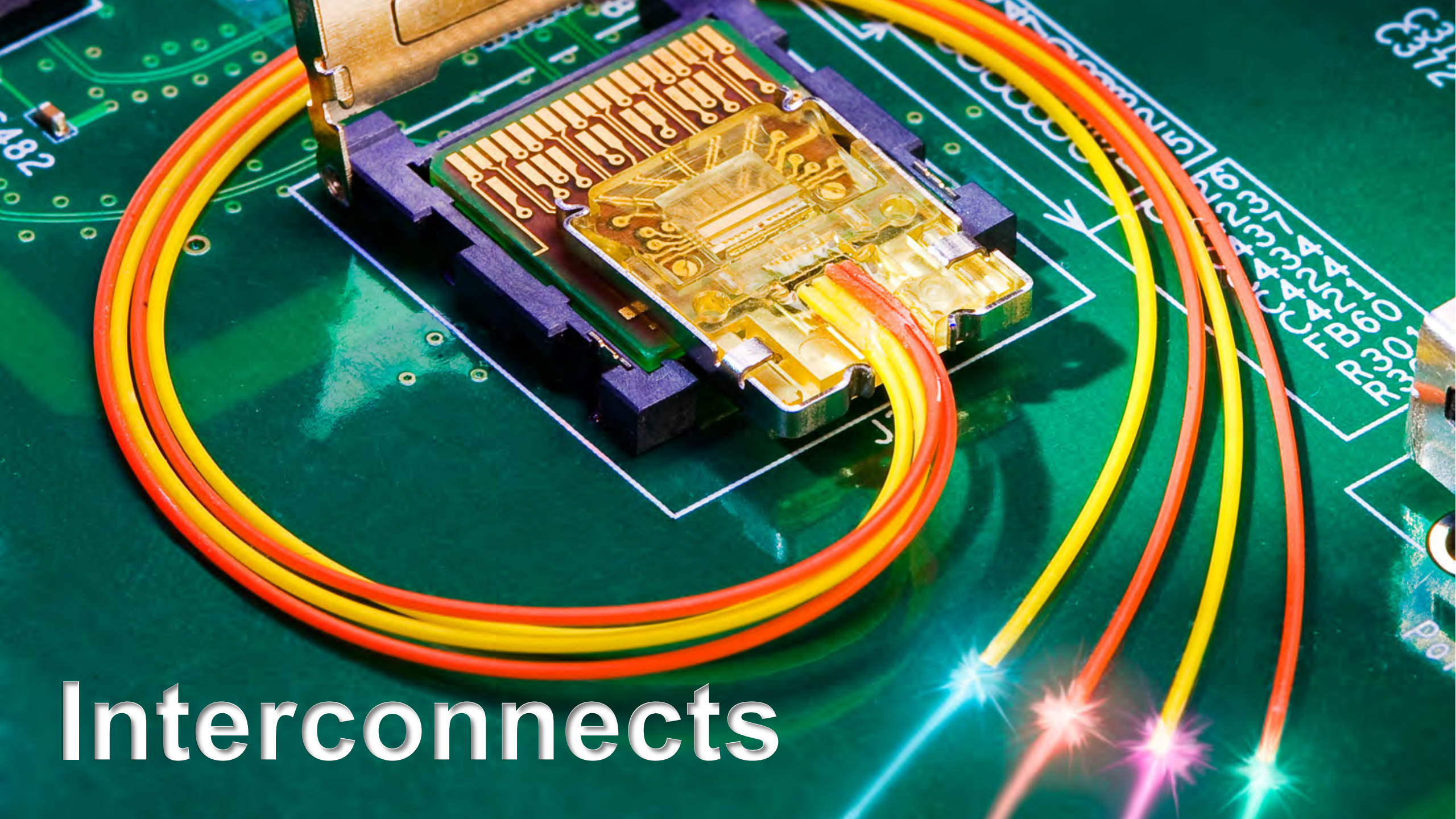
12 Million Random 4K IOPS!

140 DataWarp Nodes
4k random writes and reads
4,480 1GB Files

Storage Architecture Exploration at Cray

- **Extending DataWarp to become resilient *primary* storage**
- **Reducing cost and complexity by using simple storage devices**
  - Object storage, software defined storage
- **Reducing software overhead**
  - Efficient client access to data over the interconnect
- **Improving scalability**
  - Relaxing POSIX global consistency
  - Scalable metadata via NoSQL with collections and arbitrary relationships
- **Accommodating new ways of organizing data**
  - Spark RDDs, key/value stores, NoSQL databases, etc.
- **Enhanced telemetry and analytics**

