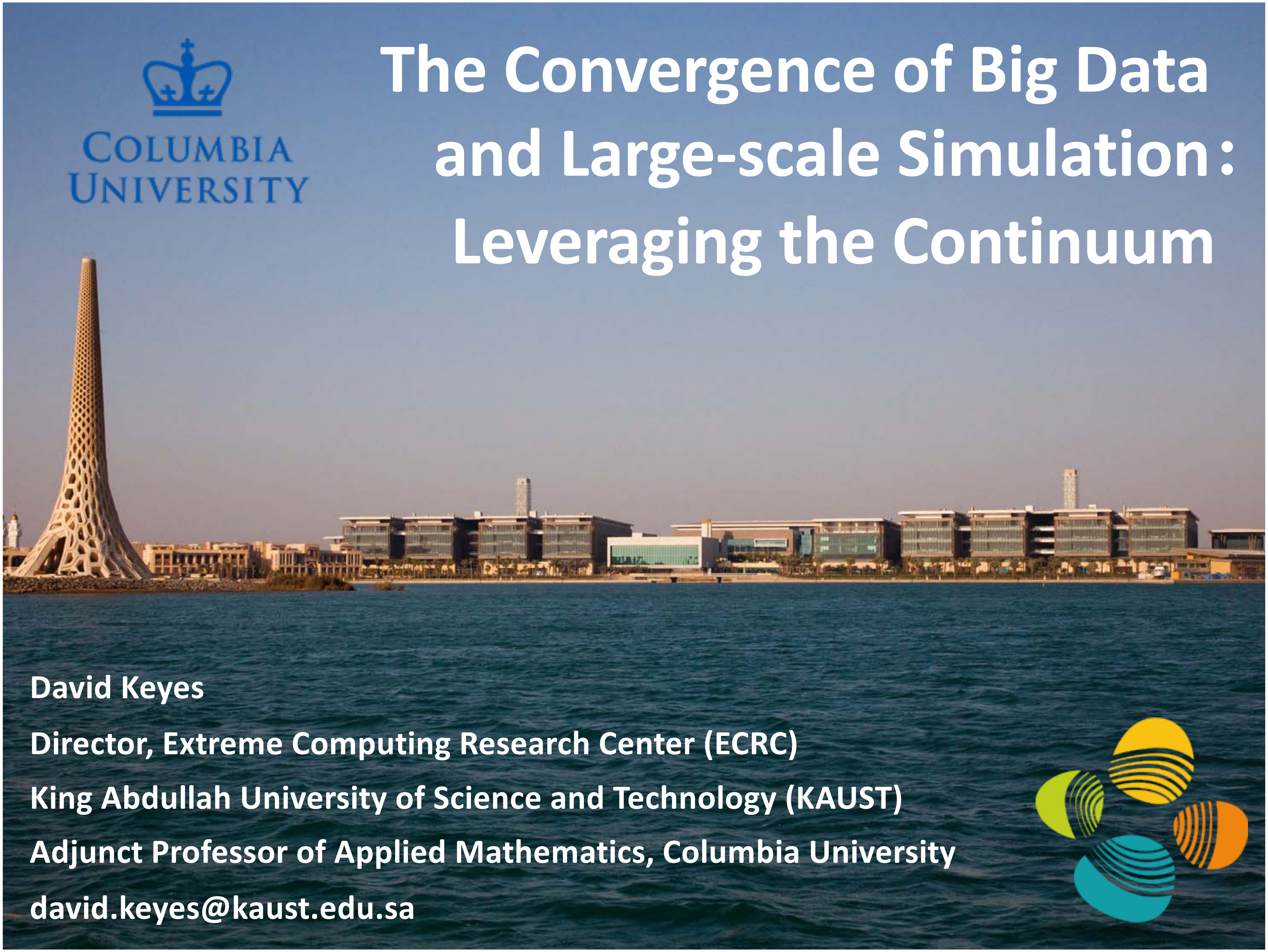


The Convergence of Big Data and Large-scale Simulation : Leveraging the Continuum

David Keyes

Director, Extreme Computing Research Center (ECRC)

King Abdullah University of Science and Technology (KAUST)

