Architectural Challenges Emerging From The Convergence of Big Data, HPC and AI

PDSW-DISC Keynote @ SC18

ssukumar@cray.com
@Rangan_Sukumar
https://www.linkedin.com/in/rangan/

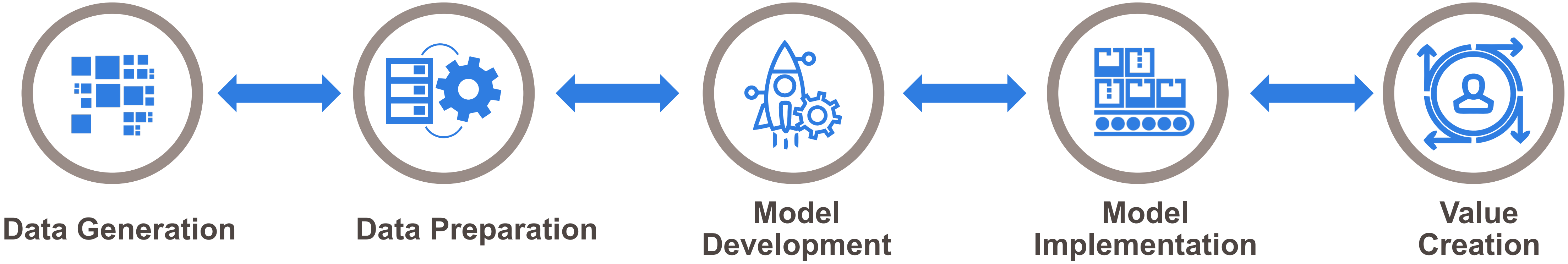
CRAY™

SAFE HARBOR STATEMENT

This presentation may contain forward-looking statements that are based on our current expectations. Forward looking statements may include statements about our financial guidance and expected operating results, our opportunities and future potential, our product development and new product introduction plans, our ability to expand and penetrate our addressable markets and other statements that are not historical facts.

These statements are only predictions and actual results may materially vary from those projected. Please refer to Cray's documents filed with the SEC from time to time concerning factors that could affect the Company and these forward-looking statements.

ARCHITECTING A CONVERGENT SYSTEM



Levels of Architectural Maturity Towards Convergence

- **Level 1:** Can a system run HPC, Big Data and AI applications/workloads?
- **Level 2:** Can a system execute a “convergent workflow” consisting of HPC, Big Data and AI tools, codes and frameworks in reasonable time?
- **Level 3:** Can a system accelerate/scale the workflow when required?
- **Level 4:** Given a workflow, is this the top “performant” system one can build?

THE CHALLENGE OF ARCHITECTING HPC SYSTEMS

- **Design specifications**
  - maximize(performance-per-$)
  - minimize($-to-insight)
  - maximize(architected performance * community productivity) <= budget
  - minimum(benchmark-performance) >= scaling factor
  - maximum(app-to-app performance variation) <= epsilon
  - minimize(operating costs ~ power, downtime, human resources)

- **Figures-of-merit**
  - FLOPS, Programmability, Utilization, Benchmark, Scientific Innovation, …

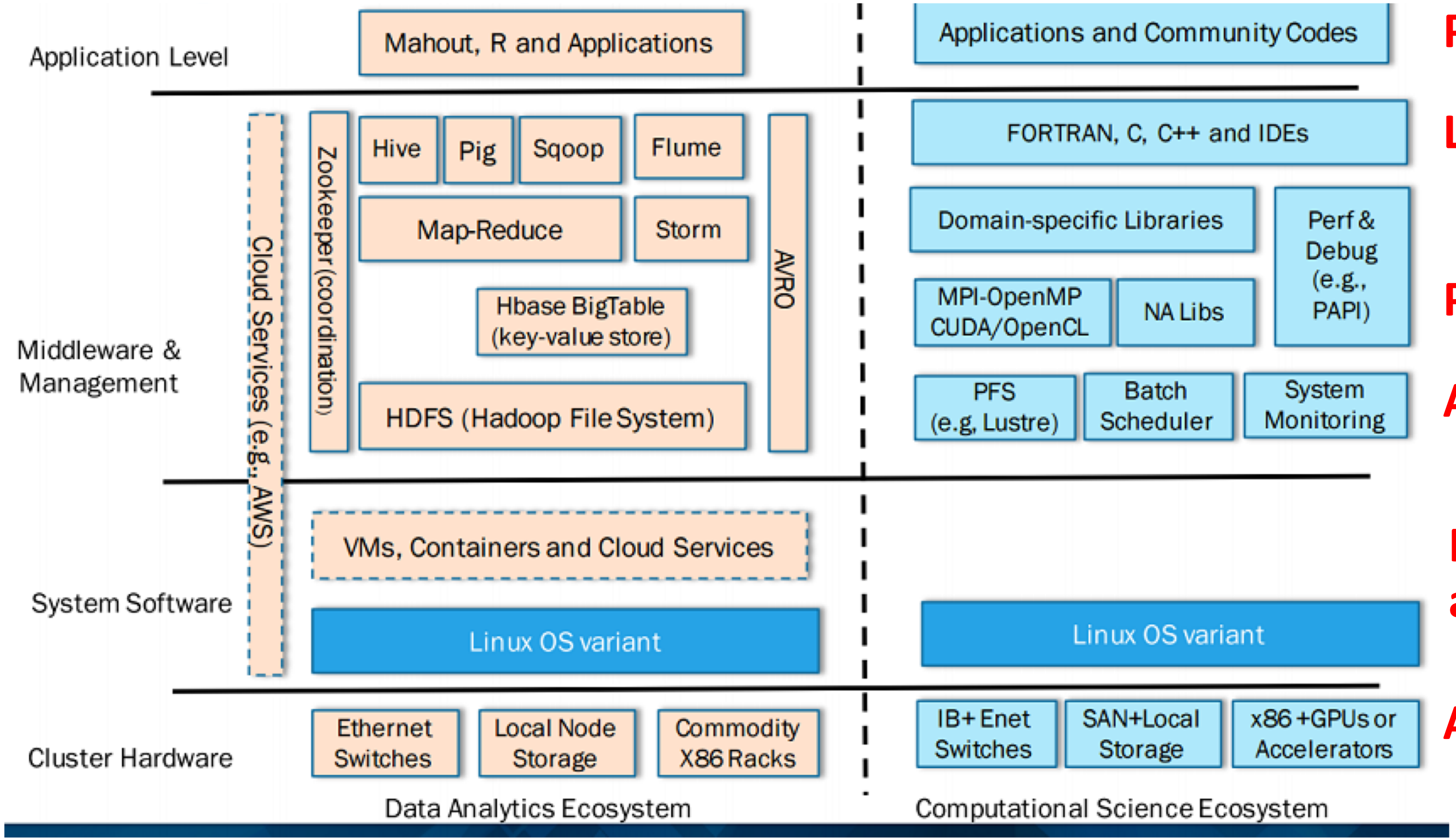
THE CHALLENGE OF ARCHITECTING BIG-DATA SYSTEMS

- **Design specifications**
  - maximize(ROI-per-byte)
  - maximize(capacity * consistency * availability * fault-tolerance)
  - maximize(open-source tool support)
  - minimize(time-to-prototype + time-to-production)
  - minimize(security risk)
  - minimize(operating costs ~ power, downtime, human resources)

- **Figures-of-merit**
  - ROI, Elasticity, Multi-tenancy, Ease-of-use, Time-to-accuracy,…

TODAY: IT IS THE TALE OF TWO ECOSYSTEMS

CRAY



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

REQUIREMENTS: MODE OF OPERATION

	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

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.

REQUIREMENTS: PROCESS AND 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

REQUIREMENTS: USER EXPERIENCE

	Scientific Computing	Enterprise Computing
Programming	Vendor libraries and compilers	Open-source services and APIs
Preferred Deployment	Bare metal	Virtualized, Containers
Popular Architecture	Homogenous	Heterogenous
“Systems” Literacy	High	Low
Scheduling	Batch	Interactive, Persistent
Resource Managers	SLURM, SGE, etc.	Mesos, Kubernetes, etc.
Data in	Files	Databases (in-memory, schema)
Software	Write-once Run-many	Write-Many Run-Many
Access Interface	Terminal, SSH	Jupyter, Web-based IDEs

IS CONVERGENCE NECESSARY?

AI methods benefit from HPC best practices

CONVERGENCE HEADLINES: BIG DATA + AI

MILESTONES IN ARTIFICIAL INTELLIGENCE

CHESS
IBM's Supercomputer Deep Blue defeats Garry Kasparov .

Evaluate Moves

1997

SELF-DRIVING
Stanley completes 132-mile course through Mojave Desert

Sense, Adapt, Act and React

2005

JEOPARDY
IBM Watson wins on Jeopardy!