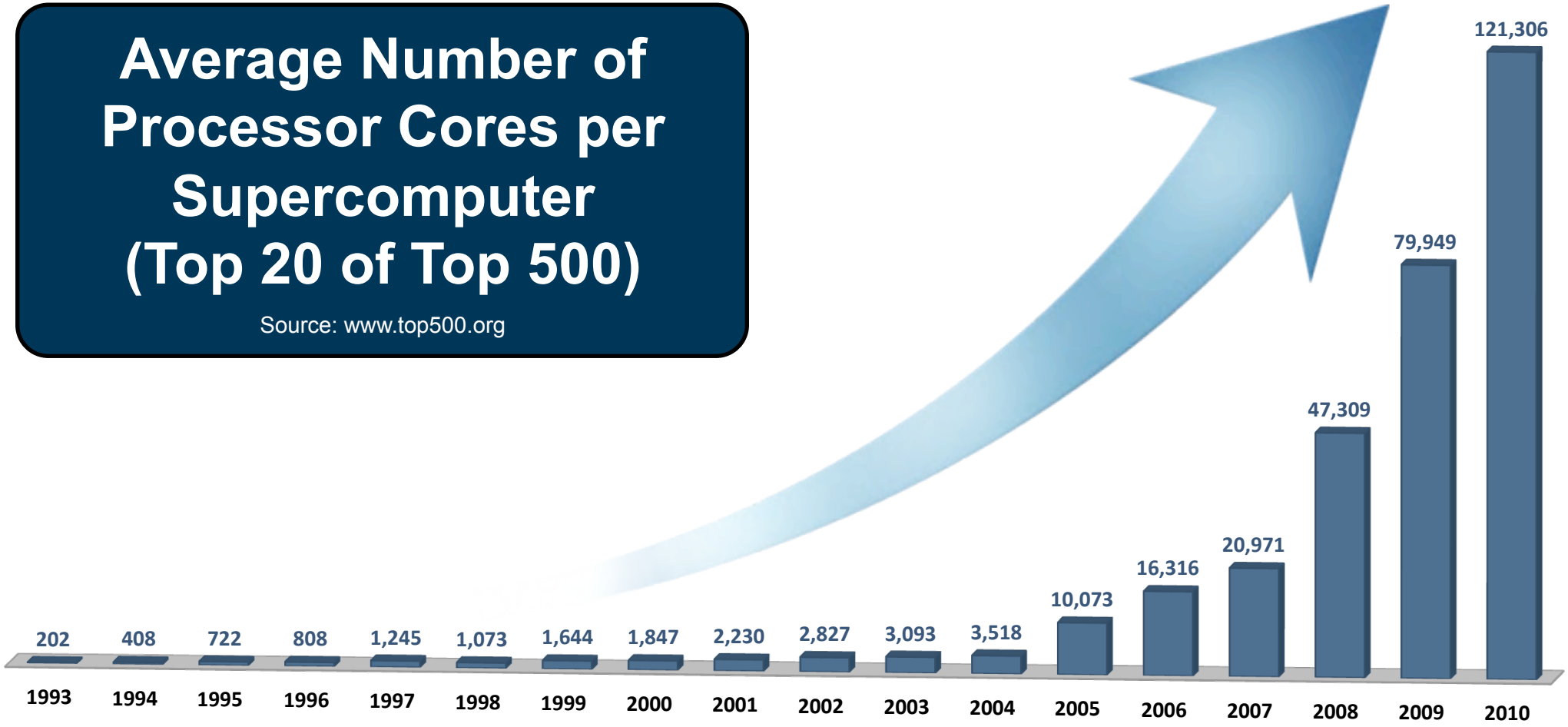



Interconnects

Interconnects Matter More and More…

Average Number of Processor Cores per Supercomputer (Top 20 of Top 500)

Source: www.top500.org



Cray Interconnects Over Time

1994

- **Cray T3E**
  - 3D Torus
  - E-Registers
  - PVM, Shmem and later MPI
  - Copper Cables

2003

- **Cray XT3 - SeaStar**
  - 3D Torus
  - Scalability – Connectionless protocol
  - Backplanes & Copper Cables

Cray Interconnects Over Time (continued...)

2010

- **Cray XE6 – Gemini**
  - Gemini Router
  - Still a 3D Torus!
  - Fast Message Rates
  - Resilience
  - Backplanes & Copper Cables

2013

- **Cray XC30 - Aries**
  - **Dragonfly – 5 Hop**
  - Global Bandwidth increase
  - Adaptive Routing
  - Hardware Collectives
  - Link Level Reliability
  - Backplanes, Copper and Active Optics

