Convergence of HPC and Big Data : Architecture Panel

NITRD Workshop @ Bethesda

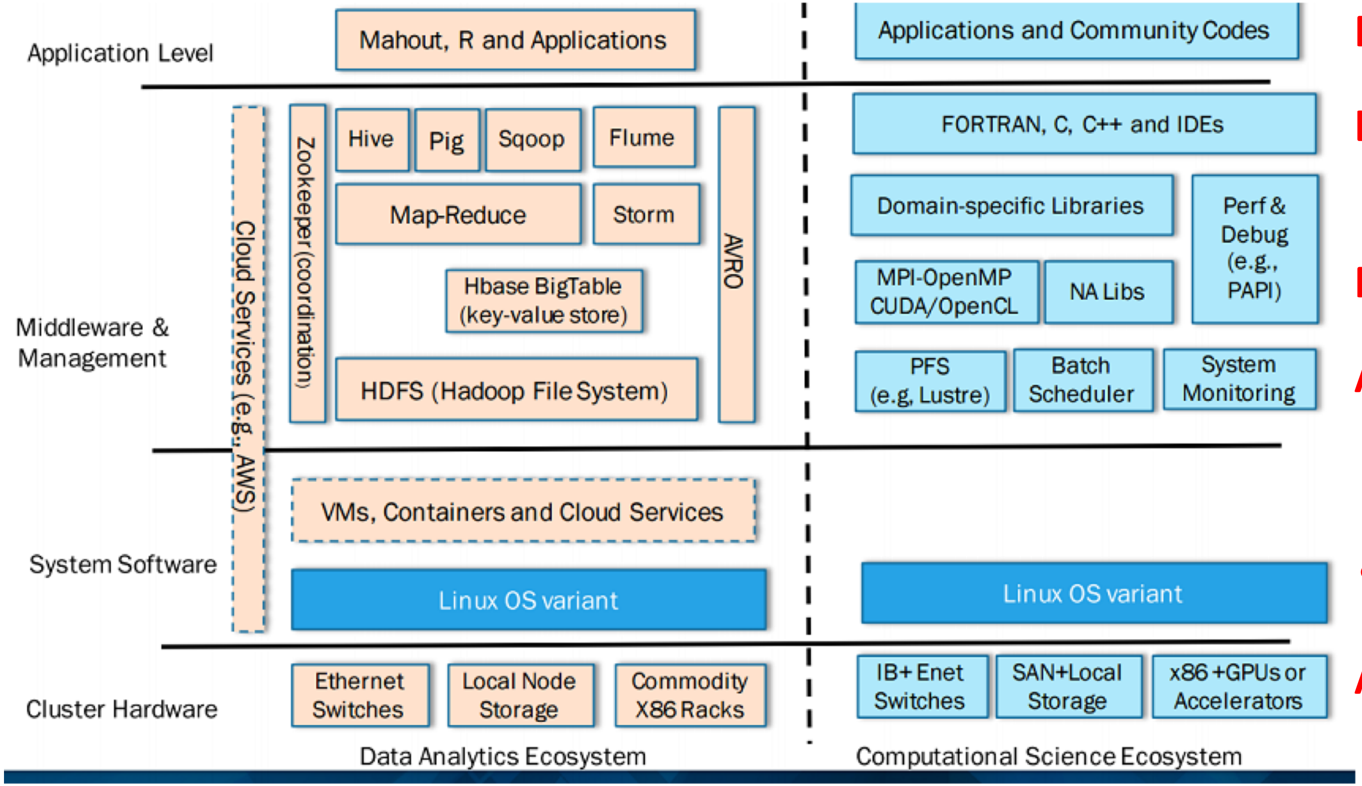
Rangan Sukumar, Senior Analytics Architect, Office of the CTO
Cray Inc.