Q&A Engine

2011

IMAGENET
Computers better than humans at recognizing images

Image interpretation

2015

GO
Google's Alpha Go beats a Go! world champion

Learning without rules

2016

1994

AMAZON
Books

1998

GOOGLE
Search
> 1200 PBs

2004

FACEBOOK
Image
> 500 TBs/day

2005

YOUTUBE
Video
> 400 PBs

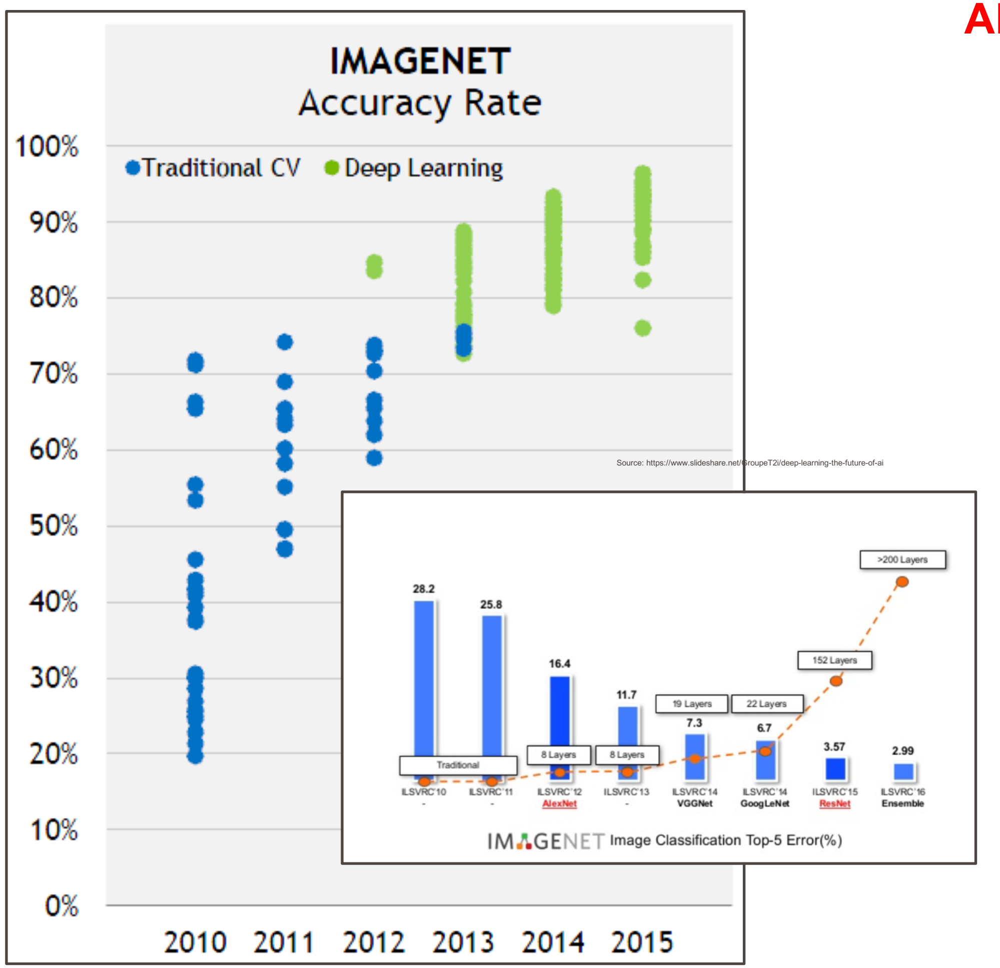
2006

SPOTIFY
Audio
>42 PBs

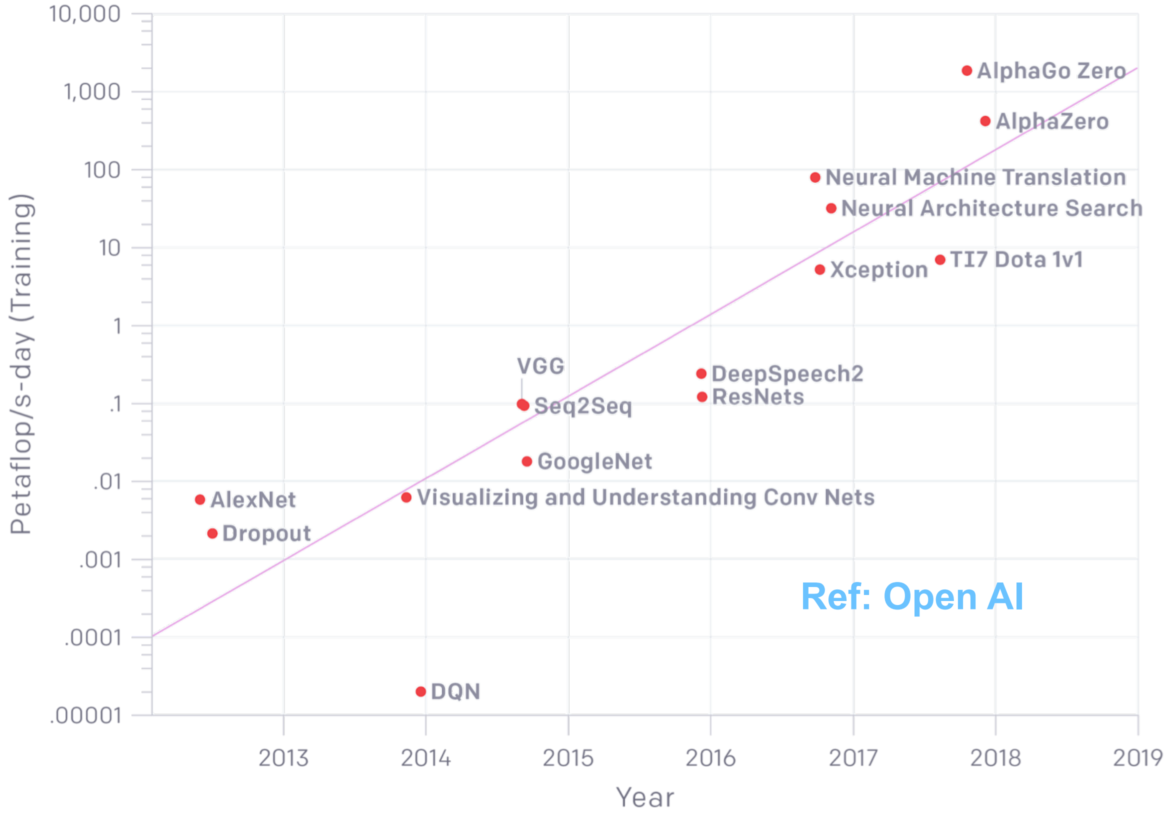
GROWTH IN UNSTRUCTURED BIG DATA

HPC-SCALE REQUIREMENTS AT AI PRACTITIONERS

Source: NVIDIA-ces-2016-press-conference

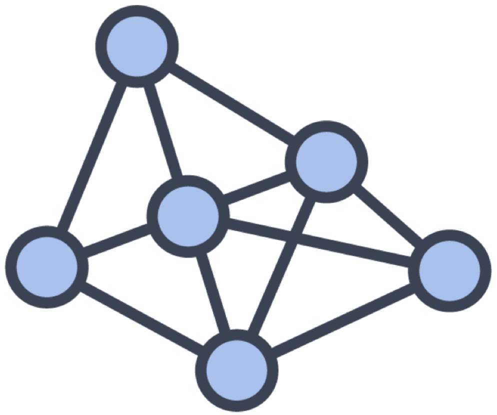


ALEXNET TO ALPHAGO ZERO: A 300,000x INCREASE IN COMPUTE



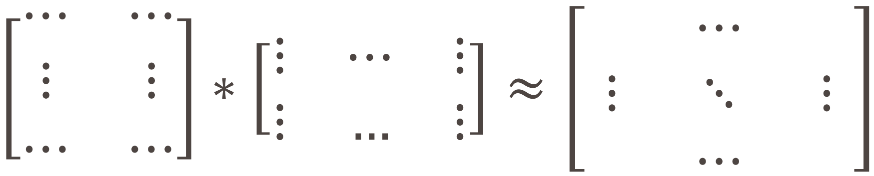
BENEFITS OF HPC ADOPTION

Graph Analytics



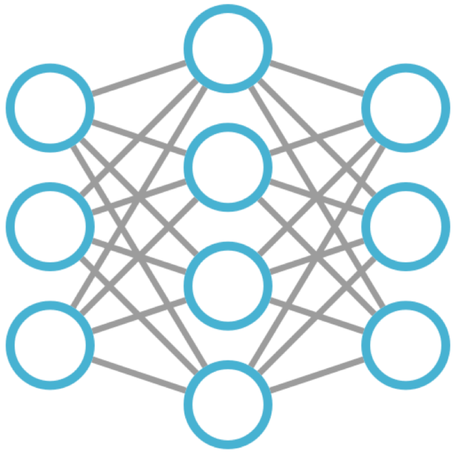
Handle 1000x bigger datasets with a 100x better speed-up with queries

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

HPC BUILDS THE MODELS OF TOMORROW

MORE DATA, BIGGER MODELS, NEED FOR MORE EFFICIENT AND PRODUCTIVE HARDWARE

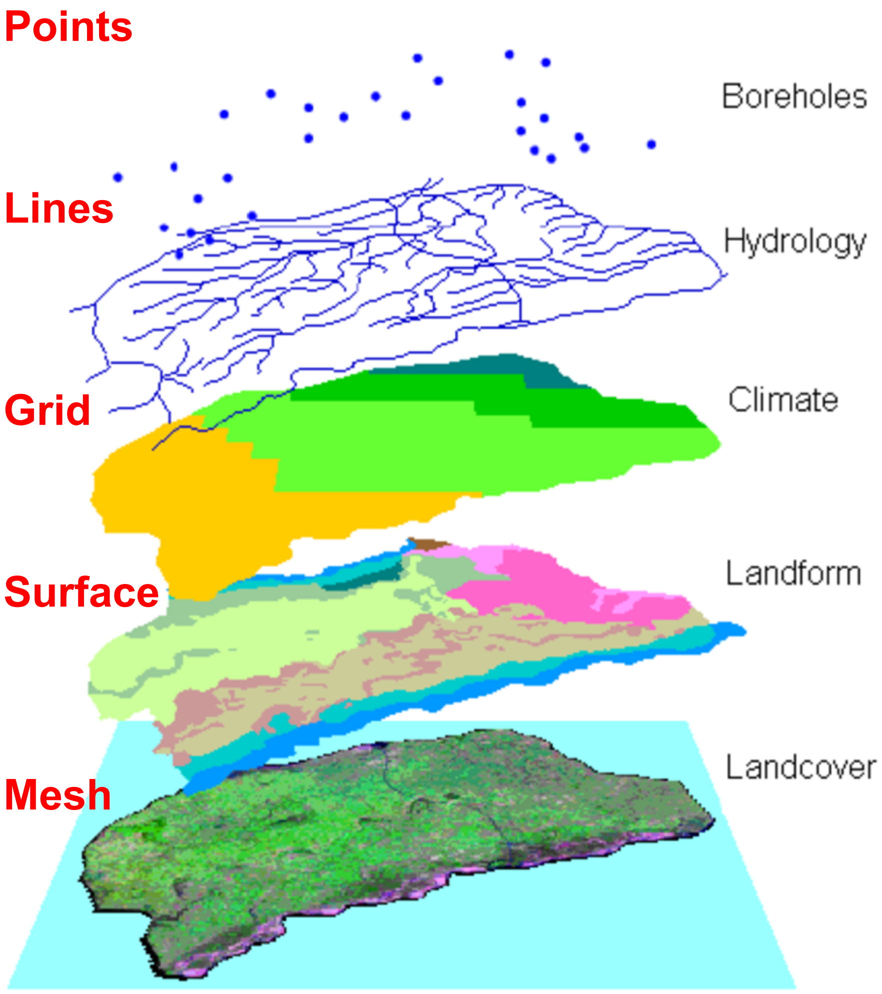
Figures-of-merit	State-of-practice	Projected 1-2 years ahead
Training-time to best accuracy	5+ days	2+ hours
Model Cost / TB (AWS GPUs)	~$25K (ResNet training on 80 GPUs for 5 days)	~10K
Hardware Efficiency	Network Depth: Flops::20x: 16x (based on AlexNet-2012 and ResNet-2015)	O(Teraflops) / problem
Statistical Efficiency	Depth: Accuracy:: 20x:13+ (based on AlexNet-2012 and ResNet-2015)	O(Teraflops) / problem
Need for compute as data grows	Data: Flops: Error:: 2x: 5x: 3+ (based on DeepSpeech1 and DeepSpeech2)	O(Petaflops) / problem
Training Cadence	~ Monthly	~ Daily
# of models per organization	1x	10-100x

IS CONVERGENCE NECESSARY?