768 Sockets less than 1µs away

2-cabinet Group

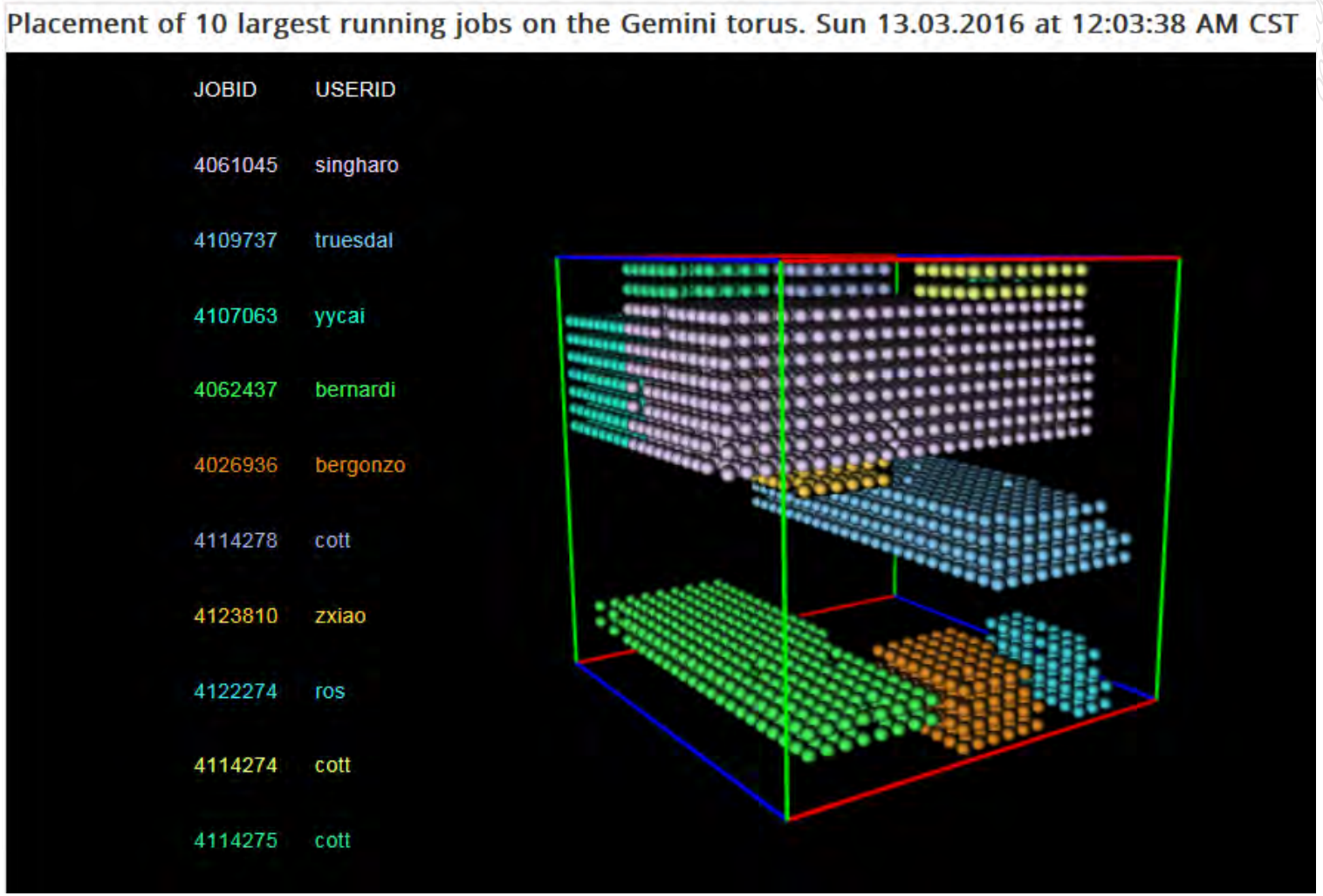
Backplane connections within chassis

Copper cables between chassis

This basic structure is repeated in large systems with optical connections

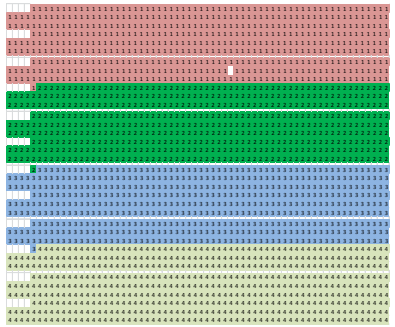
Dealing with Job Placement

- Much of this work done in partnership with NCSA on the Blue Waters system… where we found that placement really matters!
- Large jobs are scheduled in rectangles to minimize inter-job contention
- Works, but there is a trade-off with utilization

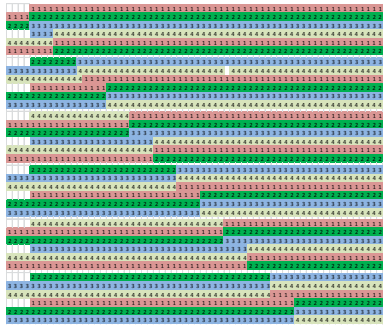


Dragonfly is placement insensitive

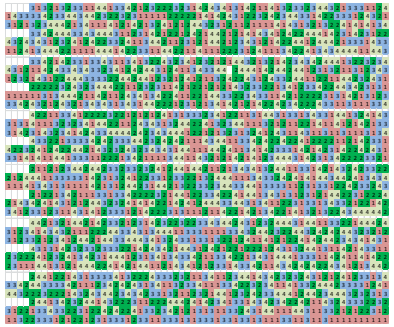
- Example: Sandia miniApp, miniGhost
- Running on 2256 node CSCS system (¼ global bandwidth)
  - Runtime in seconds for 100 cycles



Contiguous Blocks of 512 nodes			
69.0	68.8	68.9	68.9



Random blocks of 64 nodes			
69.4	69.4	69.4	69.5



Random layout of nodes			
70.9	71.0	70.6	70.5

Perfect Placement

< 3% variance from best-case to worst-case placement

Worst-Case Placement

Connectionless Protocol and Scalability

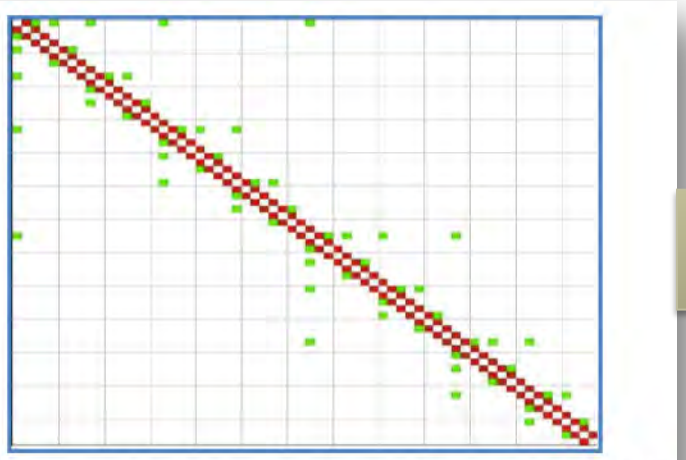


Figure 17. AMG inter-processor communication CrayPat plot

Nearest Neighbor

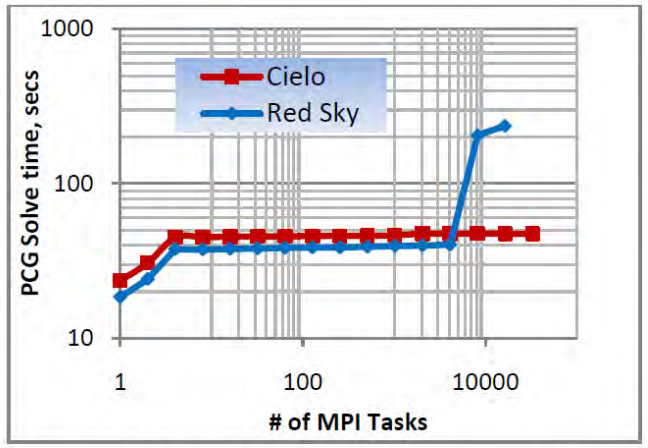


Figure 16. AMG Scaling Performance (Lower is better)