Adjunct Professor of Applied Mathematics, Columbia University

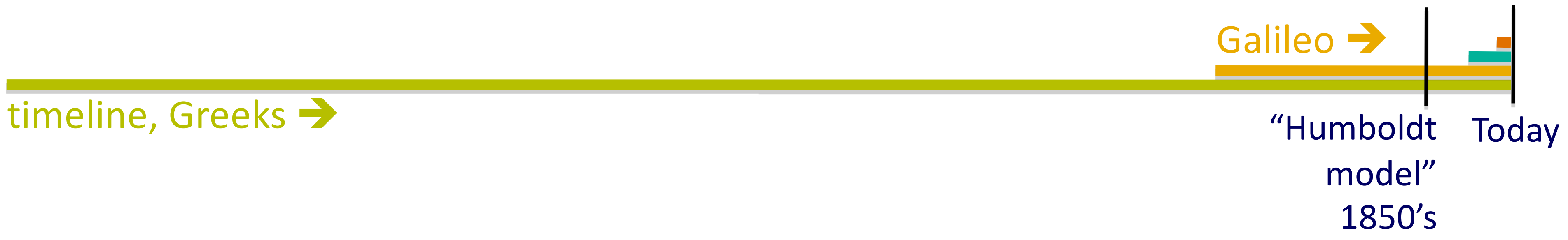
david.keyes@kaust.edu.sa

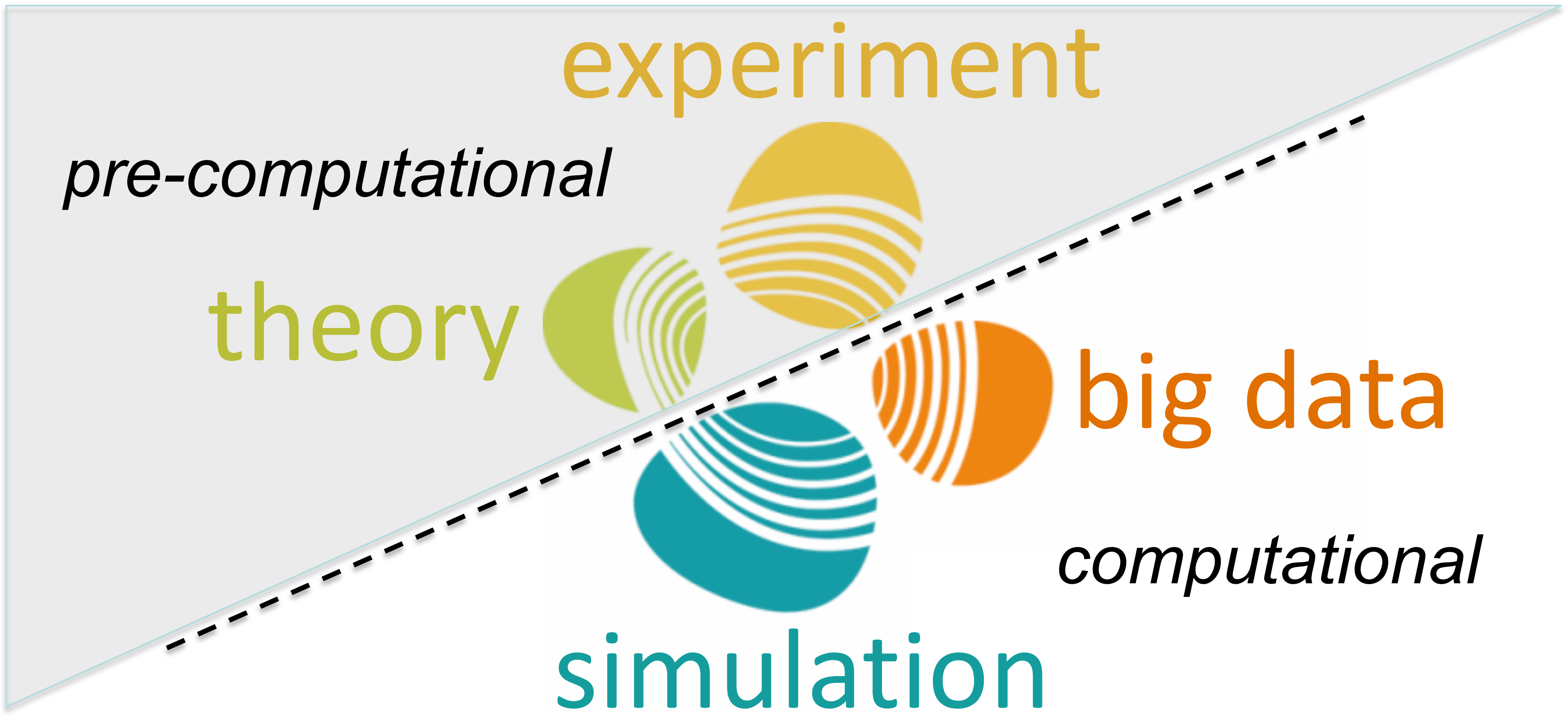
Greetings from KAUST's new President

Tony Chan, *formerly:*

- **President, HKUST**
- **Director, Div Math & Phys Sci, NSF**
- **Dean, Phys Sci, UCLA**
- **Chair, Math, UCLA**
- **Co-founder, IPAM**
- **Member, NAE**
- **Fellow, SIAM, IEEE, AAAS**
- **ISI highly cited, imaging sciences, numerical analysis**

Four paradigms for understanding





Convergence potential

- The convergence of *theory* and *experiment* in the pre-computational era launched modern science

- The convergence of *simulation* and *big data* in the exascale computational era will give humanity predictive tools to overcome our great natural and technological challenges

Convergence of 3rd and 4th paradigms

10^{18}

BDEC

BIG DATA AND EXTREME-SCALE COMPUTING

Big Data and Extreme Computing: Pathways to Convergence **(2017)**

downloadable at exascale.org

successor to the 2011 ***International Exascale Software Roadmap***

Int J High Performance Computing Applications **34**:435-479 (2018)

A vision for BDEC 2

10^18 BDEC2 BIG DATA AND EXTREME-SCALE COMPUTING 2

- Edge data is too large to collect and transmit
- Need lightweight learning at the edge: *sorting, searching, learning about the distribution*
- Edge data is pulled into the cloud to learn
- Inference model is sent back to the edge

Roles for Artificial Intelligence

- **Machine learning in the application**
  - for enhanced scientific discovery
- **Machine learning in the computational infrastructure**
  - for improved performance
- **Machine learning at the edge**
  - for managing data volume

A tale of two communities…

- **HPC: high performance computing**
  - grew up around Moore’s Law multiplied by massive parallelism
  - predictive on par with experiments (e.g., Nobel prizes in chemistry)
  - recognized for policy support (e.g., nuclear weapons, climate treaties)
  - recognized for decision support (e.g., oil drilling, therapy planning)
- **HDA: high-end data analytics**
  - grew up around open source tools (e.g., Hadoop) from online search and service providers
  - created trillion-dollar market in analyzing human preferences
  - now dictating the design of network and computer architecture
  - now transforming university curricula and national investments
  - now migrating to scientific data, evolving as it goes

Pressure on HPC

- Vendors, even those responding to the lucrative call for exascale systems by government, must leverage their technology developments for the much larger data science markets
- This includes exploitation of lower precision floating point pervasive in deep learning applications
- Fortunately, the concerns are the same:
  - energy efficiency
  - limited memory per core
  - limited memory bandwidth per core