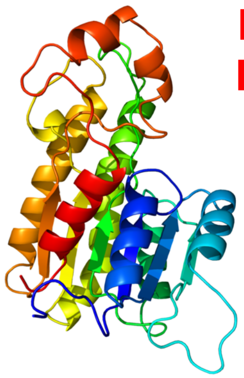
AI methods offer tremendous capabilities for scientific data

DIFFERENT SHAPES OF SCIENTIFIC DATA

Geospatial Data



Molecular Data



Molecular homology

Omics Data

Sequences

Streaming Data

Time series

Graph Data

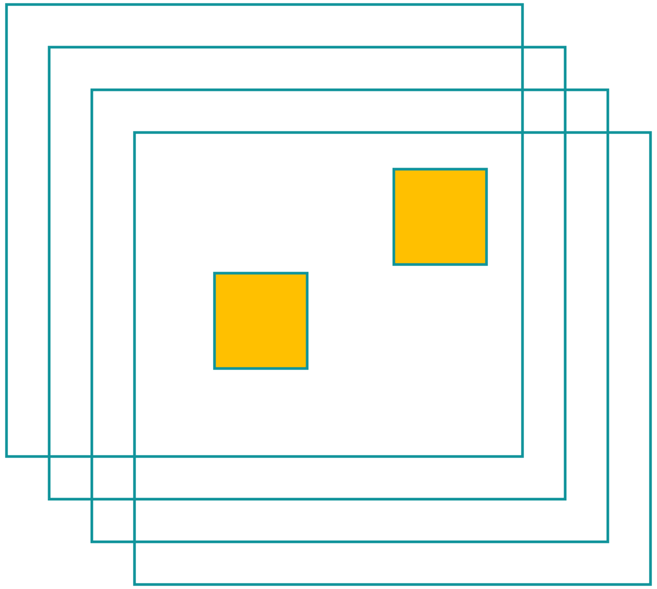
Associations

2D, 3D, 4D volumes, Higher precision (32, 64 bit), Higher # channels (3, 16, 1024), Sparse + Dense, Resolution

DIFFERENT SEARCH SPACES OF SCIENTIFIC DATA

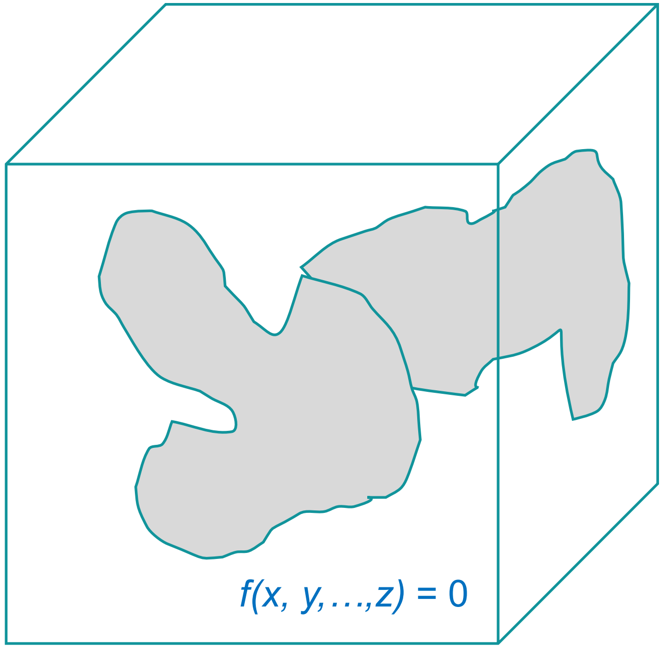
CRAY

Feature-based



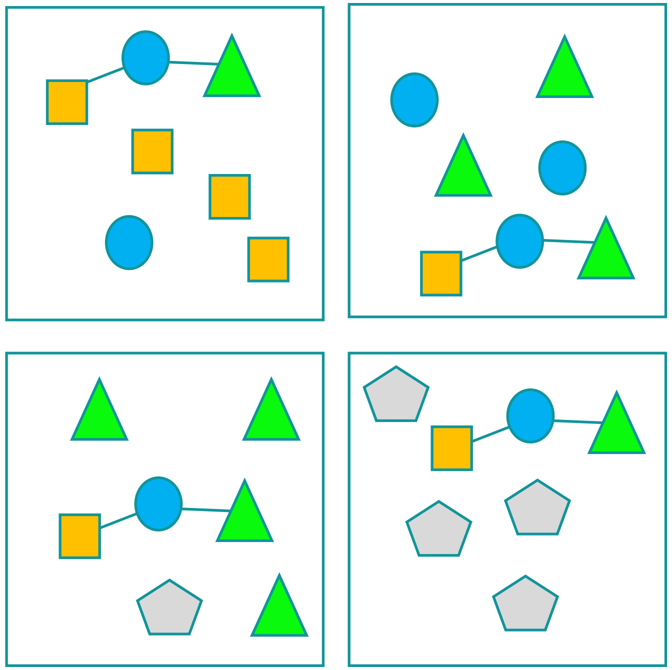
Structured Data
Feature-space is ill-posed
Search is well-defined

Function-based



Unstructured Data
Feature-space is theoretical
Search is empirical

Pattern-based



Semi-Structured Data
Feature-space to be discovered
Search is P or NP-hard

Opportunity to create the ”models of tomorrow”

MODELS EXPLORE AUTO-ENGINEERED FEATURE SPACES

Path towards explain-ability

Capsule Networks

Mixture of Experts

Neural Collaborative Filtering

Block-Sparse LSTM

ReLu

Dropout

BatchNorm

Decoder/Encoder

Pooling

LSTM
GRU

WaveNet

Beam Search

Attention

Simulation

DQN

DDPG

Domain-specific Models

Reinforcement Learning

Generative Models

Recurrent Neural Networks

Convolutional Neural Models

3D-GAN

ConditionalGAN

CoupledGAN

SEGAN

MedGAN

Before 2012	2012	2013	2014	2015	2016	2017	2018	2019

AI ADOPTION IS INCREASING IN THE SCIENCES

Domain-Specific Models

- Fluid Dynamics
- Schrodinger Equation
- Full-Wave Inversion
- Integration with user-facilities

Combinatorial Patterns

- Search for meta-stable states
- Search for particles
- Search for N-way correlations
- Search in a high-dimensional feature space

Emerging Use-cases

Model Repurposing

- Deep Learning for X
- Feature engineering
- Predictive modeling
- Hypothesis creation with interpretation

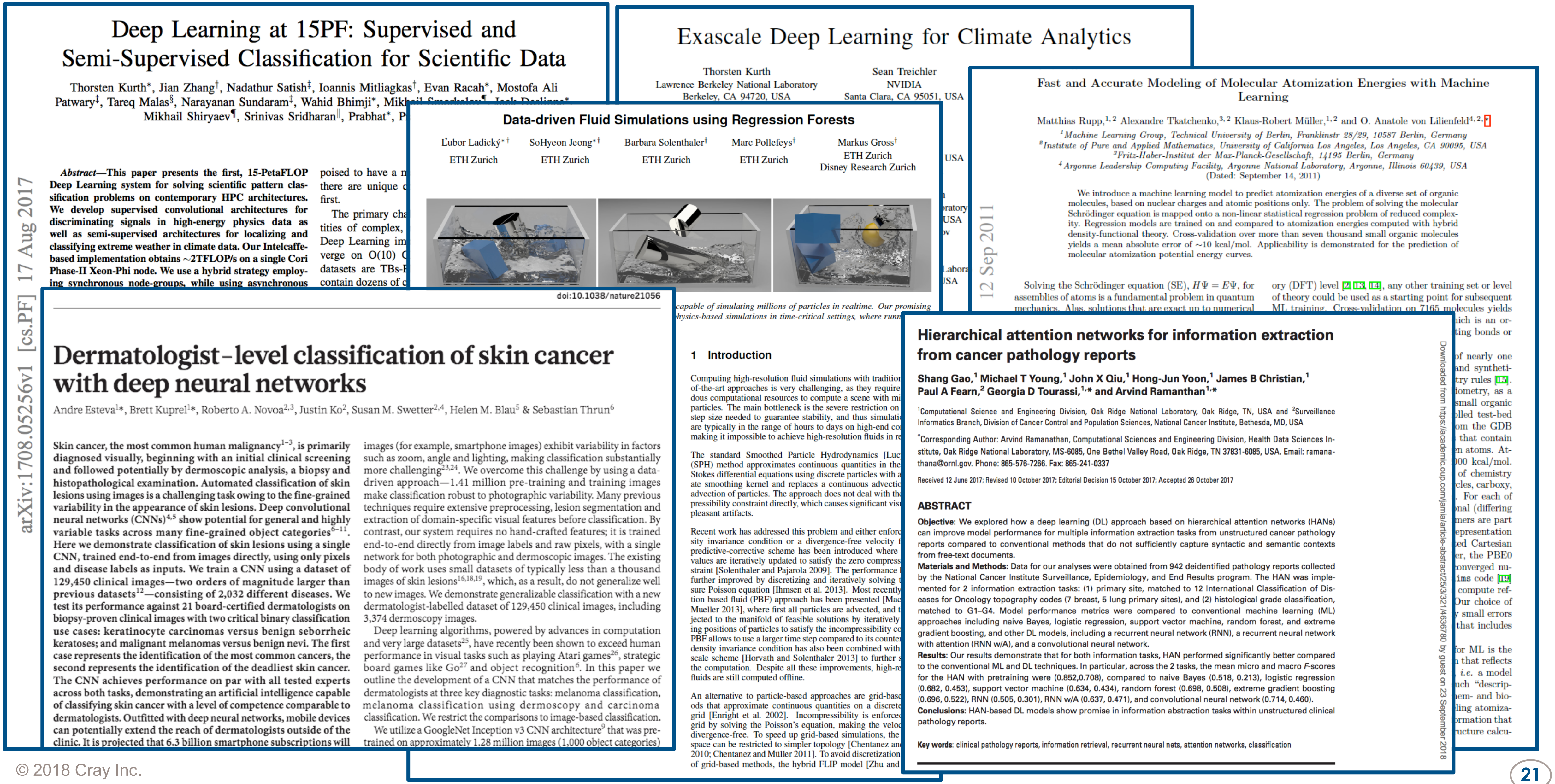
Workflows and Automation

- Smarter initialization for simulations
- Computational steering
- Hybrid models of theory and AI

Model Complexity

Computational Complexity

CONVERGENCE HEADLINES: AI @ HPC CENTERS



Deep Learning at 15PF: Supervised and Semi-Supervised Classification for Scientific Data

Thorsten Kurth*, Jian Zhang†, Nadathur Satish‡, Ioannis Mitliagkas†, Evan Racah*, Mostofa Ali Patwary‡, Tareq Malas§, Narayanan Sundaram‡, Wahid Bhimji*, Mik…

Mikhail Shiryaev¶, Srinivas Sridharan‖, Prabhat*, P…

arXiv:1708.05256v1 [cs.PF] 17 Aug 2017

Abstract—This paper presents the first, 15-PetaFLOP Deep Learning system for solving scientific pattern classification problems on contemporary HPC architectures. We develop supervised convolutional architectures for discriminating signals in high-energy physics data as well as semi-supervised architectures for localizing and classifying extreme weather in climate data. Our Intelcaffe-based implementation obtains ~2TFLOP/s on a single Cori Phase-II Xeon-Phi node. We use a hybrid strategy employing synchronous node-groups, while using asynchronous

Exascale Deep Learning for Climate Analytics

Thorsten Kurth, Lawrence Berkeley National Laboratory, Berkeley, CA 94720, USA

Sean Treichler, NVIDIA, Santa Clara, CA 95051, USA

Data-driven Fluid Simulations using Regression Forests

Ľubor Ladický*† (ETH Zurich), SoHyeon Jeong*† (ETH Zurich), Barbara Solenthaler† (ETH Zurich), Marc Pollefeys† (ETH Zurich), Markus Gross† (ETH Zurich, Disney Research Zurich)

1 Introduction

Fast and Accurate Modeling of Molecular Atomization Energies with Machine Learning

Matthias Rupp,[1,2] Alexandre Tkatchenko,[3,2] Klaus-Robert Müller,[1,2] and O. Anatole von Lilienfeld[4,2,*]

[1]Machine Learning Group, Technical University of Berlin, Franklinstr 28/29, 10587 Berlin, Germany
[2]Institute of Pure and Applied Mathematics, University of California Los Angeles, Los Angeles, CA 90095, USA
[3]Fritz-Haber-Institut der Max-Planck-Gesellschaft, 14195 Berlin, Germany
[4]Argonne Leadership Computing Facility, Argonne National Laboratory, Argonne, Illinois 60439, USA
(Dated: September 14, 2011)

We introduce a machine learning model to predict atomization energies of a diverse set of organic molecules, based on nuclear charges and atomic positions only. The problem of solving the molecular Schrödinger equation is mapped onto a non-linear statistical regression problem of reduced complexity. Regression models are trained on and compared to atomization energies computed with hybrid density-functional theory. Cross-validation over more than seven thousand small organic molecules yields a mean absolute error of ~10 kcal/mol. Applicability is demonstrated for the prediction of molecular atomization potential energy curves.