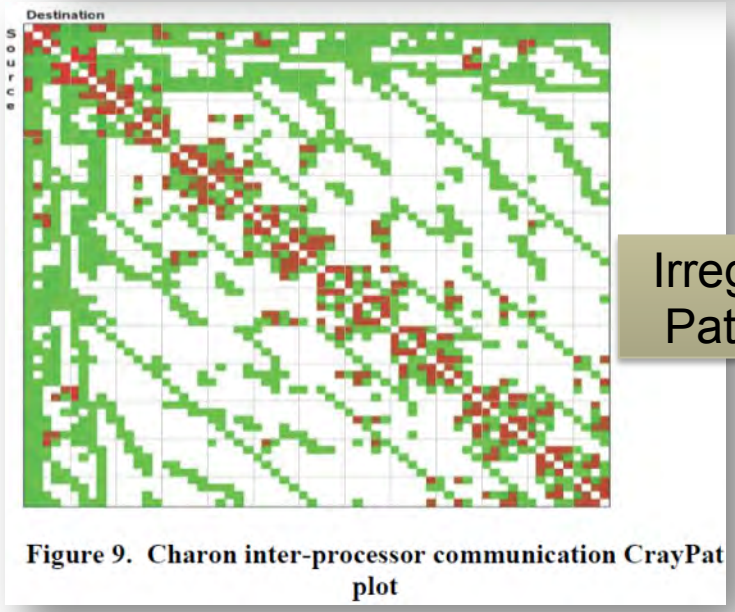


Figure 9. Charon inter-processor communication CrayPat plot

Irregular Pattern

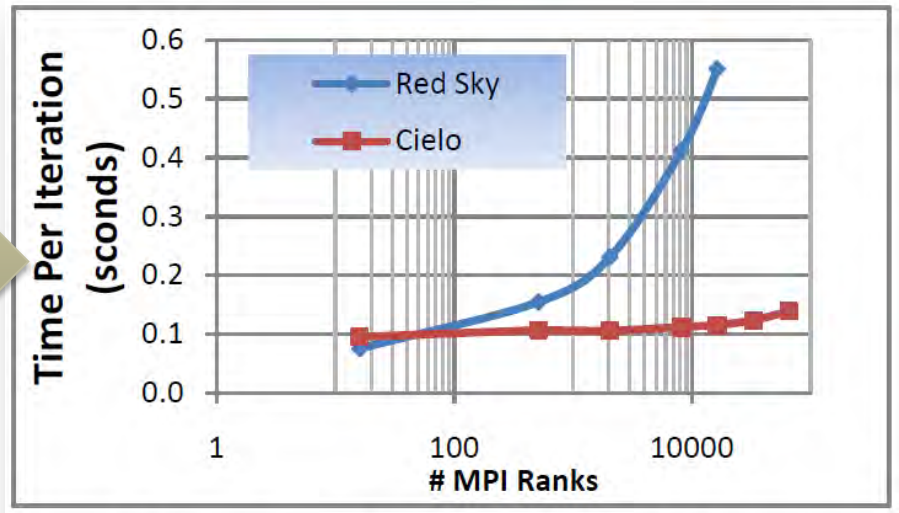


Figure 8. Charon scaling performance (Lower is better)

A Comparison of the Performance Characteristics of Capability and Capacity Class HPC Systems

By Douglas Doerfler, Mahesh Rajan, Marcus Epperson, Courtenay Vaughan, Kevin Pedretti, Richard Barrett, Brian Barrett , Sandia National Laboratories

Workloads

Diverse and growing set of users

Trend: Increasing varied workloads

Fusion of Supercomputing and Big (& Fast) Data

Converged Architecture for HPC and Data Analytics

CS400
Standards based cluster

Standard infrastructure

XC40
World's leading Supercomputer

Aries network

SONEXION
Scalable storage

High performance storage

Urika-GX

Open analytics framework

Cray Graph Engine

Urika-XA
Hadoop, Spark, NoSQL

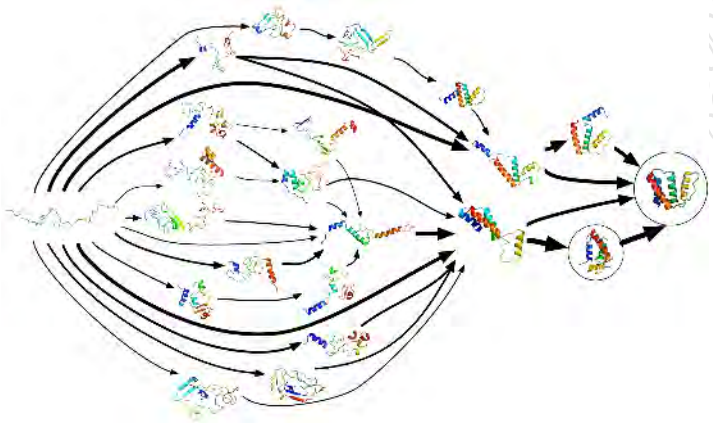
Urika-GD
World's best Graph Analytics engine

From Converged Systems to a Single Supercomputer

Urika-GX

Analytics environment

"Minerva"