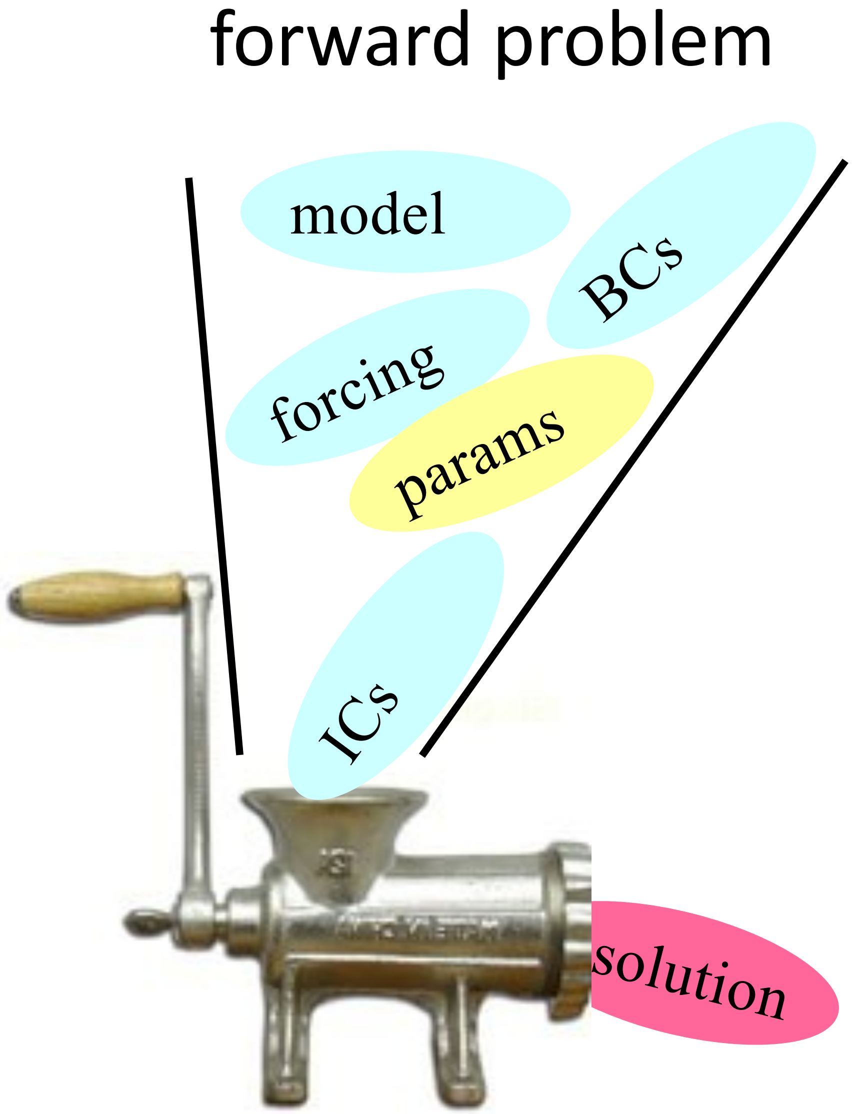
Pressure on HDA

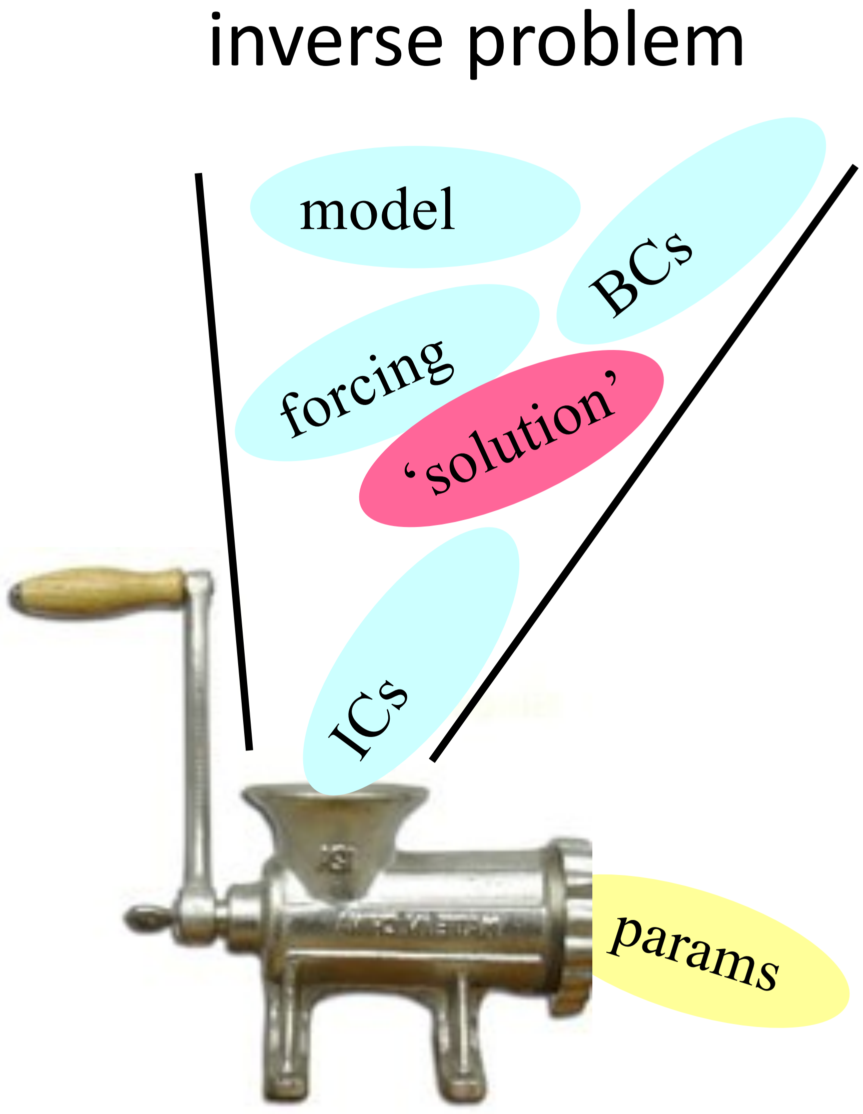
- Since the beginning of the big data age, data has been moved over “stateless” networks
  - routing is based on address bits in the data packets
  - no system-wide coordination of data sets or buffering
- Workarounds coped with volume but are now creaking
  - ftp mirror sites, web-caching (e.g., Akamai out of MIT)
- Solutions for buffering massive data sets from the HPC “edge” …
  - seismic arrays, satellite networks, telescopes, scanning electron microscopes, beamlines, sensors, drones, etc.
- …will be useful for the “fog” environments of the big data “cloud”

Some BDEC report findings

- Many motivations to bring together large-scale simulation and big data analytics (“convergence”)
- Should be combined *in situ*
  - pipelining between simulation and analytics through disk files with sequential applications leaves too many benefits “on the table”
- Many hurdles to convergence of HPC and HDA
  - but ultimately, this will not be a “forced marriage”
- Science and engineering may be minority users of “big data” (today and perhaps forever) but can become leaders in the “big data” community
  - by harnessing high performance computing
  - being pathfinders for other applications, once again!

Traditional combination of 3rd/4th paradigms: from forward to inverse problem





+ regularization

Traditional combination of 3rd/4th paradigms: data assimilation

Theory

Fully Nonlinear Filters

Dual Filters Coupled Models

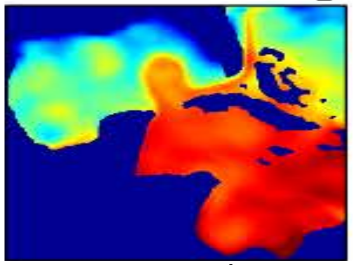
Robust Ensemble Filters

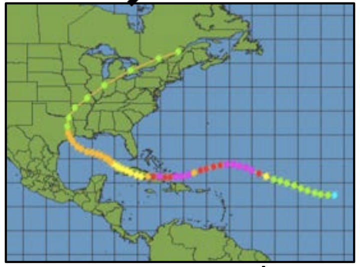
Hybrid Adjoint-Ensemble Filters

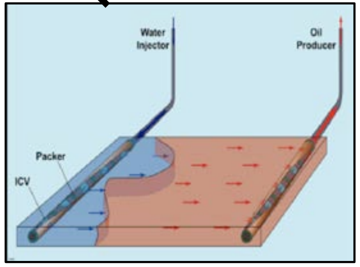
Bayesian Filtering

Data Assimilation

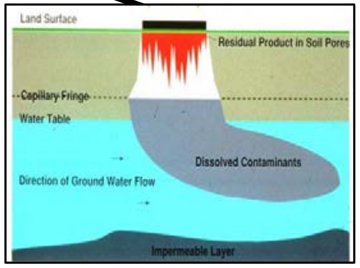
Applications


Ocean Circulation


Storm Surge Prediction



Reservoir Exploitation



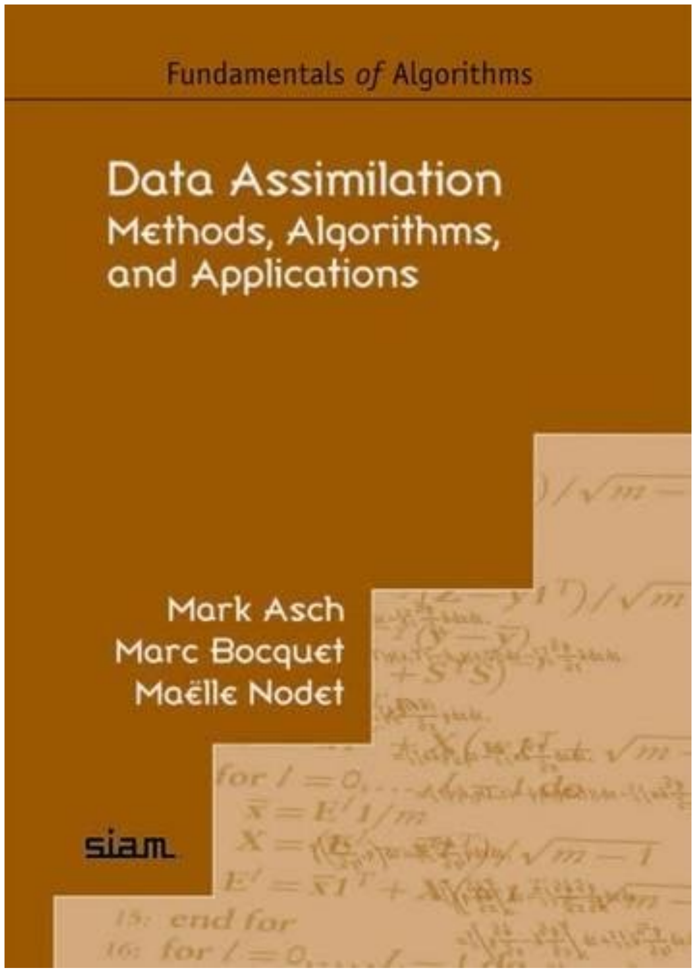
Contaminant Transport

c/o I. Hoteit, KAUST

My definition of data assimilation

“When two ugly parents have a beautiful child”

A beautiful book


Fundamentals of Algorithms
Data Assimilation
Methods, Algorithms, and Applications
Mark Asch
Marc Bocquet
Maëlle Nodet
siam

Coming interactions between paradigms

opportunities of *in situ* convergence

		To Simulation	To Analytics	To Learning
3rd	Simulation provides	—		
4th (a)	Analytics provides		—	
4th (b)	Learning provides			—

Table 1 from the BDEC Report

Coming interactions between paradigms

opportunities of *in situ* convergence

		To Simulation	To Analytics	To Learning
3rd	Simulation provides	—		
4th (a)	Analytics provides	Steering in high dimensional parameter space; *In situ* processing	—	
4th (b)	Learning provides	Smart data compression; Replacement of models with learned functions		—

Coming interactions between paradigms

opportunities of *in situ* convergence

		To Simulation	To Analytics	To Learning
3rd	Simulation provides	–	Physics-based “regularization”	Data for training, augmenting real-world data
4th (a)	Analytics provides	Steering in high dimensional parameter space; *In situ* processing	–	
4th (b)	Learning provides	Smart data compression; Replacement of models with learned functions		–

Coming interactions between paradigms

opportunities of *in situ* convergence

		To Simulation	To Analytics	To Learning
3rd	Simulation provides	–	Physics-based "regularization"	Data for training, augmenting real-world data
4th (a)	Analytics provides	Steering in high dimensional parameter space; *In situ* processing	–	Feature vectors for training
4th (b)	Learning provides	Smart data compression; Replacement of models with learned functions	Imputation of missing data; Detection and classification	–

Convergence for performance

- It is not only the HPC *application* that benefits from convergence
- *Performance tuning* of the HPC hardware-software environment also will benefit
  - iterative linear solvers, alone, have a dozen or more problem- and architecture-dependent tuning parameters that cannot be set automatically, but can be learned
  - nonlinear solvers have additional parameters
  - emerging architectures have a complex memory hierarchy of many modes for which optimal data placement can be learned

To good to be practical?