12 Sep 2011

doi:10.1038/nature21056

Dermatologist-level classification of skin cancer with deep neural networks

Andre Esteva[1]*, Brett Kuprel[1]*, Roberto A. Novoa[2,3], Justin Ko[2], Susan M. Swetter[2,4], Helen M. Blau[5] & Sebastian Thrun[6]

Skin cancer, the most common human malignancy[1–3], is primarily diagnosed visually, beginning with an initial clinical screening and followed potentially by dermoscopic analysis, a biopsy and histopathological examination. Automated classification of skin lesions using images is a challenging task owing to the fine-grained variability in the appearance of skin lesions. Deep convolutional neural networks (CNNs)[4,5] show potential for general and highly variable tasks across many fine-grained object categories[6–11]. Here we demonstrate classification of skin lesions using a single CNN, trained end-to-end from images directly, using only pixels and disease labels as inputs. We train a CNN using a dataset of 129,450 clinical images—two orders of magnitude larger than previous datasets[12]—consisting of 2,032 different diseases. We test its performance against 21 board-certified dermatologists on biopsy-proven clinical images with two critical binary classification use cases: keratinocyte carcinomas versus benign seborrheic keratoses; and malignant melanomas versus benign nevi. The first case represents the identification of the most common cancers, the second represents the identification of the deadliest skin cancer. The CNN achieves performance on par with all tested experts across both tasks, demonstrating an artificial intelligence capable of classifying skin cancer with a level of competence comparable to dermatologists. Outfitted with deep neural networks, mobile devices can potentially extend the reach of dermatologists outside of the clinic. It is projected that 6.3 billion smartphone subscriptions will

images (for example, smartphone images) exhibit variability in factors such as zoom, angle and lighting, making classification substantially more challenging[23,24]. We overcome this challenge by using a data-driven approach—1.41 million pre-training and training images make classification robust to photographic variability. Many previous techniques require extensive preprocessing, lesion segmentation and extraction of domain-specific visual features before classification. By contrast, our system requires no hand-crafted features; it is trained end-to-end directly from image labels and raw pixels, with a single network for both photographic and dermoscopic images. The existing body of work uses small datasets of typically less than a thousand images of skin lesions[16,18,19], which, as a result, do not generalize well to new images. We demonstrate generalizable classification with a new dermatologist-labelled dataset of 129,450 clinical images, including 3,374 dermoscopy images.

Deep learning algorithms, powered by advances in computation and very large datasets[25], have recently been shown to exceed human performance in visual tasks such as playing Atari games[26], strategic board games like Go[27] and object recognition[6]. In this paper we outline the development of a CNN that matches the performance of dermatologists at three key diagnostic tasks: melanoma classification, melanoma classification using dermoscopy and carcinoma classification. We restrict the comparisons to image-based classification.

We utilize a GoogleNet Inception v3 CNN architecture[9] that was pre-trained on approximately 1.28 million images (1,000 object categories)

Hierarchical attention networks for information extraction from cancer pathology reports

Shang Gao,[1] Michael T Young,[1] John X Qiu,[1] Hong-Jun Yoon,[1] James B Christian,[1] Paul A Fearn,[2] Georgia D Tourassi,[1,*] and Arvind Ramanathan[1,*]

[1]Computational Science and Engineering Division, Oak Ridge National Laboratory, Oak Ridge, TN, USA and [2]Surveillance Informatics Branch, Division of Cancer Control and Population Sciences, National Cancer Institute, Bethesda, MD, USA

*Corresponding Author: Arvind Ramanathan, Computational Sciences and Engineering Division, Health Data Sciences Institute, Oak Ridge National Laboratory, MS-6085, One Bethel Valley Road, Oak Ridge, TN 37831-6085, USA. Email: ramanathana@ornl.gov. Phone: 865-576-7266. Fax: 865-241-0337

Received 12 June 2017; Revised 10 October 2017; Editorial Decision 15 October 2017; Accepted 26 October 2017

ABSTRACT

Objective: We explored how a deep learning (DL) approach based on hierarchical attention networks (HANs) can improve model performance for multiple information extraction tasks from unstructured cancer pathology reports compared to conventional methods that do not sufficiently capture syntactic and semantic contexts from free-text documents.

Materials and Methods: Data for our analyses were obtained from 942 deidentified pathology reports collected by the National Cancer Institute Surveillance, Epidemiology, and End Results program. The HAN was implemented for 2 information extraction tasks: (1) primary site, matched to 12 International Classification of Diseases for Oncology topography codes (7 breast, 5 lung primary sites), and (2) histological grade classification, matched to G1–G4. Model performance metrics were compared to conventional machine learning (ML) approaches including naive Bayes, logistic regression, support vector machine, random forest, and extreme gradient boosting, and other DL models, including a recurrent neural network (RNN), a recurrent neural network with attention (RNN w/A), and a convolutional neural network.

Results: Our results demonstrate that for both information tasks, HAN performed significantly better compared to the conventional ML and DL techniques. In particular, across the 2 tasks, the mean micro and macro F-scores for the HAN with pretraining were (0.852,0.708), compared to naive Bayes (0.518, 0.213), logistic regression (0.682, 0.453), support vector machine (0.634, 0.434), random forest (0.698, 0.508), extreme gradient boosting (0.696, 0.522), RNN (0.505, 0.301), RNN w/A (0.637, 0.471), and convolutional neural network (0.714, 0.460).

Conclusions: HAN-based DL models show promise in information abstraction tasks within unstructured clinical pathology reports.

Key words: clinical pathology reports, information retrieval, recurrent neural nets, attention networks, classification

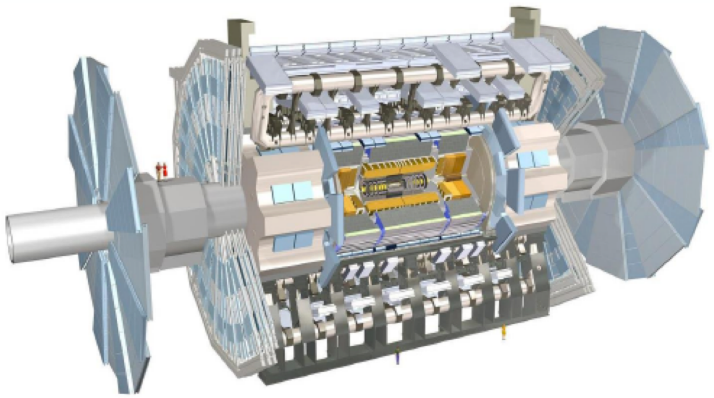
Downloaded from https://academic.oup.com/jamia/article-abstract/25/3/321/4636780 by guest on 23 September 2018