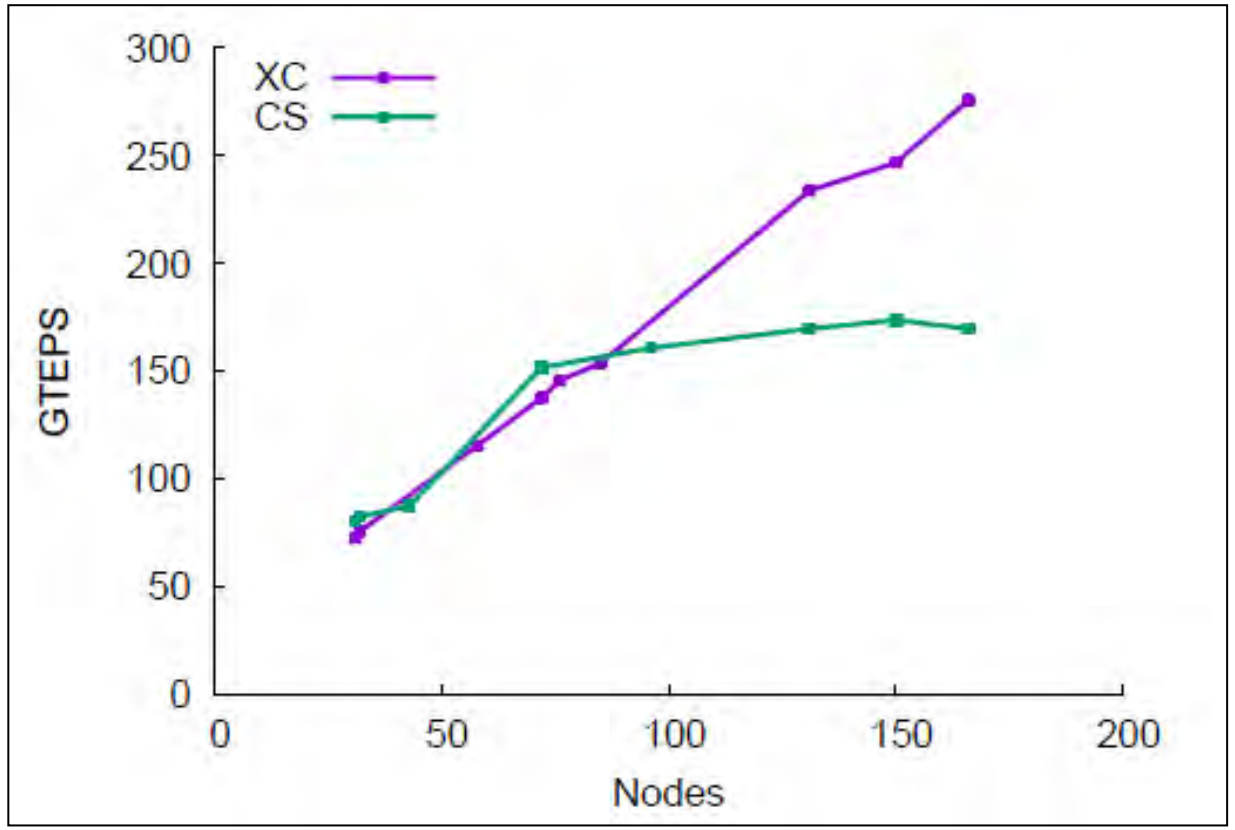


HPC + Analytics workflows

HPC underneath the covers

Why combine HPC and Analytics solutions in a single box?

Combining Analytics and Simulation on a Single Supercomputer @ ICM



Graph500 Results – Aries (XC40) compared with FDR (CS400)

The Interdisciplinary Centre for Mathematical and Computational Modelling (ICM) will run simulation and analytics workloads on their new Cray XC40

Scaling Spark on XC

"Overall, we feel optimistic about the performance of data analytics frameworks in HPC environments. Our results are directly translatable to others, e.g. Hadoop. Spark scales up to O(10,000) cores when running the problem in memory…"

Scaling Spark on HPC Systems

Nicholas Chaimov, Allen Malony
University of Oregon
{nchaimov,malony}@cs.uoregon.edu

Shane Canon, Khaled Z. Ibrahim, Costin Iancu, Jay Srinivasan
Lawrence Berkeley National Laboratory
{scanon,cciancu,kzibrahim,jsrinivasan}@lbl.gov

ABSTRACT

We report our experiences porting Spark to large production HPC systems. While Spark performance in a data center installation (with local disks) is dominated by the network, our results show that file system metadata access latency can dominate in a HPC installation using Lustre: it determines single node performance up to 4× slower than a typical workstation. We evaluate a combination of software techniques and hardware configurations designed to address this problem. For example, on the software side we develop a file pooling layer able to improve per node performance up to 2.8×. On the hardware side we evaluate a system with a large NVRAM buffer between compute nodes and the backend Lustre file system; this improves scaling at the expense of per-node performance. Overall, our results indicate that scalability is currently limited to $O(10^2)$ cores in a HPC installation with Lustre and default Spark. After careful configuration combined with our pooling we can scale up to $O(10^4)$. Higher scalability will require engineering work across application level libraries, as well as Spark runtime internals. As our analysis indicates, it is feasible to observe much higher scalability in the near future.

I. INTRODUCTION

Frameworks such as Hadoop [27] and Spark [29] provide a high level and productive programming interface for large scale data processing and analytics. Through specialized runtimes they attain good performance and resilience on data center systems for a robust ecosystem of application specific libraries [13, 20, 5]. This combination resulted in widespread adoption that continues to open new problem domains.

As multiple science fields have started to use analytics for filtering results between coupled simulations (e.g. materials science or climate) or extracting interesting features from high throughput observations (e.g. telescopes, particle accelerators), there exists plenty incentive for the deployment of the existing large scale data analytics tools on High Performance Computing systems. Yet, most solutions are ad-hoc and data center frameworks have not gained traction in our community. In this paper we report our experiences porting and scaling Spark on two current very large scale Cray XC systems (Edison and Cori), deployed in production at National Energy Research Scientific Computing Center (NERSC) [2].

In a distributed data center environment disk I/O is optimized for latency by using local disks and the network between nodes nodes is optimized primarily for bandwidth. In contrast, HPC systems use a global parallel file system with no local storage: disk I/O is optimized primarily for bandwidth, while the network is optimized for latency. Our initial expectation, was that after porting Spark to Cray, we can then couple large scale simulations using $O(10^4)$ cores, benchmark and start optimizing it to exploit the strengths of HPC hardware: low latency networking and tightly coupled global name spaces on disk and in memory.

We ported Spark to run on the Cray XC family in Extreme Scalability Mode (ESM) and started by calibrating single node performance when using the Lustre [7] global file system against that of an workstation with local SSDs: in this configuration a Cray node performed up to 4× slower than the workstation. Unlike clouds, where due to the presence of local disks Spark shuffle performance is dominated by the network [23], file system metadata performance is dominated by local I/O experts [19], the determining performance factor is the file system metadata latency (e.g. occurring in fopen), rather than the latency or bandwidth of read or write operations. We found the magnitude of this problem surprising, even at small scale. Scalability of Spark when using the back-end Lustre file system is limited to $O(10^2)$ cores.