If

the convergence of theory and experiment in the pre-computational era launched modern science

And If

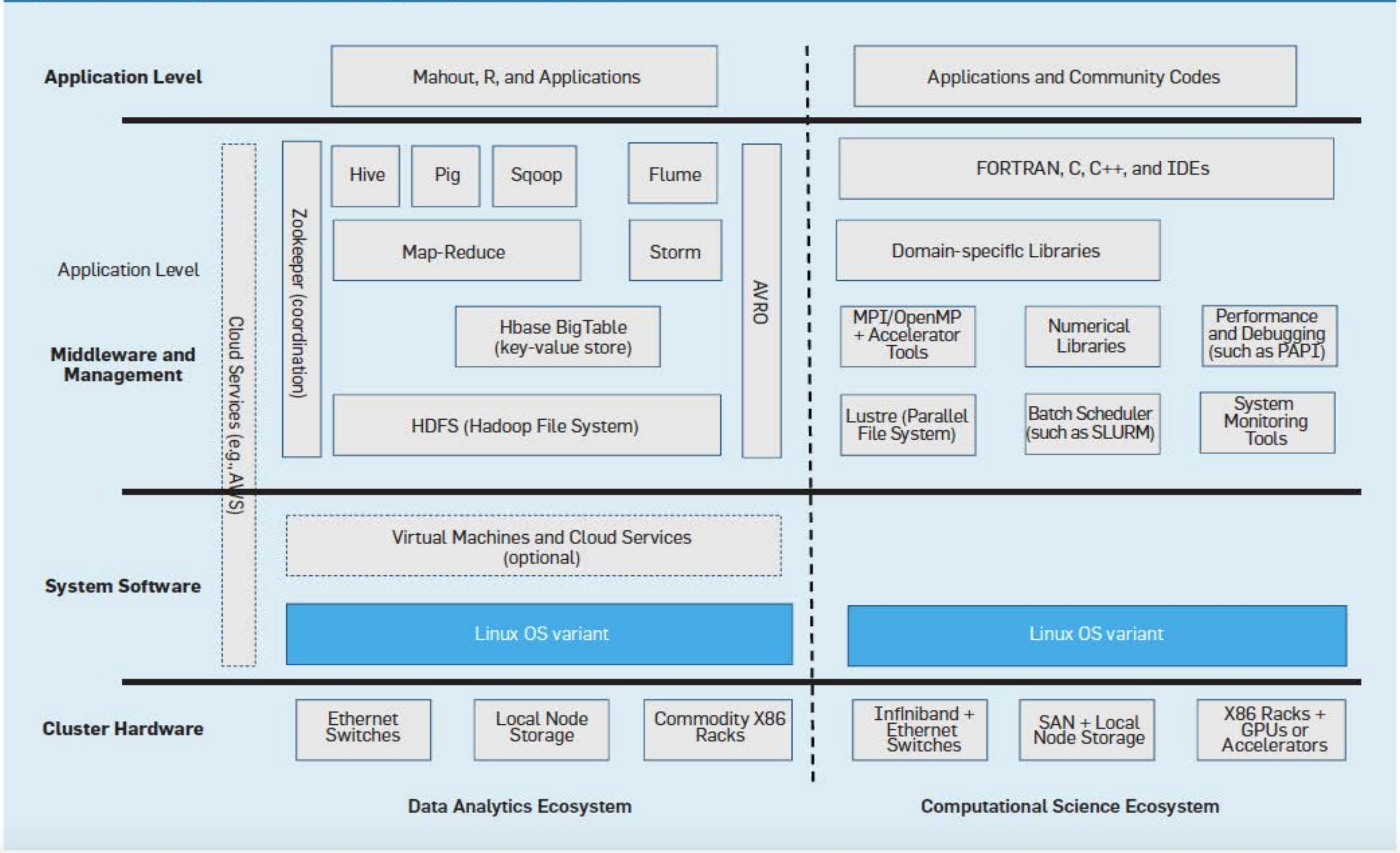
the convergence of simulation and big data in the exascale computational era has potential for similar impact

Then

What are the challenges?

Software of the 3rd and 4th paradigms

Figure 1. Data analytics and computing ecosystem compared.


Application Level
Mahout, R, and Applications
Applications and Community Codes
Application Level
Middleware and Management
Cloud Services (e.g., AWS)
Zookeeper (coordination)
Hive
Pig
Sqoop
Flume
Map-Reduce
Storm
AVRO
Hbase BigTable (key-value store)
HDFS (Hadoop File System)
FORTRAN, C, C++, and IDEs
Domain-specific Libraries
MPI/OpenMP + Accelerator Tools
Numerical Libraries
Performance and Debugging (such as PAPI)
Lustre (Parallel File System)
Batch Scheduler (such as SLURM)
System Monitoring Tools
System Software
Virtual Machines and Cloud Services (optional)
Linux OS variant
Linux OS variant
Cluster Hardware
Ethernet Switches
Local Node Storage
Commodity X86 Racks
Infiniband + Ethernet Switches
SAN + Local Node Storage
X86 Racks + GPUs or Accelerators
Data Analytics Ecosystem
Computational Science Ecosystem

c/o Reed & Dongarra, Comm. ACM, July 2015

Divergent features

- Software stacks
- Computing facilities
  - execution and storage policies
- Research communities
  - conferences, and journals
- University curricula
  - next generation workforce
- *Some* hardware forcings
  - natural precisions, specialty instructions

…divergent not only in software stacks

- Data ownership

  HPC: *generally* private   HDA: *often* curated by community

- Data access

  HPC: bulk access, fixed   HDA: fine-grained access, elastic

- Data storage

  HPC: local, temporary   HDA: cloud-based, persistent

…divergent not only in software stacks

- **Scheduling policies**

HPC: batch	HDA: interactive
HPC: exclusive space	HDA: shared space

- **Community premiums**

HPC: capability, reliability	HDA: capacity, resilience

- **Hardware infrastructure**

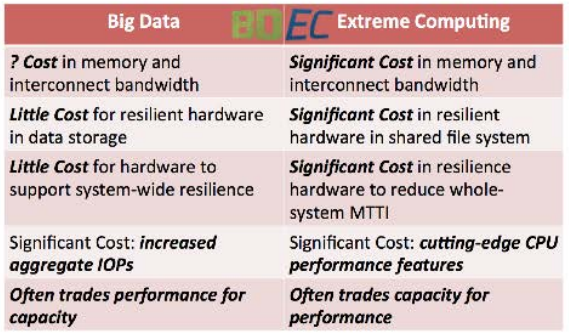
HPC: “fork-lift upgrades”	HDA: incremental upgrades