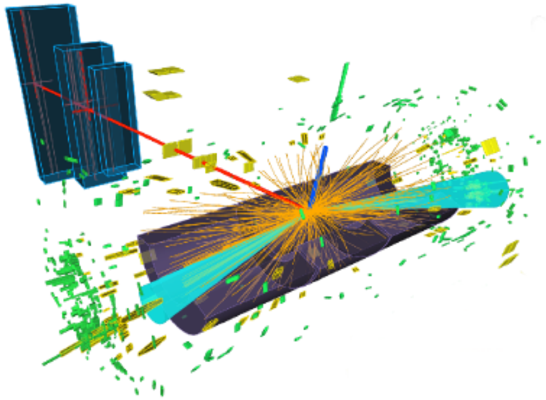
EXAMPLES OF CONVERGENT WORKFLOWS

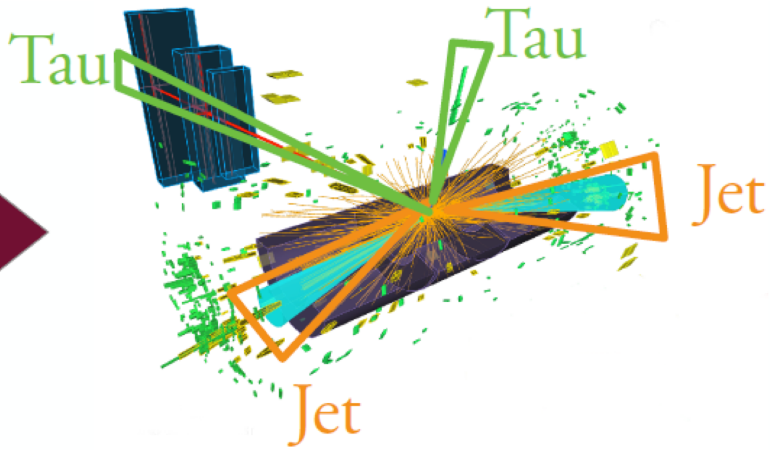
Big Data and AI @ HPC Centers

#1: COMPUTING AT SCIENTIFIC FACILITIES

State-of-the-art: Took 3000 collaborators nearly 10 years to build

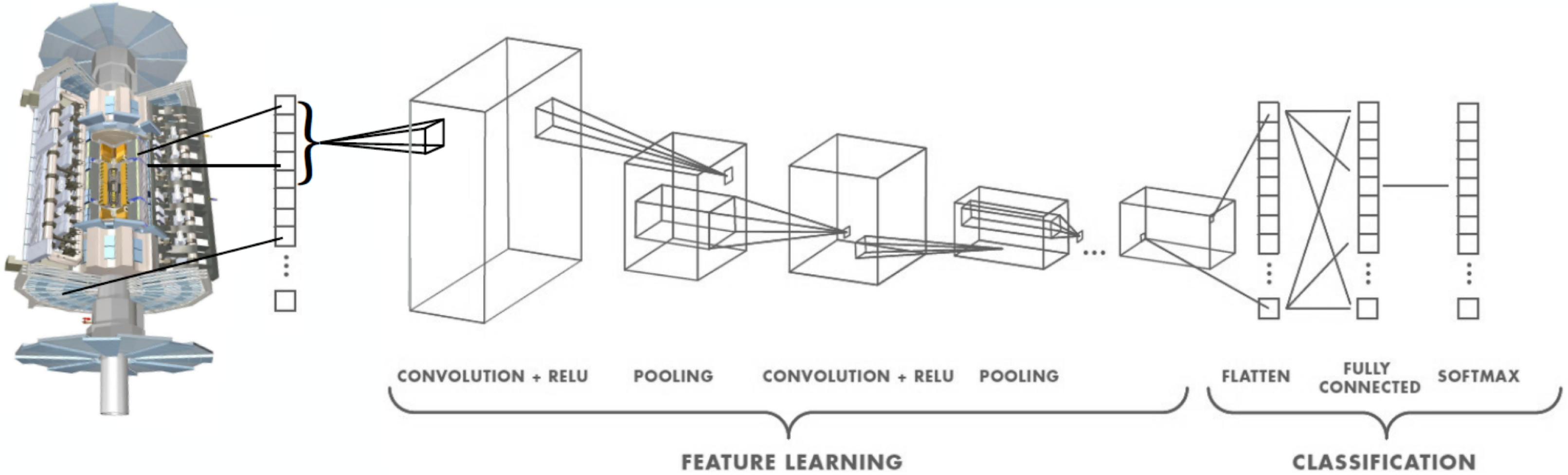


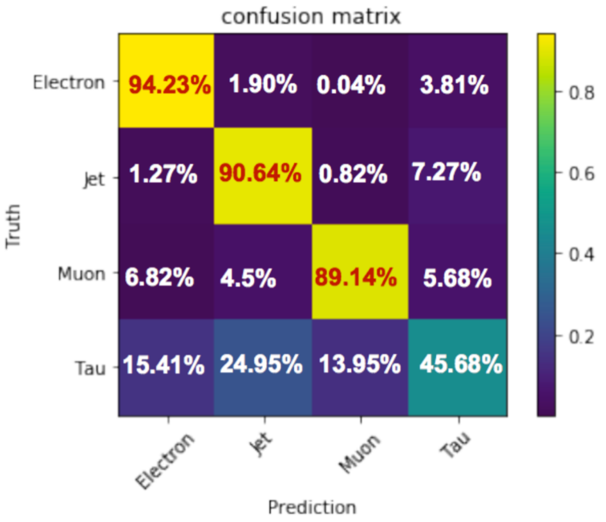




Using the labels from previous data analysis efforts….

Saves compute cycles….

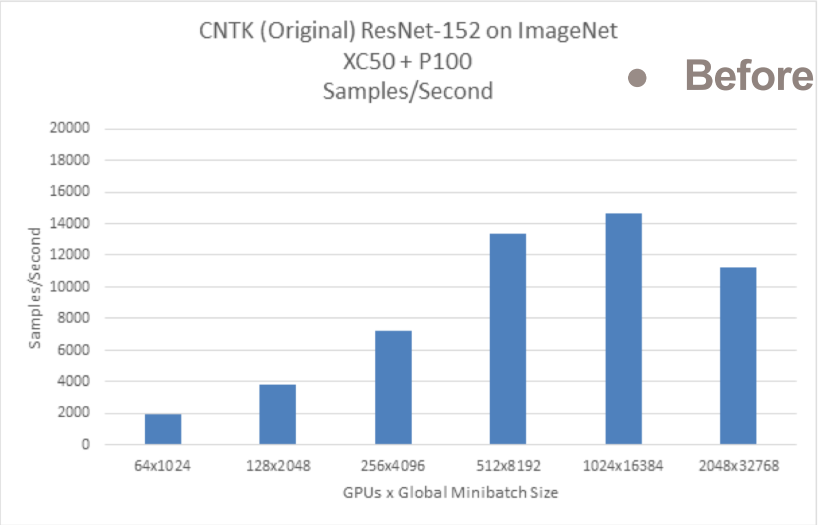


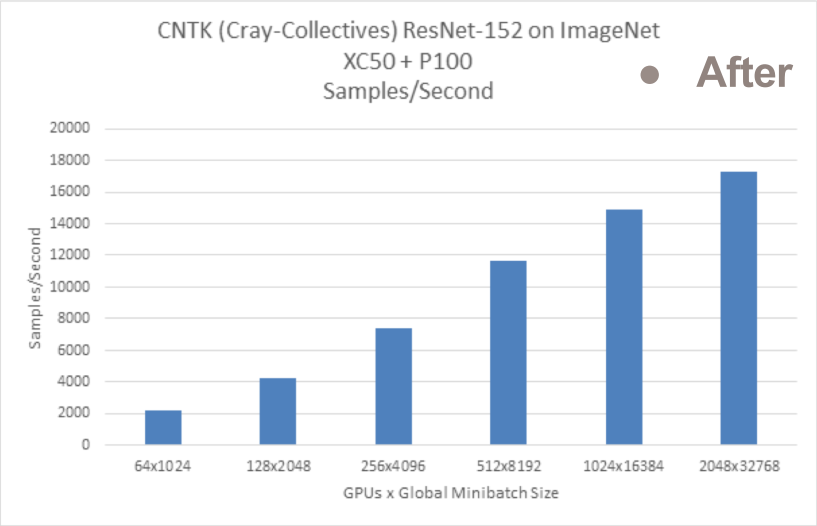


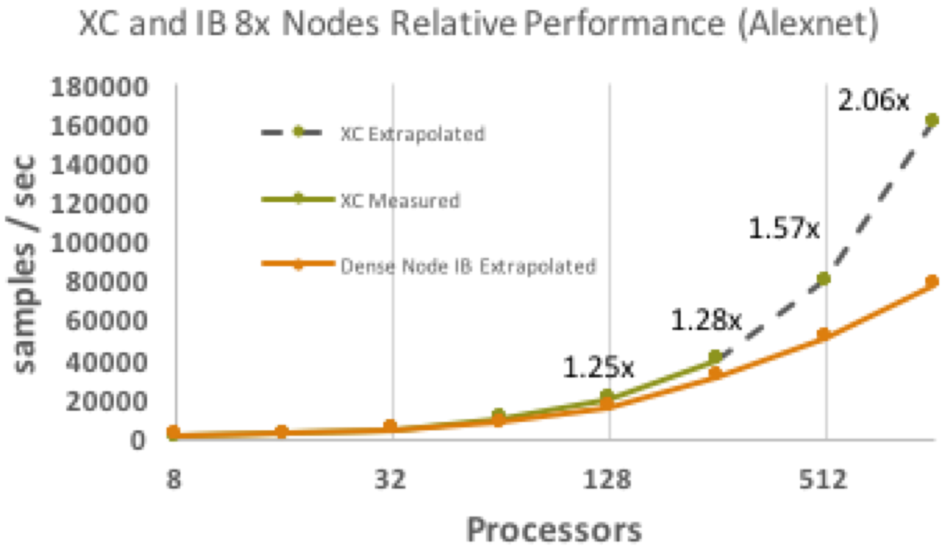
Convergence saves "flops"

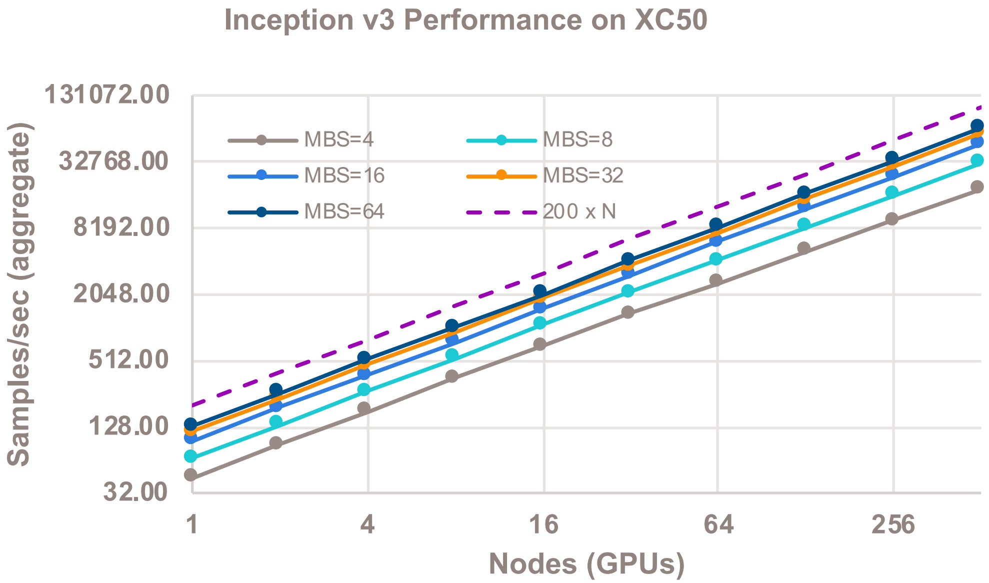
#2: DISTRIBUTED TRAINING OF AI/ML CODES

State-of-the-art: Takes 6+days to train a model









Convergence "trains" in minutes

S.R. Sukumar et al., in the Proceedings of the NIPS Workshop on Deep Learning at Supercomputing scale, 2017

#3: INTERACTIVE ANALYSIS OF BIG DATA

State-of-the-art: Exploratory data analysis is not interactive

- Cori @ NERSC
- 1630 compute nodes
- Memory: 128 GB/node,
- 32 2.3GHz Haswell cores/node

Science Area	Format/Files	Dimensions	Size
MSI	Parquet/2880	$8,258,911 \times 131,048$	1.1TB
Daya Bay	HDF5/1	$1,099,413,914 \times 192$	1.6TB
Ocean	HDF5/1	$6,349,676 \times 46,715$	2.2TB
Atmosphere	HDF5/1	$26,542,080 \times 81,600$	16TB

Gittens, Alex, et al. "Matrix factorizations at scale: A comparison of scientific data analytics in spark and C+ MPI using three case studies." , *IEEE International Conference on Big Data*.2016.

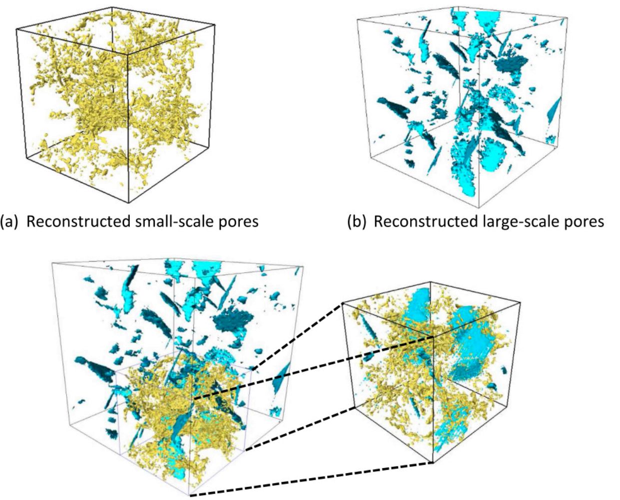
Convergence enables "iterative discovery"

	Nodes / cores	MPI Time	Spark Time	Gap
NMF	50 / 1,600	1 min 6 s	4 min 38 s	4.2x
	100 / 3,200	45 s	3 min 27 s	4.6x
	300 / 9,600	30 s	70 s	2.3x
PCA (2.2TB)	100 / 3,200	1 min 34 s	15 min 34 s	9.9x
	300 / 9,600	1 min	13 min 47 s	13.8x
	500 / 16,000	56 s	19 min 20 s	20.7x
PCA (16TB)	MPI: 1,600 / 51,200 Spark: 1,522 / 48,704	2 min 40 s	69 min 35 s	26x