After instrumenting Spark and the domain libraries evaluated (Spark SQL, GraphX), the conclusion was that a solution has to handle both high level domain libraries (e.g. Parquet data readers or application input stage) and the Spark internals. We calibrated single node performance, then we performed strong and weak scaling studies on both systems. We evaluate software techniques to alleviate the single node performance gap in the presence of a parallel file system:

- First and most obvious configuration is to use a local file system, in main memory or mounted to a single Lustre file, to handle the intermediate results generated during the computation. While this configuration does not handle the application level I/O, it improves performance during the Map and Reduce phases and a single Cray node can match the workstation performance. This configuration enables scaling up to 10,000 cores and beyond, for more details see Section 5.3.
- As the execution during both application initialization and inside Spark opens the same file multiple times, we explore "caching" solutions to eliminate file metadata operations. We developed a layer to intercept and cache file metadata operations at both levels. A single Cray node with pooling also matches workstation performance and overall we see scalability up to 1,000 cores. Combining pooling with local caching improves performance (up to 15

System Monitoring and Operational Analytics

Analytics Appliance

- Predictive failure analysis
- Job failure/performance diagnosis
- Cyber-threat detection
- System optimization
  - Power management
  - Job scheduling and placement
  - IO and network configuration
- Proactive detection of performance issues
- System dashboards

Currently collect logs in multiple places

- SMW, SDB & login nodes, Lustre service nodes

Types of data:

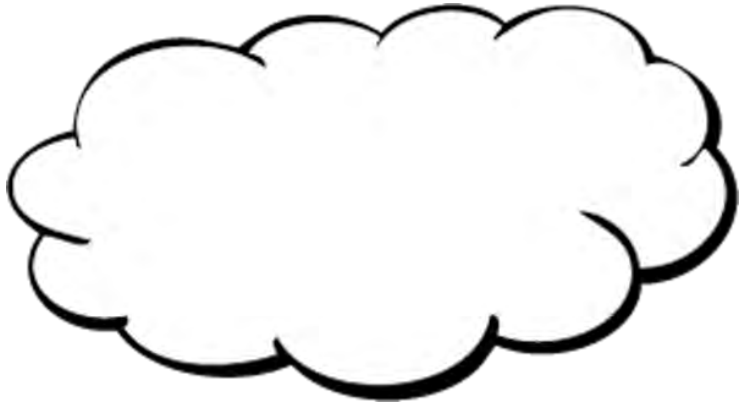
- Network health
- Console traffic (node-level OS errors)
- Temp, power, perf & status of all components
- Job scheduling and placement information
- Job performance data
- File system and network logs
- Etc.

Hard to diagnose performance problems or failures

- SSA is a first step…

What about the Cloud?

Forecast: The Future Will be Partly Cloudy

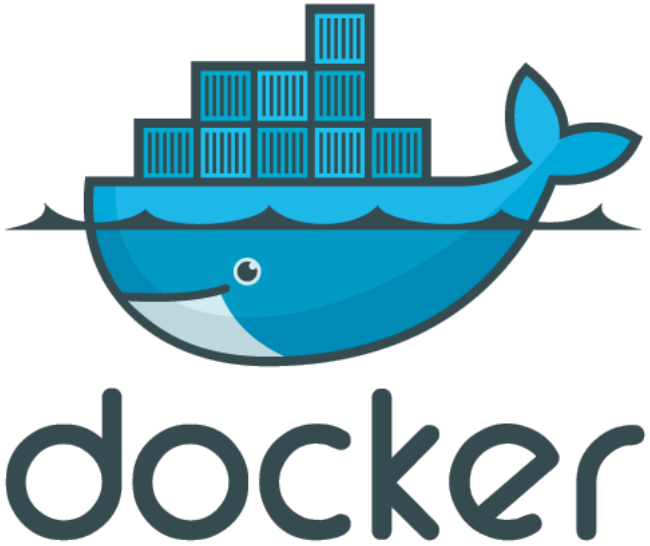


- Ease of use
- Elastic
- Leverage lots of software infrastructure
- Pay as you go
- Quick to deploy

- More expensive than (well utilized) in-house
- Need to move data off/on premise
- Not good for capability HPC

HPC can learn from the Cloud

Improving portability and ease of use



The Cray Shasta Supercomputer

Cray's Supercomputers Today

CS400

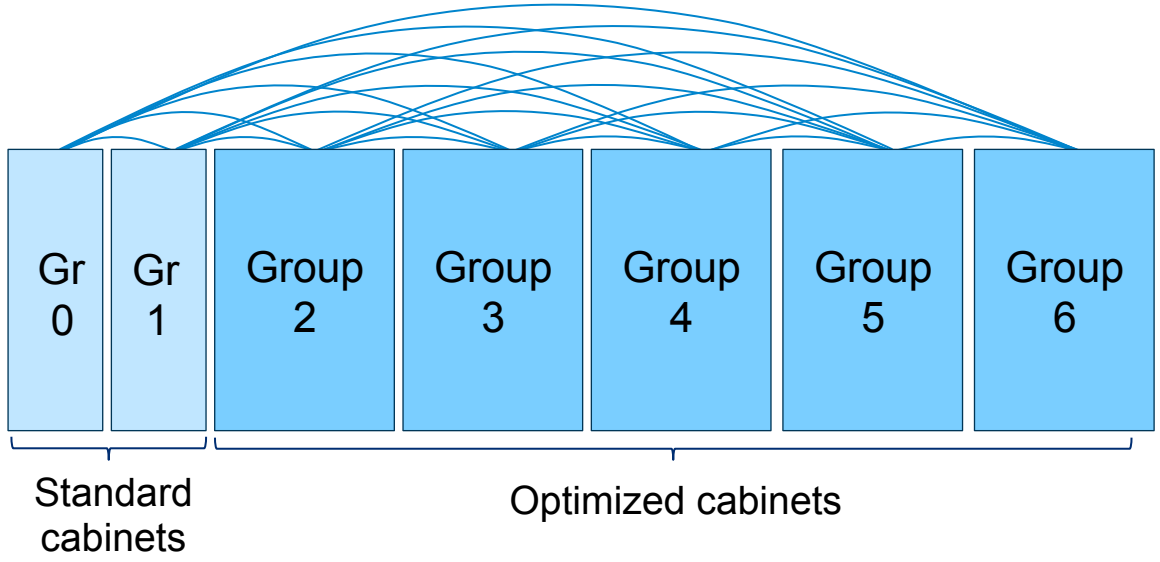
- Standard cabinet
- Standard software stack
- IB, Ethernet, or OPA networks
- Flexible, rack-mounted node options