Early BDEC workshop slide: many other divergent aspects

left side of each chart

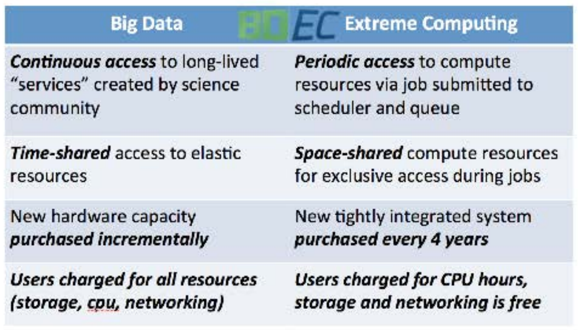
right side of each chart

Comparing Architecture



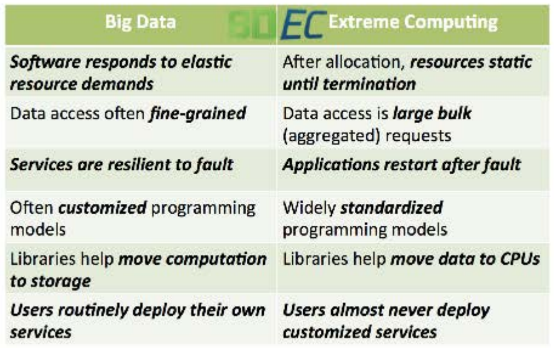
Big Data	Extreme Computing
? Cost in memory and interconnect bandwidth	***Significant Cost*** in memory and interconnect bandwidth
Little Cost for resilient hardware in data storage	***Significant Cost*** in resilient hardware in shared file system
Little Cost for hardware to support system-wide resilience	***Significant Cost*** in resilience hardware to reduce whole-system MTTI
Significant Cost: ***increased aggregate IOPs***	Significant Cost: ***cutting-edge CPU performance features***
Often trades performance for capacity	***Often trades capacity for performance***

Comparing Operations



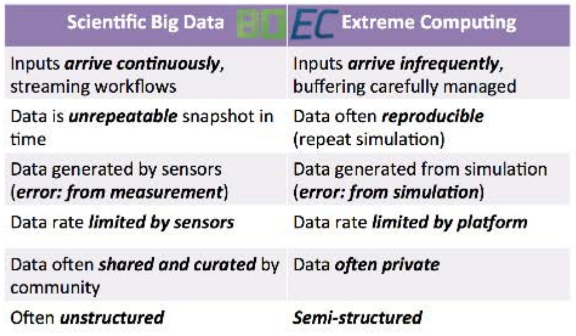
Big Data	Extreme Computing
Continuous access to long-lived "services" created by science community	***Periodic access*** to compute resources via job submitted to scheduler and queue
Time-shared access to elastic resources	***Space-shared*** compute resources for exclusive access during jobs
New hardware capacity ***purchased incrementally***	New tightly integrated system ***purchased every 4 years***
Users charged for all resources (storage, cpu, networking)	***Users charged for CPU hours, storage and networking is free***

Comparing Software



Big Data	Extreme Computing
Software responds to elastic resource demands	After allocation, ***resources static until termination***
Data access often ***fine-grained***	Data access is ***large bulk*** (aggregated) requests
Services are resilient to fault	***Applications restart after fault***
Often ***customized*** programming models	Widely ***standardized*** programming models
Libraries help ***move computation to storage***	Libraries help ***move data to CPUs***
Users routinely deploy their own services	***Users almost never deploy customized services***

Comparing Data



Scientific Big Data	Extreme Computing
Inputs ***arrive continuously***, streaming workflows	Inputs ***arrive infrequently***, buffering carefully managed
Data is ***unrepeatable*** snapshot in time	Data often ***reproducible*** (repeat simulation)
Data generated by sensors (***error: from measurement***)	Data generated from simulation (***error: from simulation***)
Data rate ***limited by sensors***	Data rate ***limited by platform***
Data often ***shared and curated*** by community	Data ***often private***
Often ***unstructured***	***Semi-structured***

following J. Ahrens, LANL

Extra motivations for convergence

- **Vendors wish to unify their offerings**
  - traditionally 3rd paradigm-serving vendors are now market-dominated by the 4th
- **Under all hardware scenarios, data movement is much more expensive than computation**
  - simulation and analytics should be done *in situ*, with each other on in-memory (in-cache?) data
  - exchange in the form of exchange of files between 3rd and 4th phrases is unwieldy

HPC benefits from visualization “the oldest form of HDA”

- Results of simulation may be unusable or less valuable without fast-turnaround viz
- Simulations at scale can be very expensive; don’t want to waste an unmonitored one that has gone awry
- Want to be able to steer

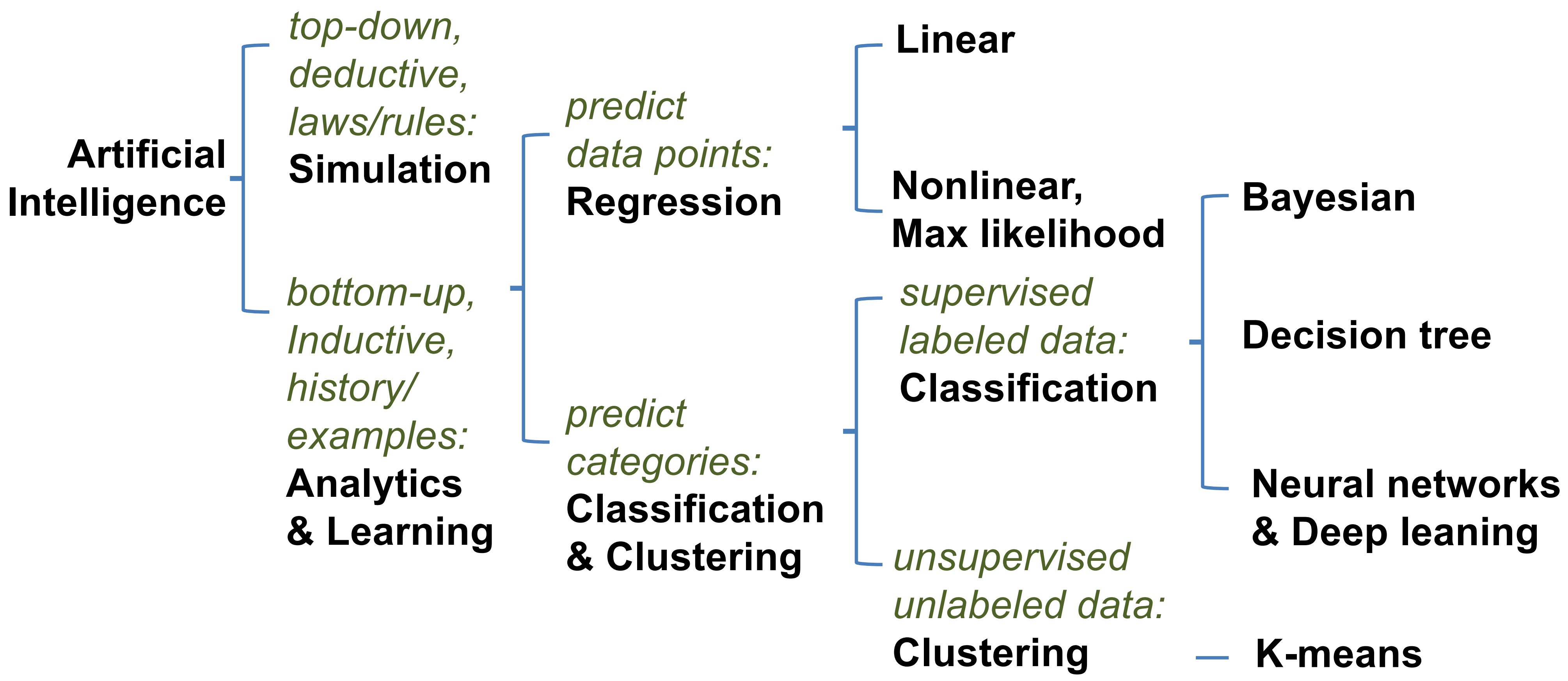
Visualization benefits from HPC

- Many visualization demands are real-time or put a premium on time-to-solution
  - there may be a viz-based human decision based in the loop
  - high performance may be required, or viz will dominate
- By the time simulations scale, all of their global data structure kernels must scale
  - e.g., linear solvers, stencil application, graph searches
  - some of the same kernels are required in visualization