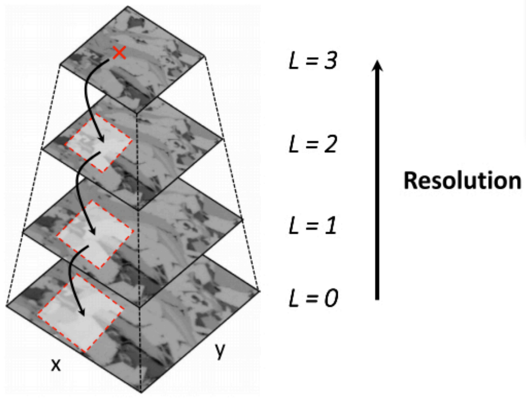
#4: HYBRID THEORY+AI MODELS

State-of-the-art: Works when manually-tuned

"Large Enough" Data + Compute Power + Advanced Algorithms and Software Frameworks + Data Science Expertise = Improved Velocity Models


(a) Reconstructed small-scale pores (b) Reconstructed large-scale pores

(c) Integrated small- and large-scale pores

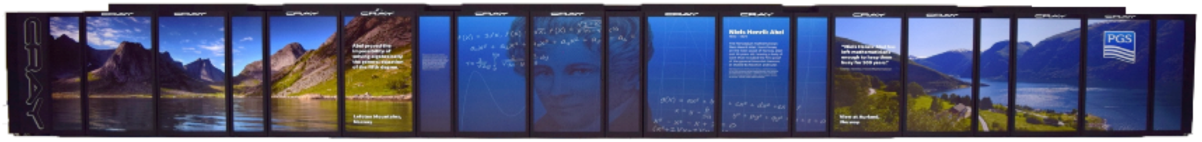


$$\min_{m} \underbrace{\left\| F(m) - d \right\|^{2}}_{\text{data misfit}} + \lambda \left(\underbrace{\left\| \Theta \nabla (m - m_r) \right\|_{L^1}}_{\text{weighted TV}} + \int_{\Omega} \underbrace{P \bullet \nabla (m - m_r)}_{\text{Steerable Variation}} \right)$$

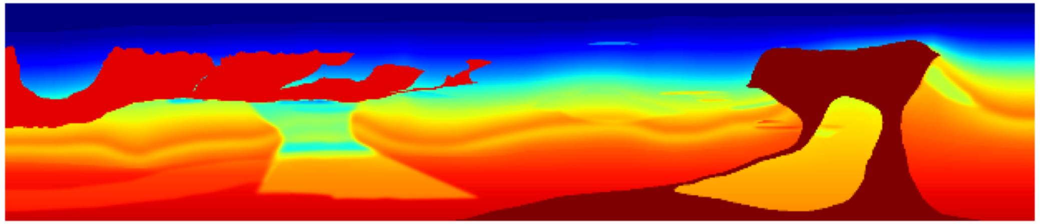
$$d_{js}(\mathbf{DI} \| \mathbf{M}) = \frac{1}{2} \int_{-\infty}^{+\infty} \mathbf{DI}_i \log \left(\frac{\mathbf{DI}_i}{\mathbf{M}_i} \right) dx + \frac{1}{2} \int_{-\infty}^{+\infty} \mathbf{M}_i \log \left(\frac{\mathbf{M}_i}{\mathbf{DI}_i} \right) dx$$

#4: HYBRID THEORY+AI MODELS

CRAY

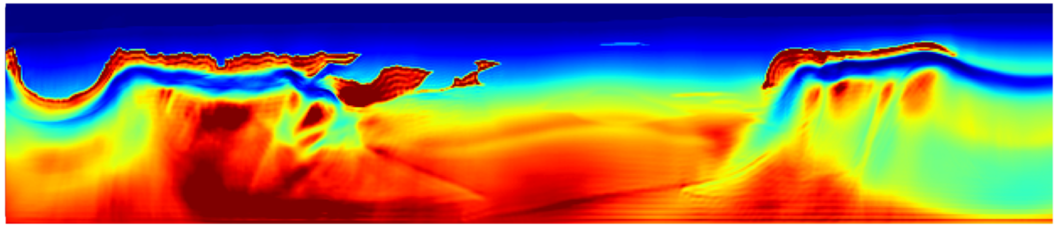


“Ground truth” Benchmark



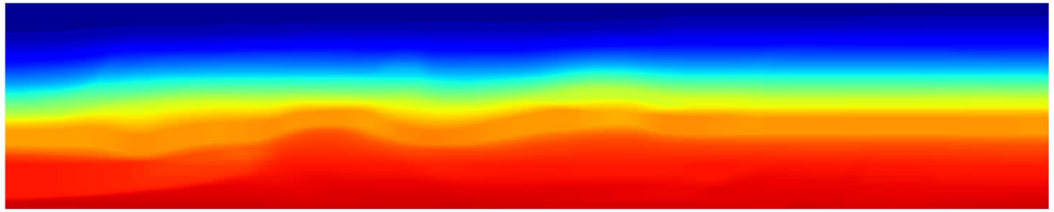
Synthetic benchmark with known subsurface geometry and seismic reflection data.

Conventional FWI



Conventional FWI attempts to derive a more accurate velocity model.

FWI with Machine Learning



Using machine learning (regularization and steering) to guide the convergence process.

Convergence enables “higher fidelity” to reality

TRENDS AND CHALLENGES

TREND: MULTI-MODEL DATA

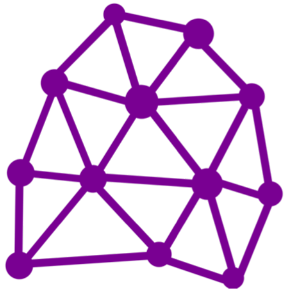
INCREASING SPATIAL, TEMPORAL AND SPECTRAL RESOLUTION

Transactional

Conversational

Genomics

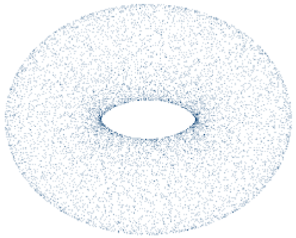
Graphs



Documents

Images

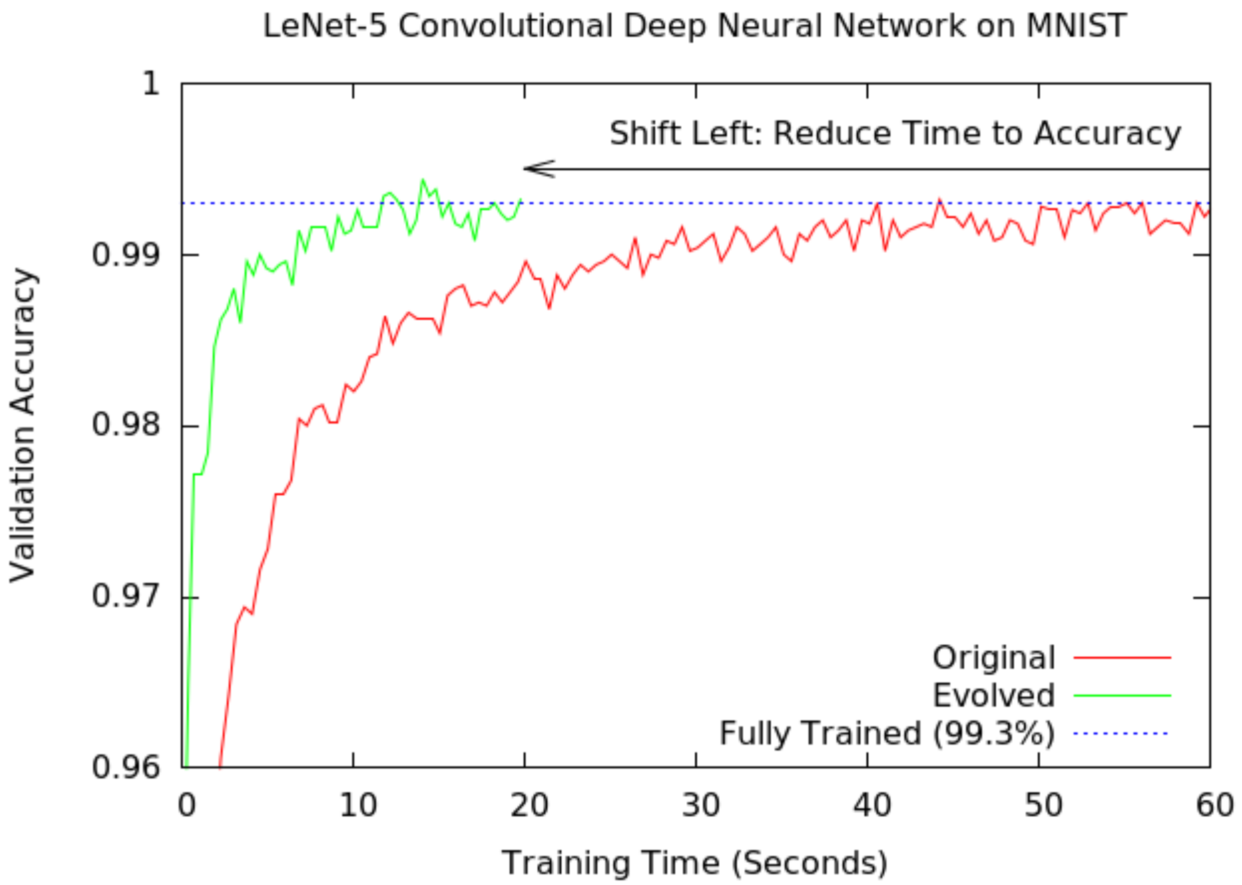
3D Point clouds



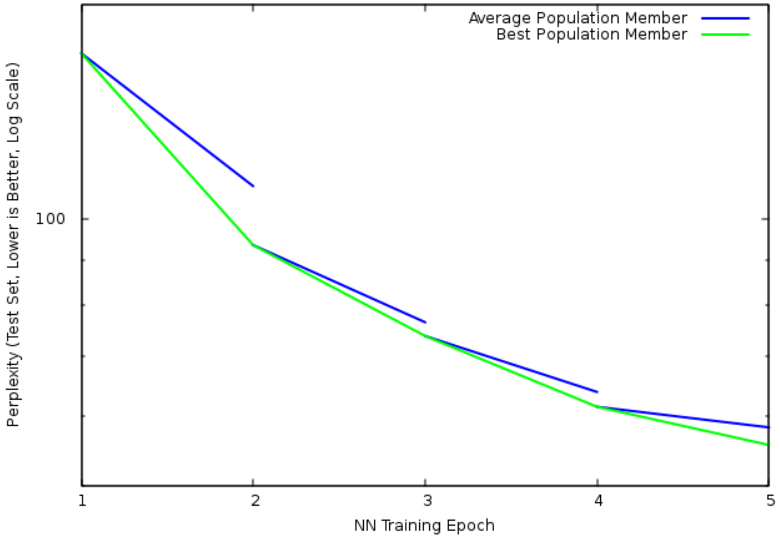
Sensors

CHALLENGE: I/O PATTERNS OF HPO LESS STUDIED

HYPERPARAMETER OPTIMIZATION CRITICAL FOR MULTI-MODEL DATA



Source: Aaron Vose, Cray Performance Team



Learning the optimal topology

Learning a “learning-rate” schedule

TREND : PROLIFERATION OF APPLIANCES

CRAY

Massively parallel processing databases | **Distributed Analytics on Storage** | **Distributed-memory Analytics**

PostgreSQL
Greenplum
Neo4j
mongoDB

hadoop
mahout
Spark
Spark MLlib

MATLAB
R
sas
Knowledge Discovery Toolbox

"Analytics is retrieval"

"Take compute to cheap storage"

"Algorithm is made to work on distributed memory-chunks"

... SOLVES THE COMPUTE PROBLEM BUT CREATES NEW I/O PROBLEMS.