Convergence: Goal and Success

- **Convergence goals – as a constrained-optimization problem**
  - maximize(performance-per-$)
  - minimize($-to-insight)
  - min(operating costs – power, downtime, human_resources)
  - max(architected performance * community productivity) <= budget
  - min(benchmark-performance) >= Scaling_factor
  - max(app-to-app performance variation) <= epsilon

- **Posit: Real success of convergence is integrating flexibility with heterogeneity**

Convergence: Tale of Two Ecosystems



Performance

Languages and tools

Programming models

Access policy

Resource manager and Job-scheduling

Architectural differences

J. Dongarra et al., Exascale computing and Big Data: The next frontier, ACM Communications 2015

Convergence: Tale of Two Ecosystems

Application Level

Mahout, R and Applications

Applications and Community Codes

- Hypothetical "best" node
  - Sub-Precision
  - Fat nodes (lots of RAM)
  - ASICs (Training + Inference)
  - Persistent node-local storage
  - Run-time code optimization

- Hypothetical "best" node
  - High-Precision
  - Vector extensions
  - CPUs/GPUs with High Bandwidth Memory
  - I/O over memory hierarchy
  - Static code-optimization

Cluster Hardware

Ethernet Switches | Local Node Storage | Commodity X86 Racks

IB+ Enet Switches | SAN+Local Storage | x86 +GPUs or Accelerators

Data Analytics Ecosystem

Computational Science Ecosystem

J. Dongarra et al., Exascale computing and Big Data: The next frontier, ACM Communications 2015

Convergence Requirements: Tale of Two Ecosystems

	Scientific Computing	Enterprise Computing
Primarily used for	Solving equations	Search/Query, Machine learning
Philosophy	Send data to compute	Send compute to data
Efficiency via	Parallelism	Distribution
Scaling expectation	Strong (scale-up)	Weak (scale-out)
Programming model	MPI, OpenMP, etc.	Map-reduce, SPMD, etc.
Popular languages	FORTRAN, C++, Python	Java, Scala, Python, R
Design strength	Multi-node communication using an interconnect	Built-in job fault tolerance over Ethernet
Access model	On-premise	Cloud-like
Preferred algebra	Dense Linear	Set-theoretic / Relational
Memory access	Predictable	Random
Storage	Centralized, POSIX/RAID	Decentralized, Duplication

Convergence Requirements: Workflows + Workload

	Scientific Computing	Enterprise Computing
Data (Structured)	Vector, Matrix, Tensor	Table, Key-Values, Objects
Data (Unstructured)	Mesh, Images (Physics-based)	Documents, Images (Camera)
Visualization	Voxel, Surface, Point Clouds	Word Cloud, Parallel Coordinates, BI Tools
Validation	Cross-validation (ROC curves, statistical significance)	Manual / Subject matter expert, A/B testing
Extract, Transform, Load	Fourier, Wavelet, Laplace, etc. Cartesian, Radial, Toroidal, etc.	File-format transformations e.g. CSV to VRML
Search (Query)	Properties such as periodicity, self-similarity, anomaly, etc.	SQL, SPARQL, etc. (Sum, Average, Group by)
Funding Model	Non-profit grand challenge (Answer matters)	Value-driven (Cost matters)

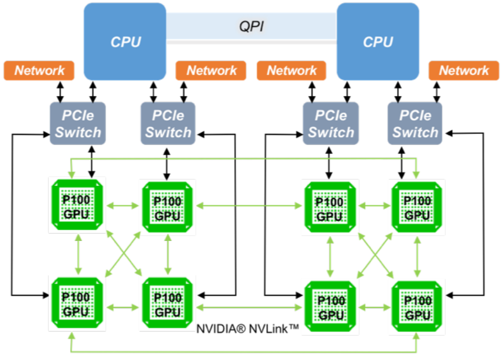
Sukumar, S. R., et al., (2016, December). Kernels for scalable data analysis in science: Towards an architecture-portable future. *In the Proc. Of the 2016 IEEE International Conference on Big Data,* pp. 1026-1031.

Convergence Requirements: AI Deployment

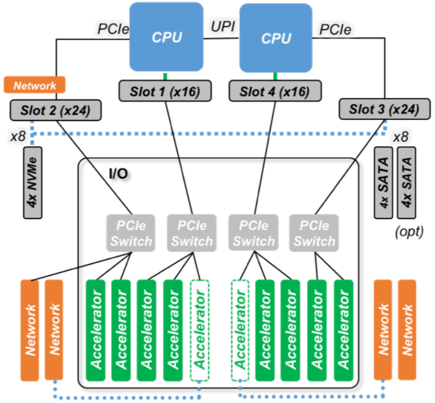
	Scientific Computing	Enterprise Computing
Model	Domain-specific	CNN, RNN, LSTM, GAN etc.
Baseline	Theoretic e.g. Navier Stokes	Humans, Other ML algorithms
Parallelism	Model, Ensemble	Data
Use Case	Computational Steering Proxy models	Speech, Test Image interpretation Hyper-personalization
Source File System	Lustre and GPFS	HDFS, S3, NFS etc.
Figure of Merit	Interpretability, Feasibility	Time-to-accuracy, Model-size
Training Data	O(GBs) per sample, O(10^3) samples, O(10) categories	O(KBs) per sample, O(10^6) samples, O(10^4) categories
Data Model	HDF5, NETCDF	Relational, Document, Key-Value