XC40

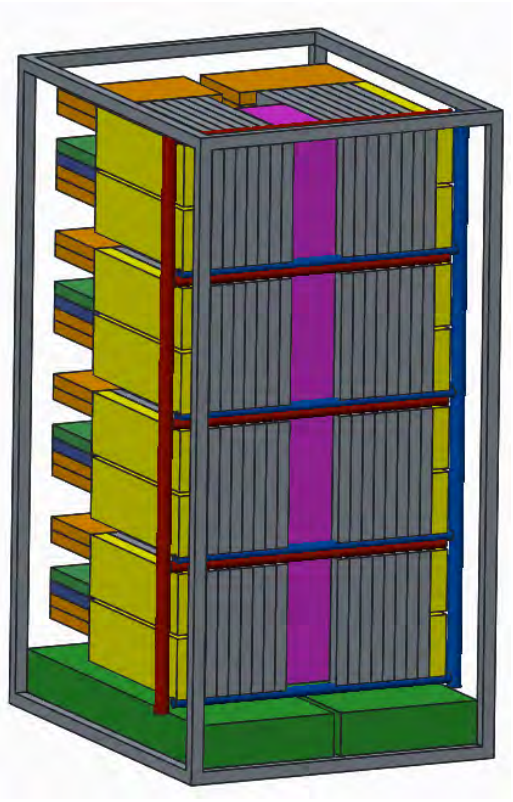
- Optimized, high-density cabinet
- Optimized, scalable software stack
- Aries interconnect
- Constrained board form factors

Cray Shasta System (2019)

- **Independent choices:**
  - Cabinet type
  - Power and cooling provisioning
  - Processor and node type
  - Software stack (incl. analytics)
  - Operational analytics
  - Interconnects



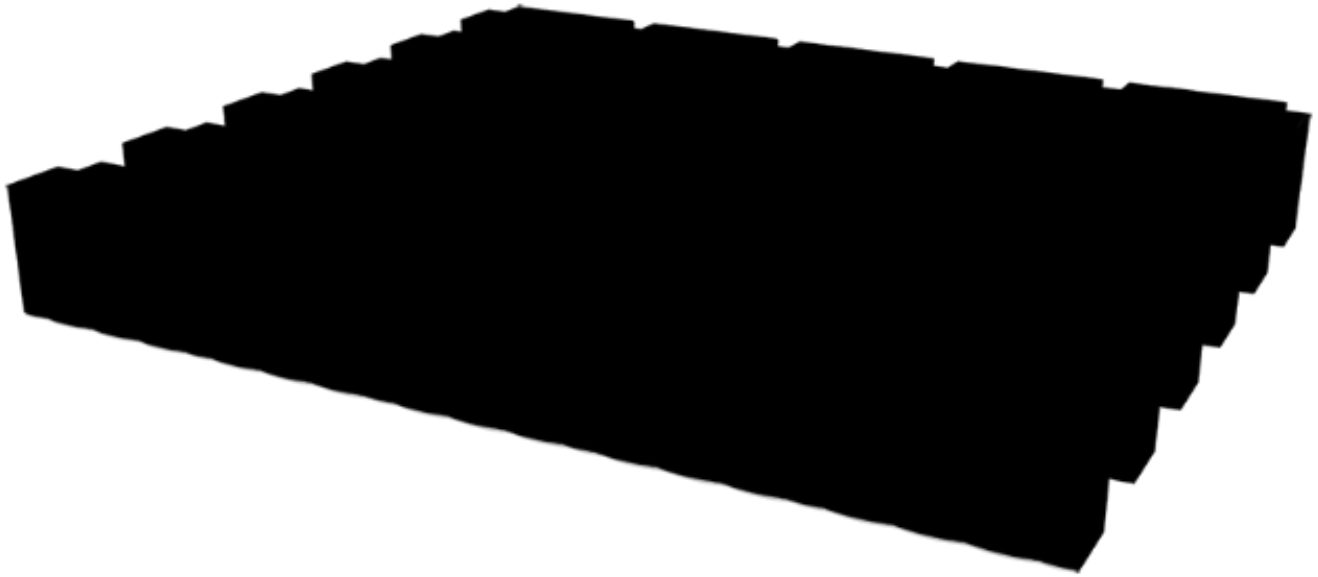
Standard rack



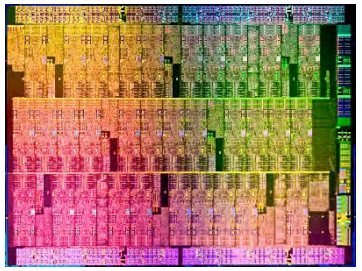
Optimized rack

Aurora – Our First Shasta System

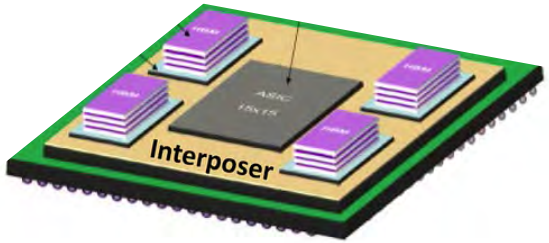
- **180 PF Peak Performance**
- **DOE/ANL ALCF (Argonne Leadership Computing Facility) Win based on Cray's:**
  - Scalable, adaptive supercomputing architecture collaboration and development with Intel
  - Targeting both supercomputing and analytics workloads



Exascale Will Be the End of the CMOS Era

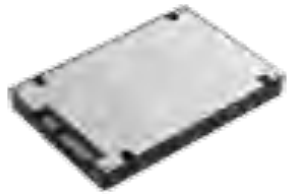


10-50 TF per node
Wide vectors (or GPUs)