Multiple classes of “big data”

- In scientific big data, different solutions may be natural for three different categories:
  - data arriving from edge devices (often in real time, e.g., beamlines) that is never centralized but processed on the fly
  - federated multi-source data (e.g., bioinformatics) intended for “permanent” archive
  - combinations of data retrieved from archival source and dynamic data from a simulation (e.g., assimilation in climate/weather)
- “Pathways” report addresses these challenges in customized sections

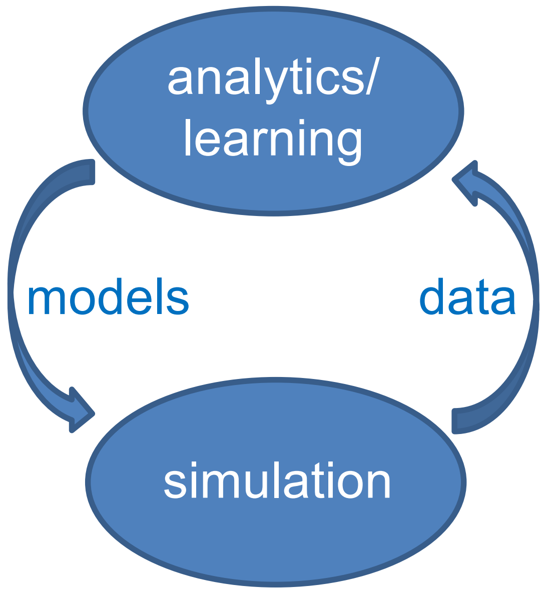
AI classification (unconventional)


Artificial Intelligence
top-down, deductive, laws/rules: **Simulation**
bottom-up, Inductive, history/ examples: **Analytics & Learning**
predict data points: **Regression**
predict categories: **Classification & Clustering**
Linear
Nonlinear, Max likelihood
supervised labeled data: **Classification**
unsupervised unlabeled data: **Clustering**
Bayesian
Decision tree
Neural networks & Deep leaning
K-means

after Eng Lim Goh (Chief Technologist, HPE)

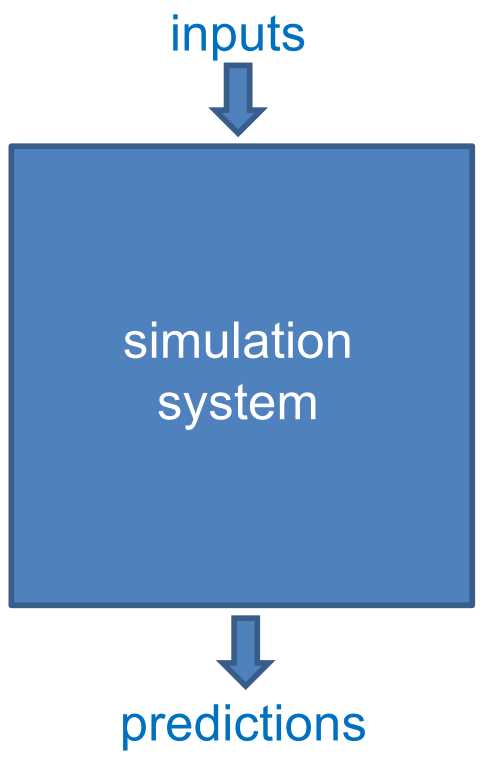
Simulation and analytics: a cute pair

- **Both simulation and analytics include both models and data**
  - simulation uses a model (mathematical) to produce data
  - analytics uses data to produce a model (statistical)
- **Models generated by analytics can be used in simulation**
  - not the only source of models, of course
- **Data generated by simulation can be used in analytics**
  - not the only source of data, of course
- A virtuous cycle can be set up



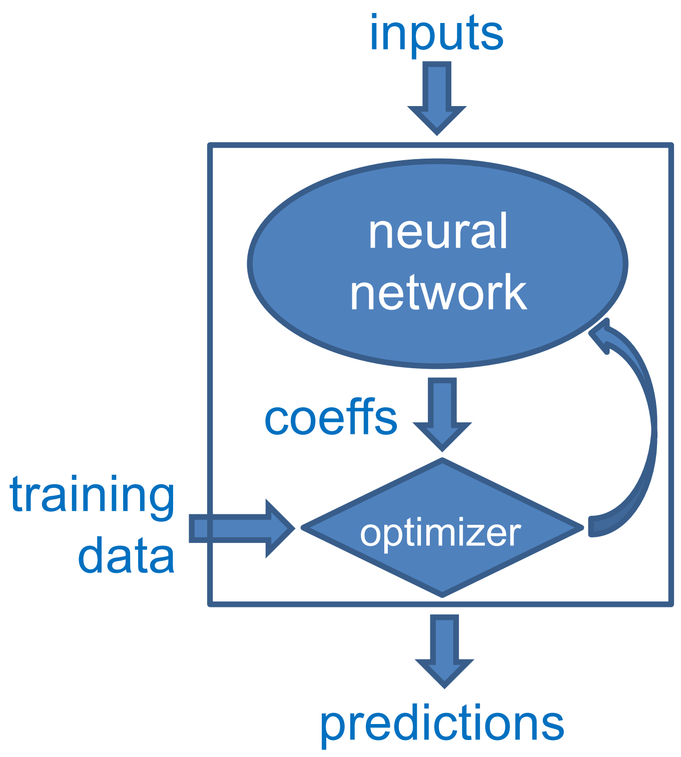
Simulation and learning: difference

- Primary novelty in machine-based “intelligence” is the learning part
- A simulation system is historically a fixed, human-engineered code that does not improve with the flow of data through it



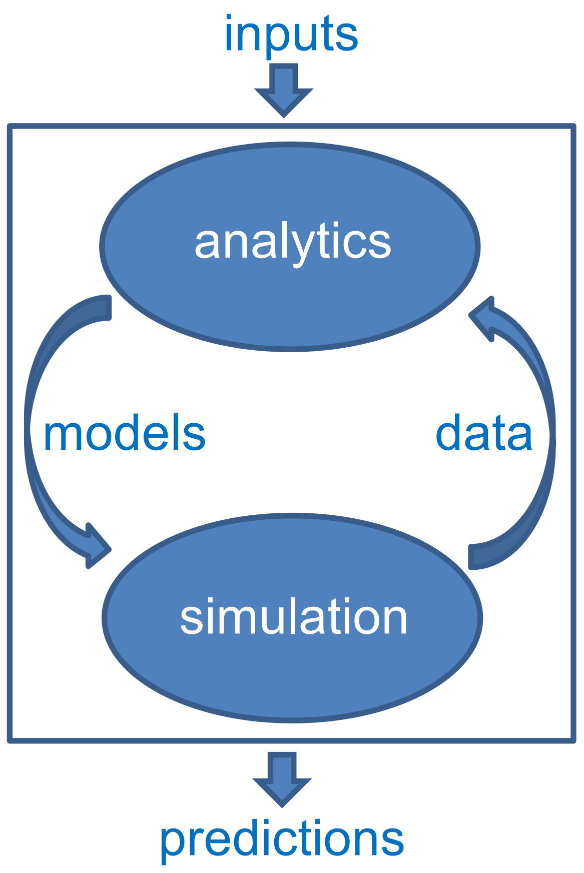
Simulation and learning: difference

- Primary novelty in machine-based "intelligence" is the learning part
- Machine learning systems improve as they ingest data
  - make inferences and decisions on their own
  - actually generate the model
- Of course, as with a child, when provided with information, a machine may learn incorrect rules and make incorrect decisions



An *in situ* converged system

- Including learning in the simulation loop can enhance the predictivity of the simulation
- Including both simulation data and observational data in the learning loop can enhance the learning
- Ultimately a win-win marriage



“Scientific method on steroids”

SC18
Dallas, TX
hpc inspires.
The International Conference for High Performance Computing, Networking, Storage, and Analysis

The “steroids” are high performance computing technologies

- Big data paper won Gordon Bell Prize for first time
- Half of the Gordon Bell finalists in big data

A new instrument is emerging!