Convergence: Early Experience @ Cray

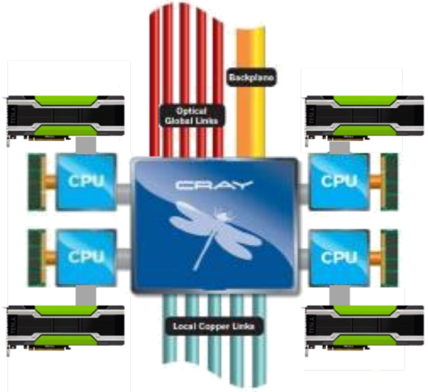
Cray CS-Storm 500NX Dense GPU System



Cray CS-Storm 500GT Dense GPU System

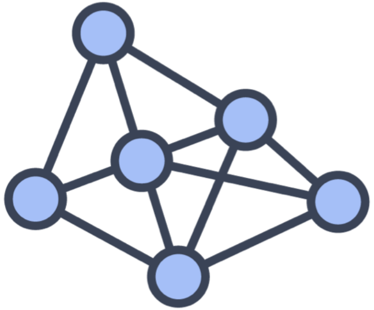


Cray XC-50 Accelerated GPU System



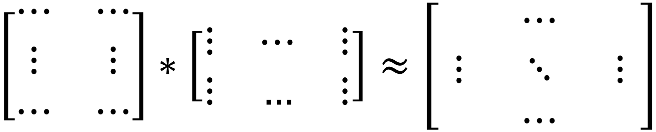
Convergence: Early Experience (Optimism)

Graph Analytics



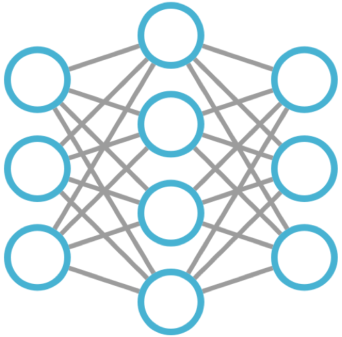
Handle 1000x bigger datasets with a 100x better speed-up with queries

Matrix Methods



Get 2-26x over Big Data Frameworks like Hadoop, Spark (for the same cluster-size)

Deep Learning

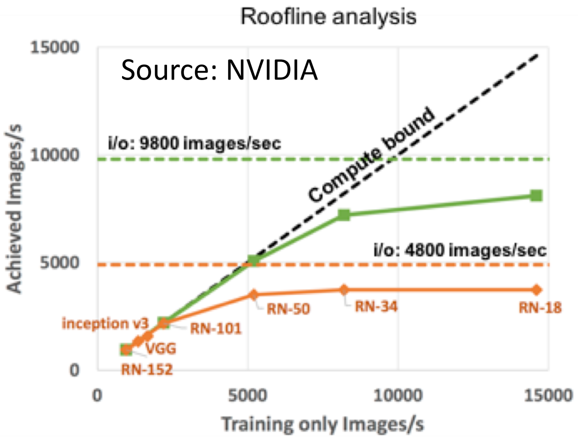


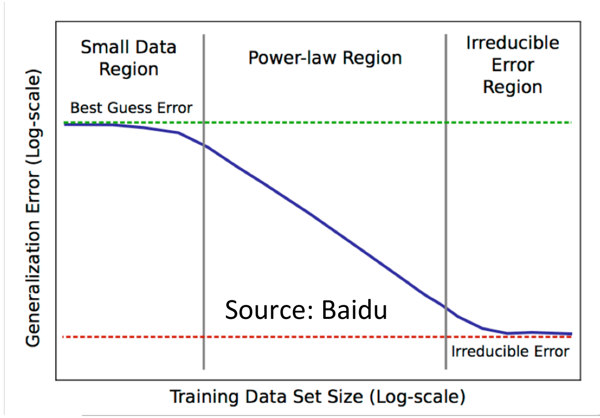
95%+ scalability efficiency that can reduce training time from days to hours