CHALLENGE: NEED A SMARTER INTERCONNECT

THE DATA MOVEMENT, MANAGEMENT AND I/O PATTERNS…

Application Level

Mahout, R and Applications

Applications and Community Codes

- **Hypothetical "best" node**
  - **Mixed-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**

High-performance Interconnect

Cluster Hardware

Ethernet Switches

Local Node Storage

Commodity X86 Racks

IB+ Enet Switches

SAN+Local Storage

x86 +GPUs or Accelerators

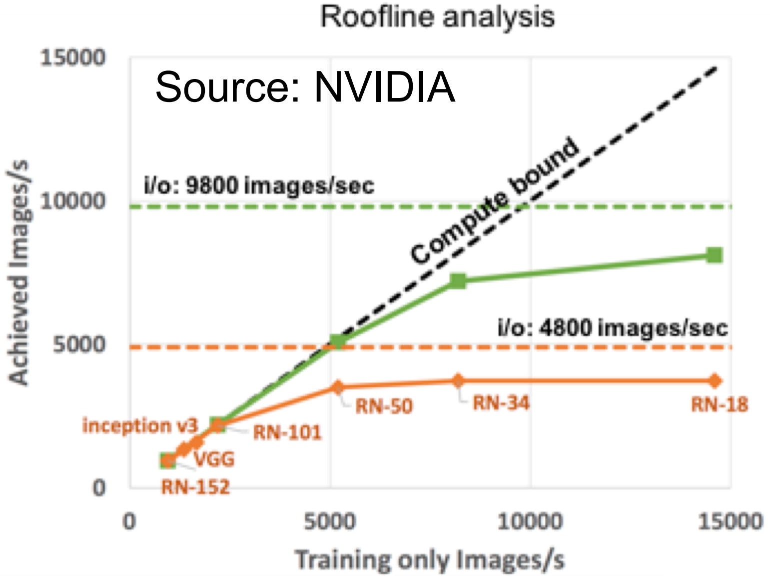
Data Analytics Ecosystem

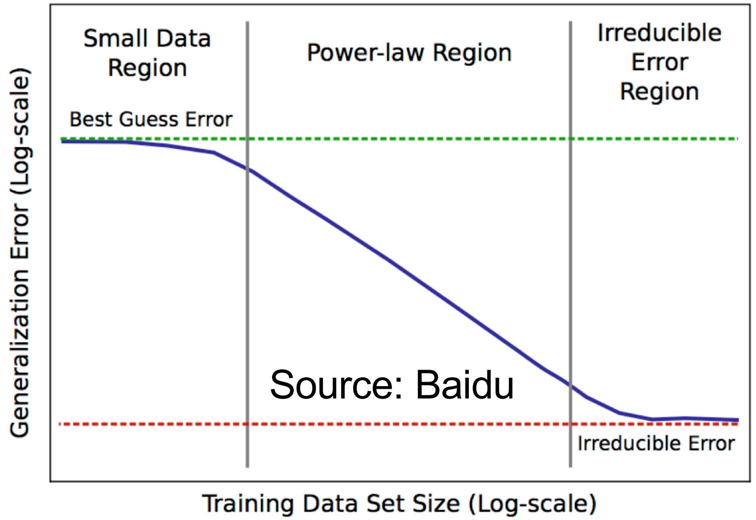
Computational Science Ecosystem

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

…WILL NEED A SMARTER INTERCONNECT.

TREND: SOFTWARE TRICKS OUTPERFORMING HARDWARE





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)

CHALLENGE: SOFTWARE FOR NEW HARDWARE

- Software : **7-10x improvement in time-to-accuracy in 1 year on CNNs**

Method	Who?
LARS (MBS – 32K)	NVIDIA
Learning Rate schedule (~64K)	Facebook
Gradient Clipping	Microsoft
Mixed Precision Training	Baidu
Optimizer Tuning (~32K) - K-FAC - Neumann	Google Research (now part of TensorFlow)
Batch Normalization	Google

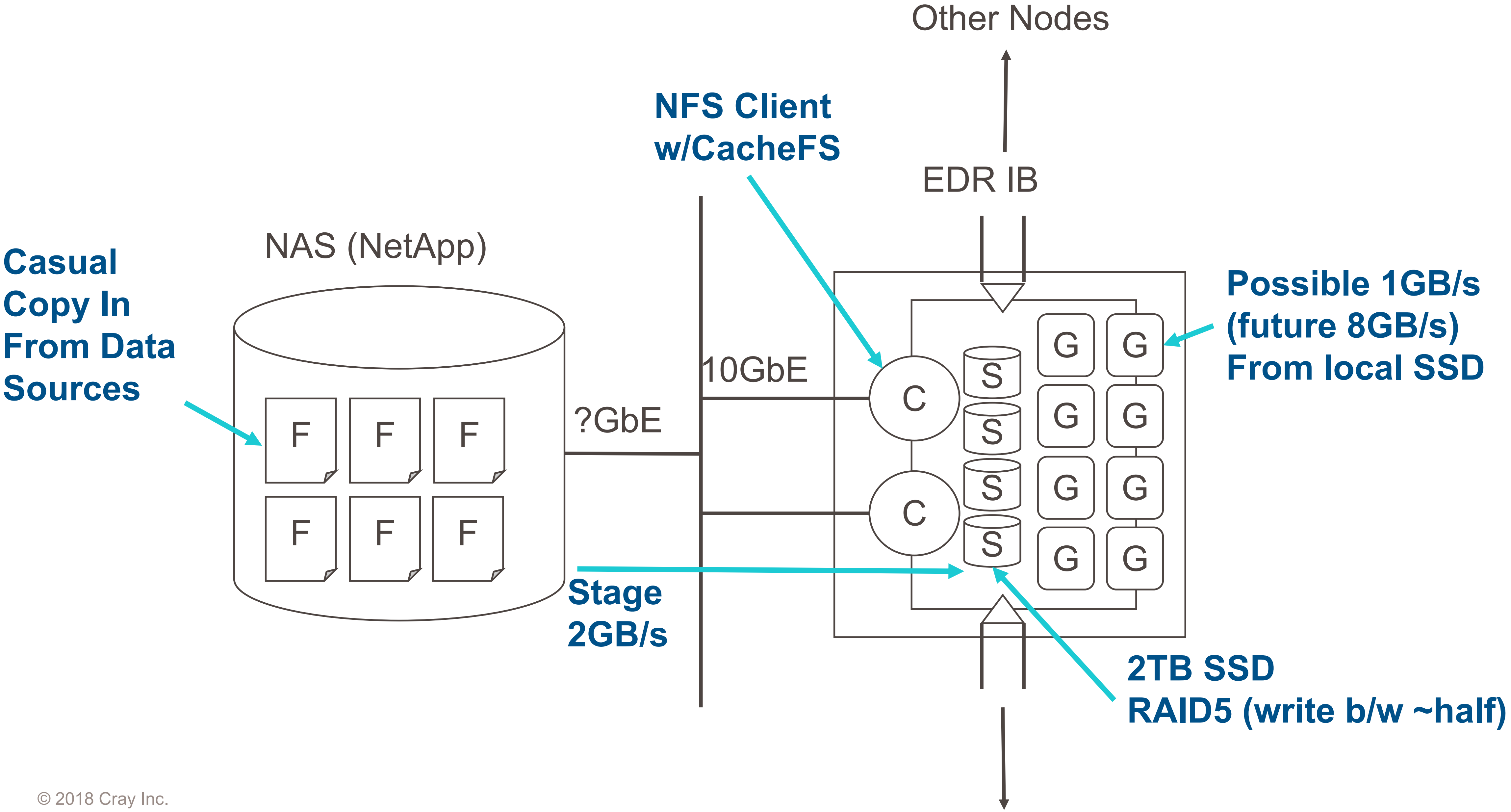
- Hardware: **10-1000x in 2 years***
  - Training
    - Intel, AMD, ARM, NVIDIA
    - Google TPU v2
    - Cerebras
    - Graphcore
    - Habana
    - **30+ startups….**
  - Inferencing
    - Intel Nervana
    - Wave Computing
    - Groq

TREND: TRIGGERED TRAINING

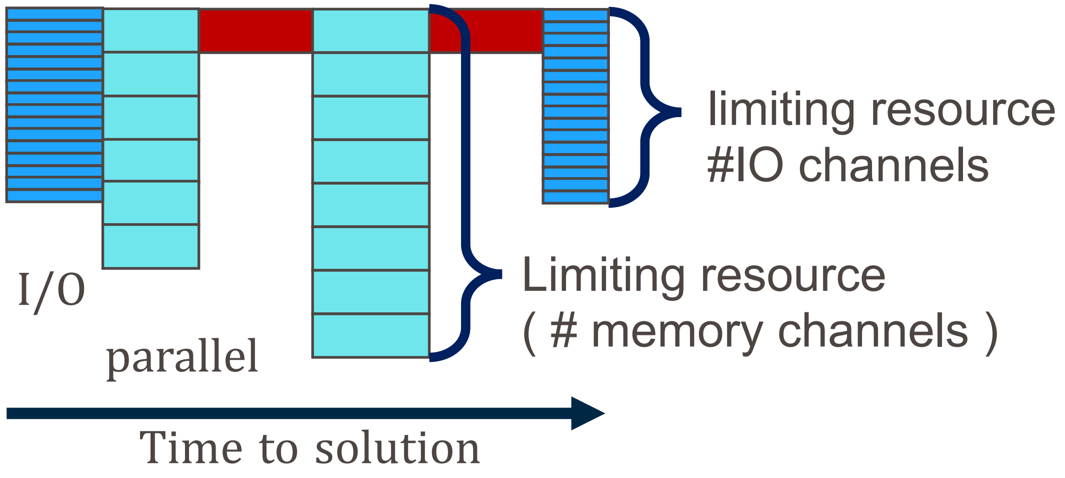
TRAINING PATTERNS DETERMINE SUPPORTING INFRASTRUCTURE FOR STORAGE AND I/O

Training	Use-case	Data size growth in unit time	Time to quality metric today	# of xPUs
Continuous	Internet-of-things	1:1	O(minutes)	O(10)
Cadence	Uber Eats prediction	n:1 (n>>1)	O(days)	O(10)
Delta	Speech (rare words)	n:1 (n~1)	O(days)	O(1)
One-time	Lower-order physics approximations	10-100n:1	O(weeks)	O(100+)
Throughput	Speech and speaker detection	1:# of users	O(days)	O(100+)