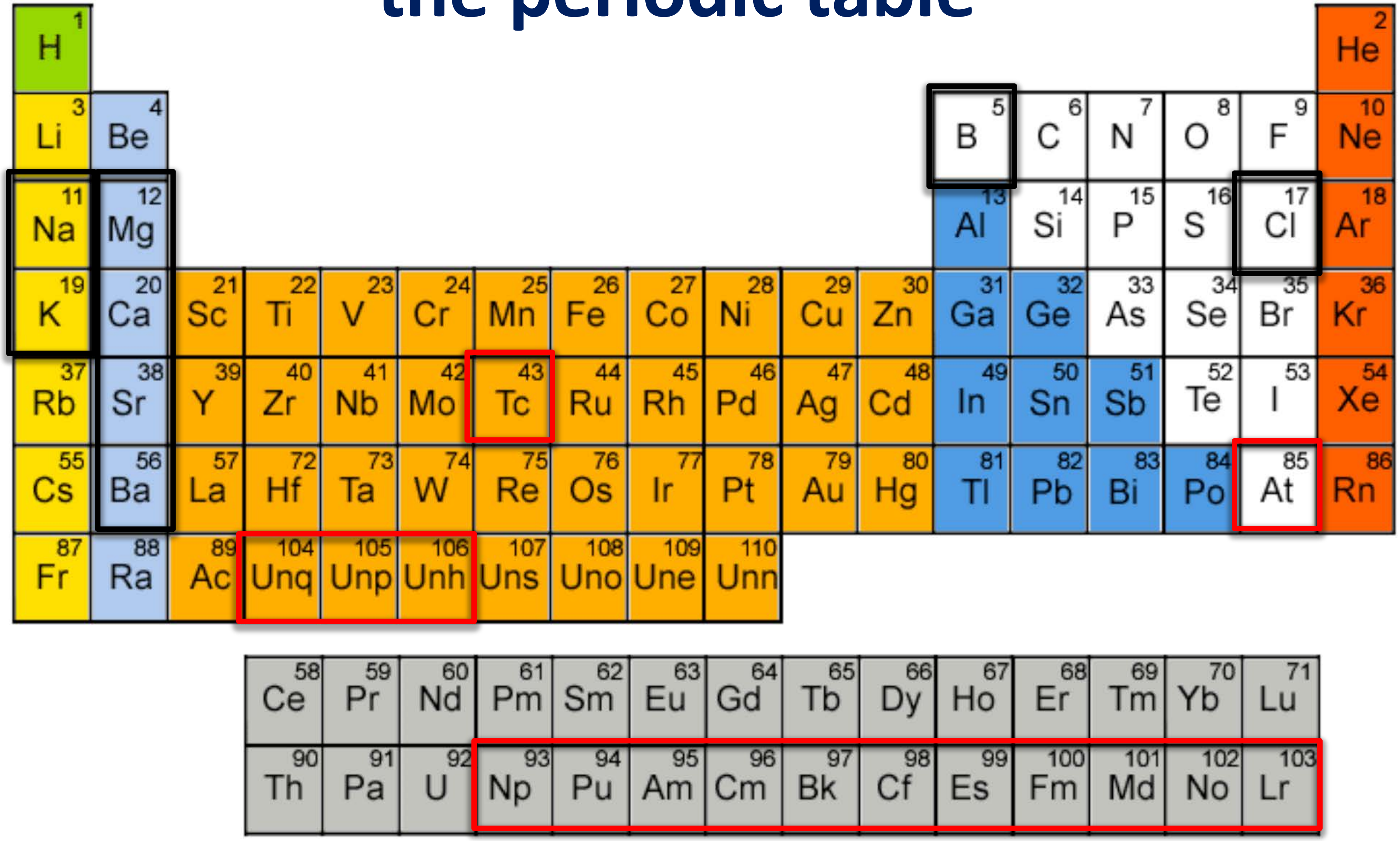
“Nothing tends so much to the advancement of knowledge as the application of a new instrument.

The native intellectual powers of people in different times are not so much the causes of the different success of their labors, as the peculiar nature of the means and artificial resources in their possession.”

— Humphrey Davy (1778-1829)

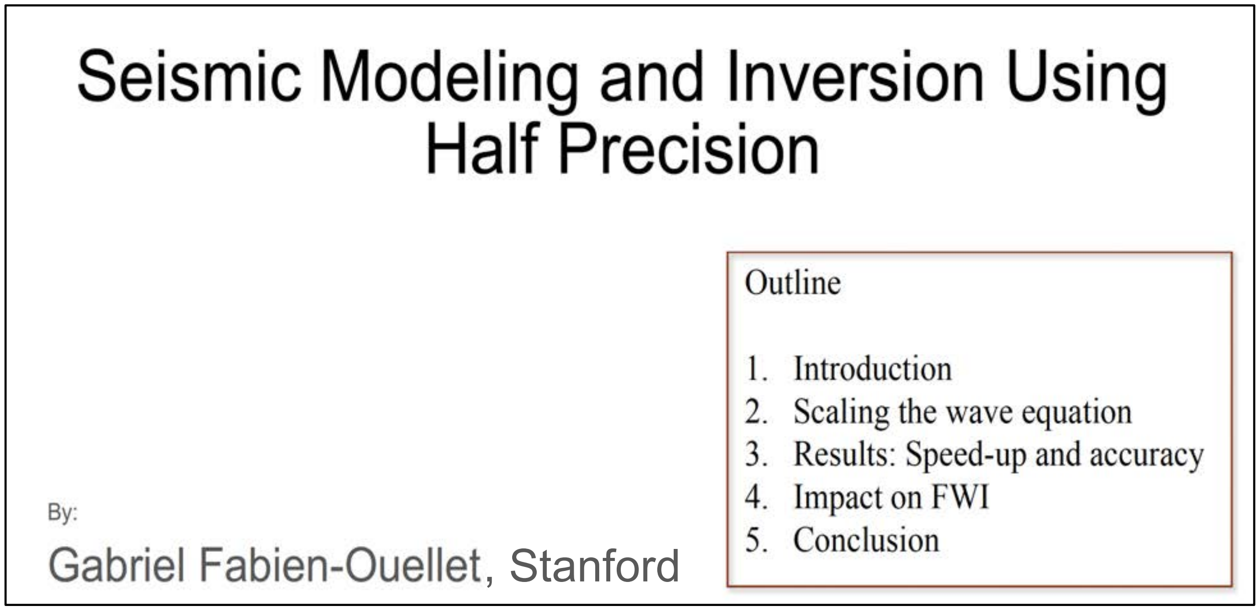
Inventor of electrochemistry (1802)
Discoverer of K, Na, Mg, Ca, Sr, Ba, B, Cl (1807-1810)

Davy’s 1807-1010 “sprint” through the periodic table


1 H
2 He
3 Li 4 Be
5 B 6 C 7 N 8 O 9 F 10 Ne
11 Na 12 Mg
13 Al 14 Si 15 P 16 S 17 Cl 18 Ar
19 K 20 Ca 21 Sc 22 Ti 23 V 24 Cr 25 Mn 26 Fe 27 Co 28 Ni 29 Cu 30 Zn 31 Ga 32 Ge 33 As 34 Se 35 Br 36 Kr
37 Rb 38 Sr 39 Y 40 Zr 41 Nb 42 Mo 43 Tc 44 Ru 45 Rh 46 Pd 47 Ag 48 Cd 49 In 50 Sn 51 Sb 52 Te 53 I 54 Xe
55 Cs 56 Ba 57 La 72 Hf 73 Ta 74 W 75 Re 76 Os 77 Ir 78 Pt 79 Au 80 Hg 81 Tl 82 Pb 83 Bi 84 Po 85 At 86 Rn
87 Fr 88 Ra 89 Ac 104 Unq 105 Unp 106 Unh 107 Uns 108 Uno 109 Une 110 Unn
58 Ce 59 Pr 60 Nd 61 Pm 62 Sm 63 Eu 64 Gd 65 Tb 66 Dy 67 Ho 68 Er 69 Tm 70 Yb 71 Lu
90 Th 91 Pa 92 U 93 Np 94 Pu 95 Am 96 Cm 97 Bk 98 Cf 99 Es 100 Fm 101 Md 102 No 103 Lr