Best practices:

- Application fine-tuning / Performance optimization
- High-performance interconnect
- Algorithmic cleverness to trade compute and i/o
- Overlap compute and i/o with programming model

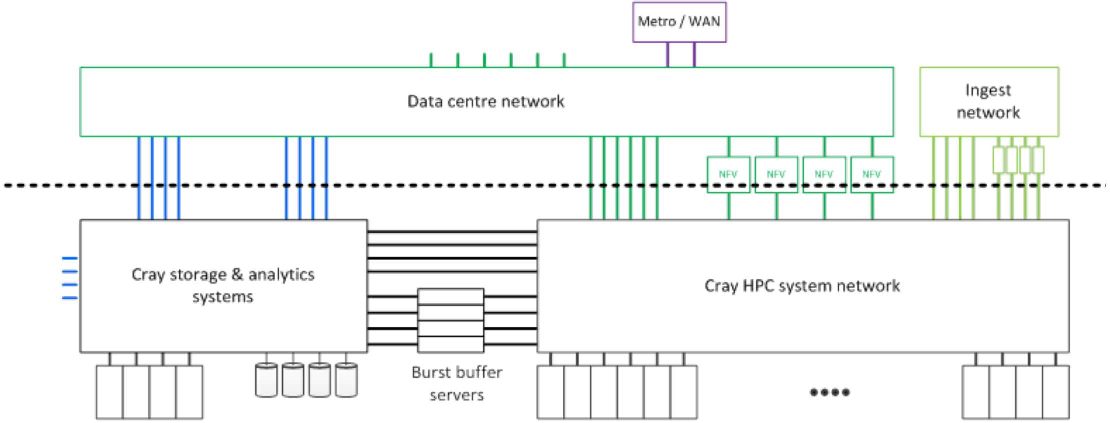
Convergence: Early Experience (Pessimism)





ResNet-50 Success	Time-to-accuracy	How many GPUs?	Scalability Efficiency
Facebook (Caffe2)	2 days 1 hour	352 GPUs 256	90% (large-batch)
IBM PowerAI (Caffe)	50 minutes	256 GPUs	95% (large-batch)
Google (TensorFlow)	~24 hours	64 TPUs	>90%
Preferred Networks (Chainer)	15 minutes	1000 GPUs	>90%
Cray @ CSCS (Tensorflow)	<14 minutes	1000 GPUs	~>95%
Tencent	< 7 minutes	2048 GPUs	Large batch @ 64K
Fast.ai on AWS (Cost: $40)	~18 minutes	128 GPUs	Not available (large batch)