Memory system
on package

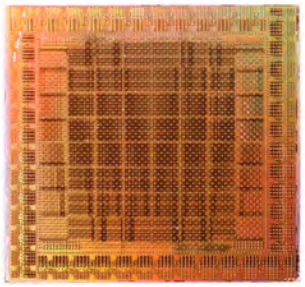
Disk for
archive



Flash for
main storage

NVM for storage
cache

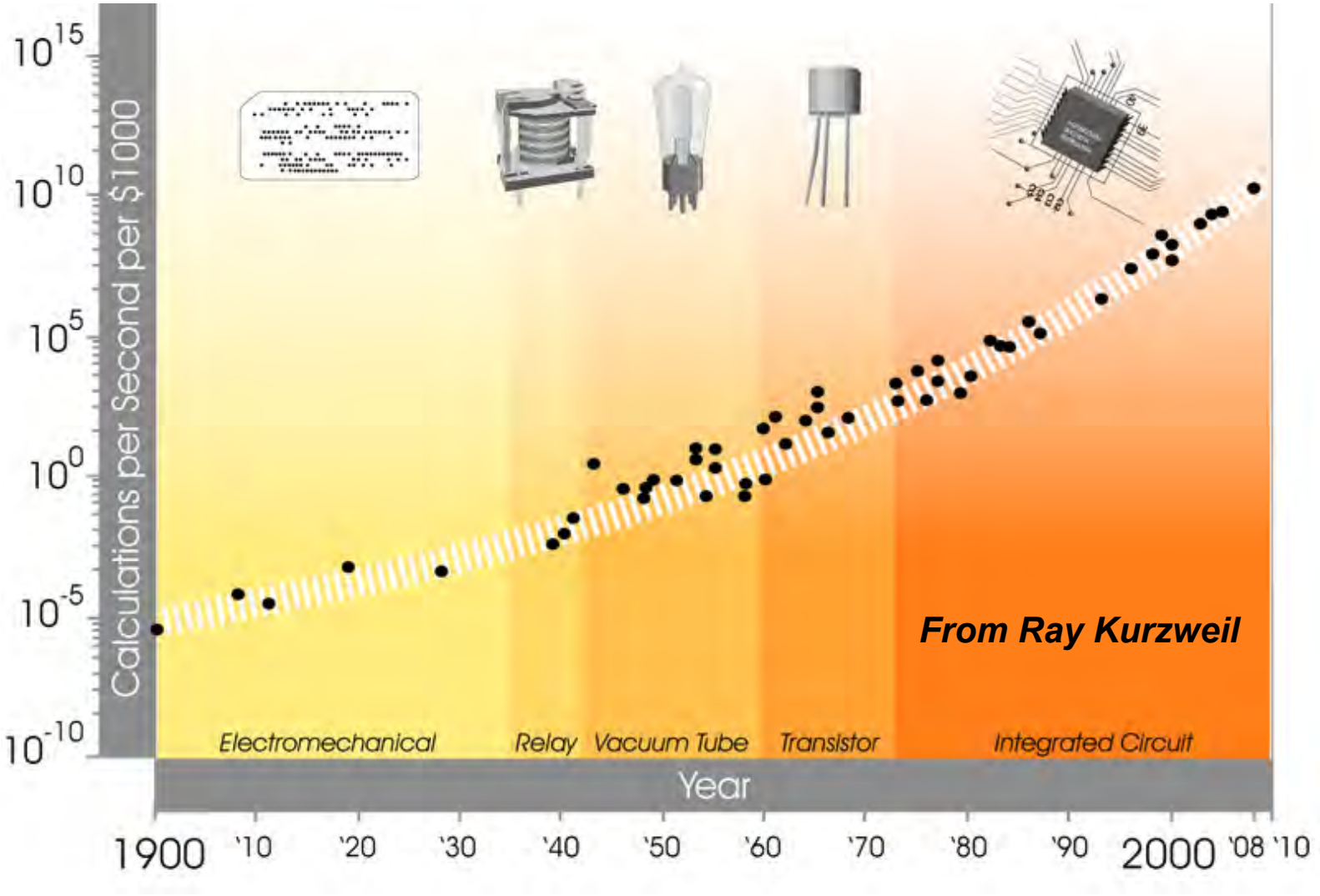
O(100k) nodes, ~30 MW
O(10-100M) system threads
< 10 Exaflops

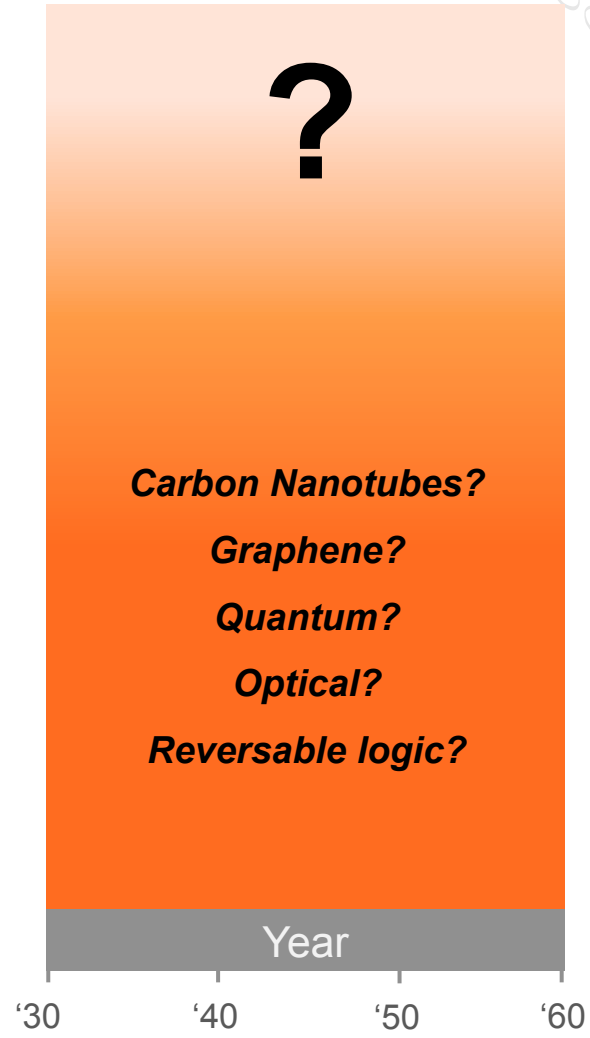


Low diameter networks with optics

Over the next decade, computers won't change that much from the current model

What Comes After CMOS?





Will Change…

01 The “Best” Processor

Applications, Workloads, Algorithms and Users 02

03 Memory and Storage Hierarchy

Won’t Change…

01 Our Commitment to Innovation

Partnership with Our Customers 02

03 Unwavering Customer Support

Remember, some great things never change…

The Rolling 5tones Fifty Years

THANKS!