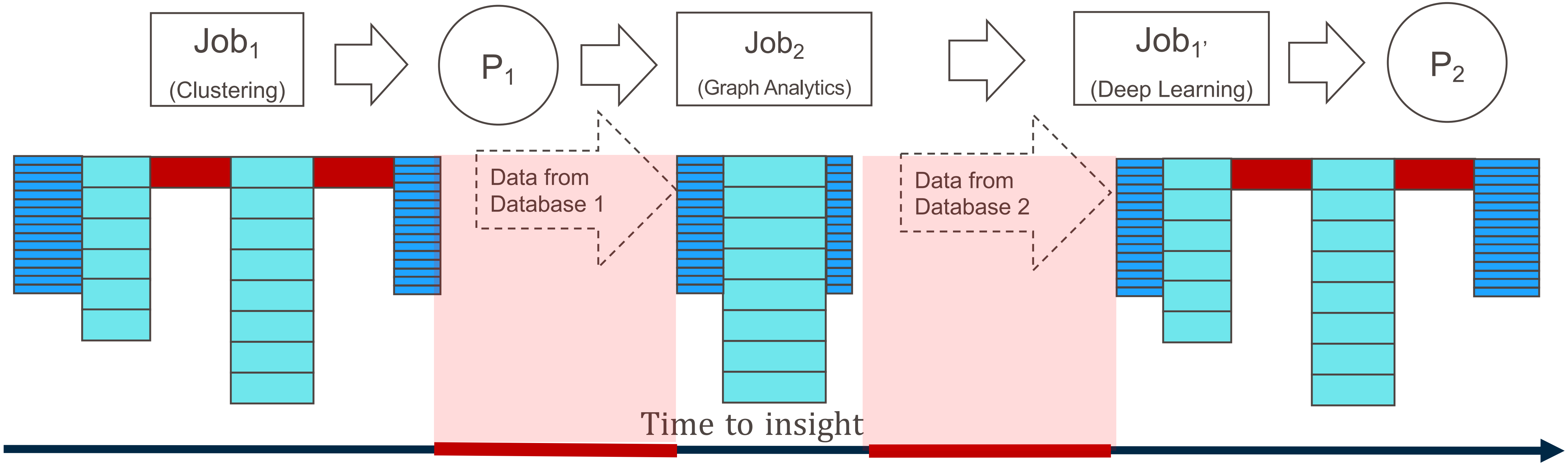
CHALLENGE: POTENTIAL OFF-NODE I/O REQUIREMENT



TREND: MULTI-TOOL WORKFLOWS ARE THE NORM

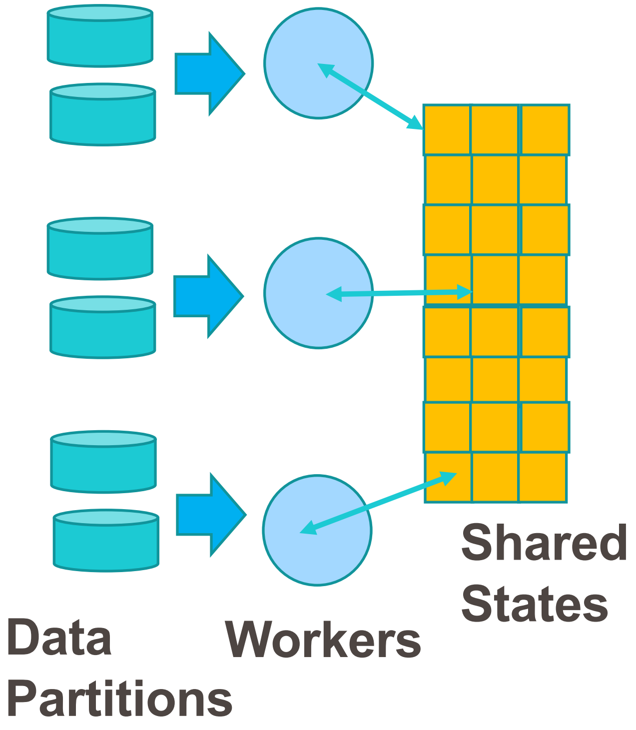


Optimized for components but not the end-to-end workflow

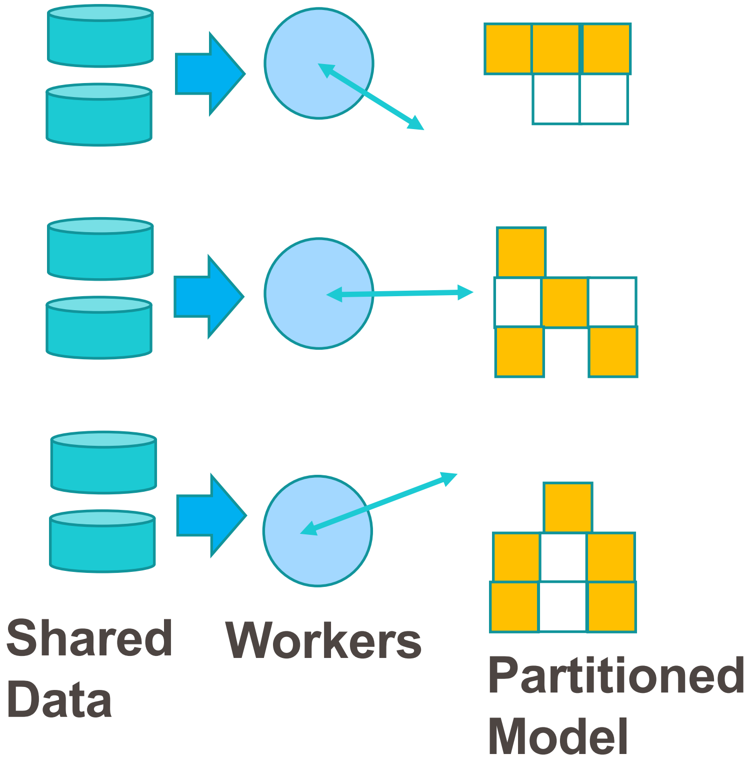


CHALLENGE: FUTURE PARALLELISM

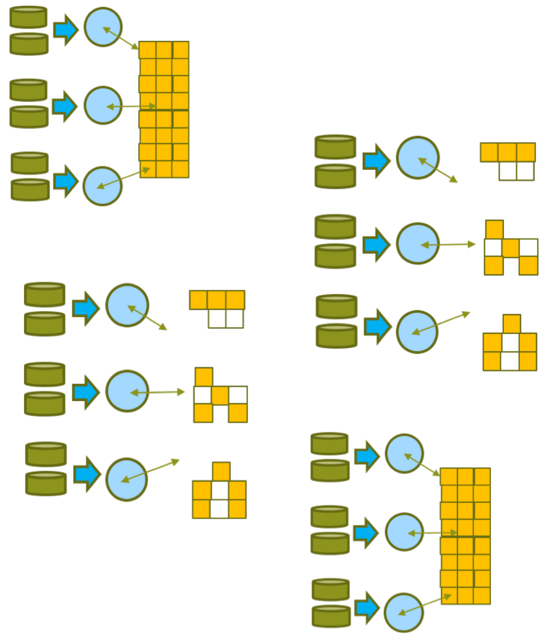
Data-Parallelism



Model-Parallelism



Ensemble-Parallelism



- Model-Parallelism (Training, Inferencing)
- Higher Resolution Images
- Intra-node vs. Inter-node bandwidth

TREND: CONVERGED HARDWARE+SOFTWARE

CRAY

Scalable simulation & modelling, analytics, and AI

CRAY URIKA-CS

CRAY CS500

CS-STORM 500NX

CS-STORM 500GT

CRAY URIKA-XC

CRAY XC SERIES

Flexibility with heterogeneity

Cloud-like interoperability

UIs: Jupyter Notebooks, TensorBoard

MLlib,Spark SQL, Spark Streaming, GraphX

BigDL

Apache Spark™

Anaconda Python, Dask

pbdR

PyTorch

Keras, TensorFlow™

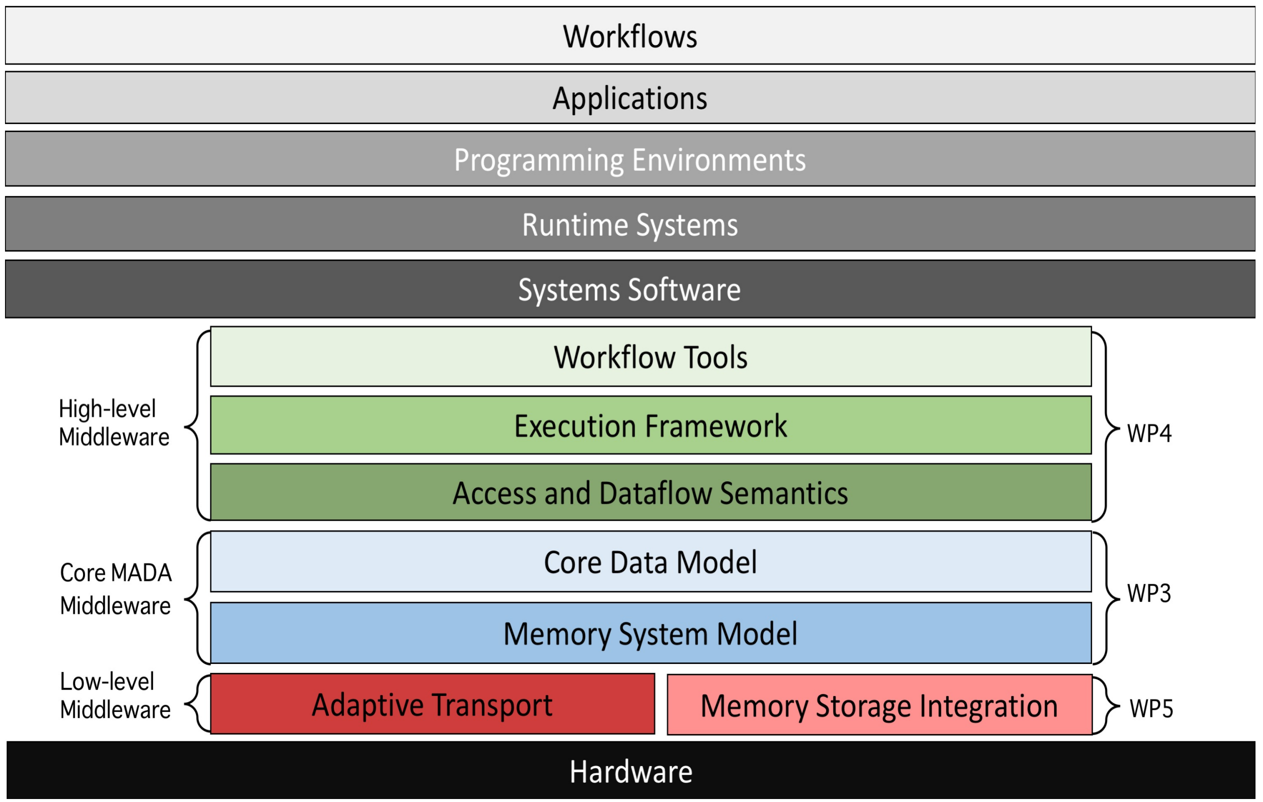
Distributed Training Framework Horovod, CrayPE ML Plugin

Cray Graph Engine (CGE)

Intel® MKL, Intel MKL-DNN, Cray MPI

CHALLENGE: CONVERGENCE REQUIRES WORK

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

Source: Adrian Tate, Cray EMEA

Lot more work before convergence can be productive….

CHALLENGE: DELIVERING A SEAMLESS EXPERIENCE

CRAY

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?

SUMMARY: SYSTEMS FOR 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

QUESTIONS?