Convergence Future: Cray Shasta System

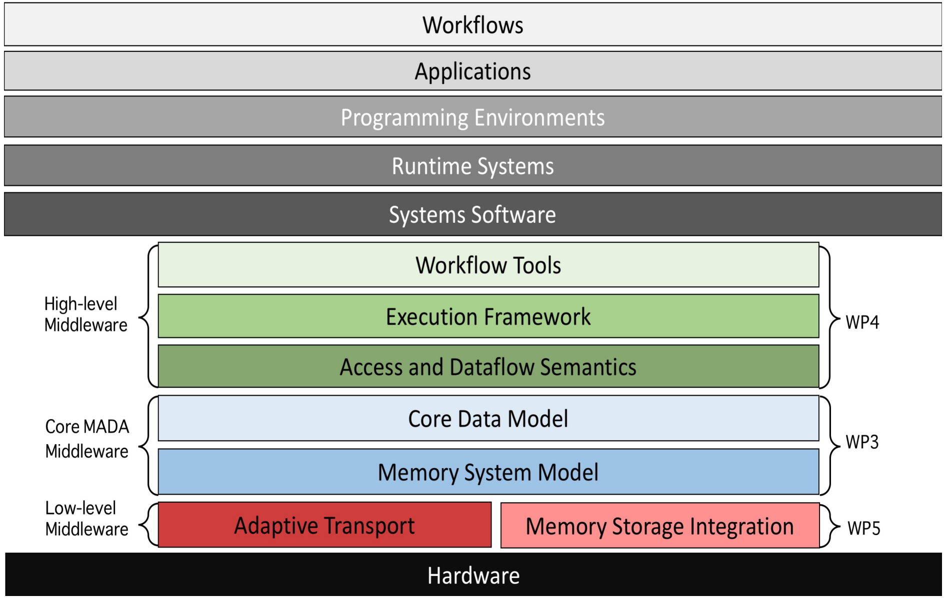


Flexibility with heterogeneity

	Vendors	Features
Integrated systems	Dell, HPE, Cray, Inspur, NVIDIA...	Integration, Scaling, Turn-key
Provisioning	Bitfusion, Ace, Bright Computing	Virtualization, Scheduling
Inter-connect	Intel, Cray, Mellanox	OPA, Aries, InfiniBand
Node architecture	NVIDIA, Facebook, Cray	Density, CPU:GPU ratios
Motherboard	Quanta, Supermicron etc.	PCIe, NCCL, GPU-Direct
xPU	Intel, NVDIA, AMD, ARM (40+ startups)	CPUs, GPUs, ASICs

Convergence Future: Technologies

Convergence is not all hardware.....



	Operating Systems	Runtimes	Systems Software	Programming Environments	Applications	Workflows
HBM		memkind			memkind	
GPU MEM		CUDA	CUDA	PTX	CUDA	
DRAM	C / ASM	C / ASM	C	C / ASM	C / Fortran	
NV-DIMM		pmem	pmem		pmem / pmemkind	pmem / pmemkind
LOCAL SSD					POSIX	POSIX
BURST BUFFER					DSL (e.g Datawarp)	DSL (e.g Datawarp)
Network SSD					POSIX	POSIX
DISK / PFS	POSIX / swap				POSIX / MPI-IO	POSIX
TAPE						TSM
CLOUD						S3

Lot more work before convergence can be productive….

Summary: What is in the future?

- **General purpose flexibility**
  - Commodity-like configurations with custom processors, chips
- **Seamless heterogeneity**
  - CPUs, GPUs, FPGAs, ASICs
- **High-performance interconnects for data centers**
  - MPI and TCP/IP collectives, compute on the network
- **Unified software stack with micro-services**
  - Programming environment for performance and productivity
- **Workflow optimization**
  - Match growth in compute, model-size and data with I/O

Thank You

Convergence: What would it take?

	Hardware	Software	Ecosystem
	System	**Function**	**Community Productivity**
Facility Performance	**Utilization** Peak vs. Sustained, Performance per $	**Application/Codes** e.g. Deep Learning, Graph analytics	**Domain-specific Creativity** Is there an ecosystem of sustainable community (open-source) engagement that enables vertical segments?
System Performance	**Reliability** Faults, MTTF, Uptime **Scalability** Weak and strong	**Kernel/Motif** e.g. DGEMM, SYRK, ReLU, inner product	**Code Portability** Does a user have to rewrite code? Does vendor support code porting for novel architectures?
Multi-node Performance	**System Architecture** **Interconnect** eth, InfiniBand, Aries **Provisioning** Mesos, Moab, SLURM	**Programming Model** e.g. MR, PGAS, GRPC	**Programmability** Does an end-user have to learn a new language or can they launch jobs with modern tools (e.g. notebooks)?
Node Performance	**Node Architecture** # of xPUs+ cache + memory + network	**Libraries** e.g. MKL, CUDA, libSci **Collectives** e.g. NCCL, MPI	**Data Pre-Processing** Does system offer tools to optimize ETL wall-time?
Component Performance	**Disk** Latency **Memory** Capacity, Latency **xPU** Speed i/o	**Data Structure** e.g. matrix, sequences, unstructured grids	**Data Movement** Does system provide ability to run multiple frameworks/applications on the same data?

The Networking and Information Technology Research and Development (NITRD) Program

Mailing Address: NCO/NITRD, 2415 Eisenhower Avenue, Alexandria, VA 22314

Physical Address: 490 L'Enfant Plaza SW, Suite 8001, Washington, DC 20024, USA Tel: 202-459-9674, Fax: 202-459-9673, Email: nco@nitrd.gov, Website: https://www.nitrd.gov