+ Berkeley cyclotron elements

Bonus convergence benefit: Rethinking HPC in HDA datatypes



Seismic Modeling and Inversion Using Half Precision

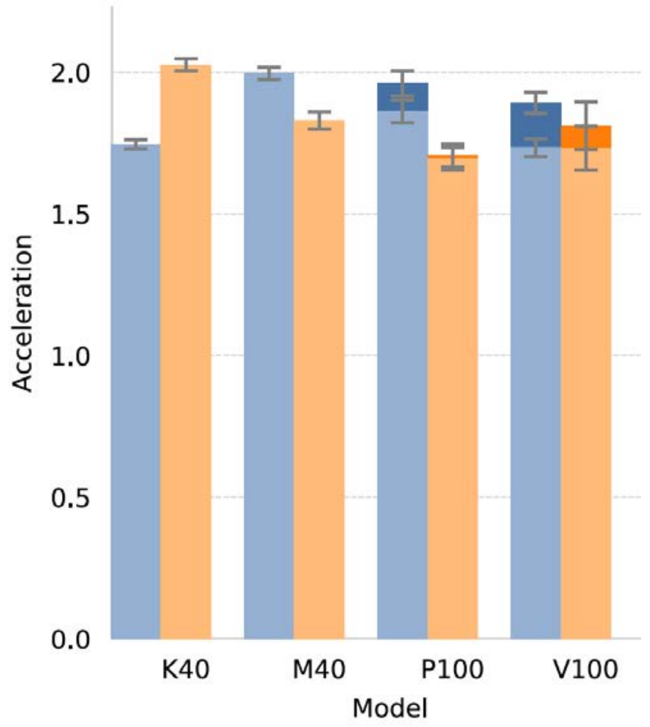
By:

Gabriel Fabien-Ouellet, Stanford

Outline

1. Introduction
2. Scaling the wave equation
3. Results: Speed-up and accuracy
4. Impact on FWI
5. Conclusion

FP16 over FP32



Fully acceptable accuracy in seismic imaging from single to half precision!

GTC 2018 Santa Clara

Bonus convergence benefit: Rethinking HPC in HDA datatypes

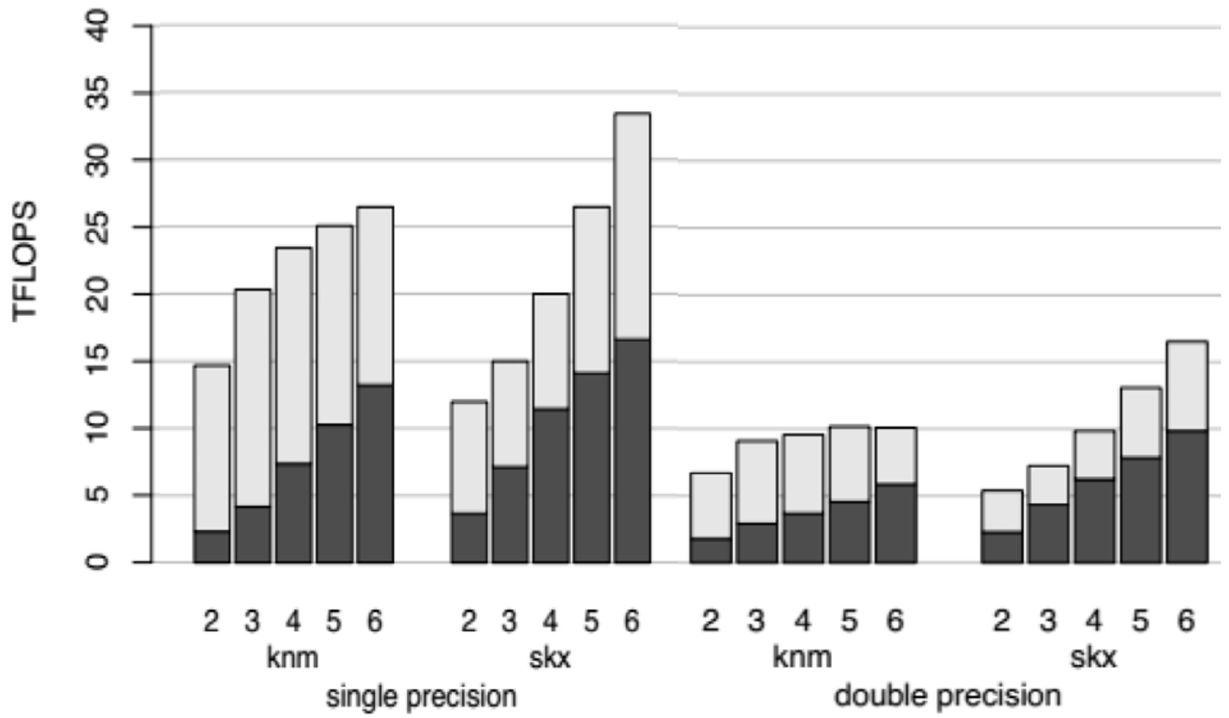
intel experience what's inside

DEEP LEARNING HARDWARE ACCELERATES FUSED DISCONTINUOUS GALERKIN SEISMIC SIMULATIONS

Alexander Heinecke

Parallel Computing Lab
Intel Labs
USA

April 23th 2018



Alexander Heinecke, Intel

Fully acceptable accuracy in seismic forward modeling from double to single precision!

IXPUG 2018 Saudi Arabia

Bonus convergence benefit: Data center economy

Reduce the time burden of I/O

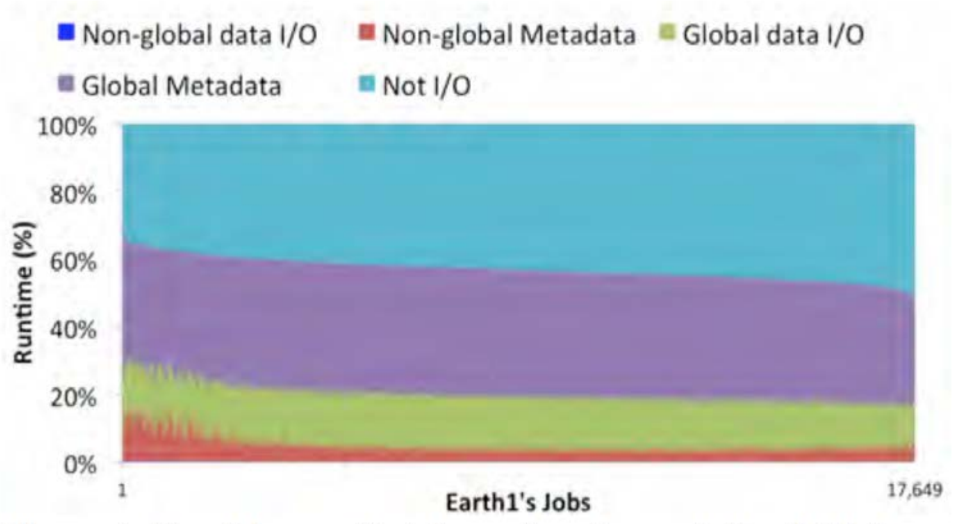


Figure 4: Breakdown of total run time for each Earth1 job.

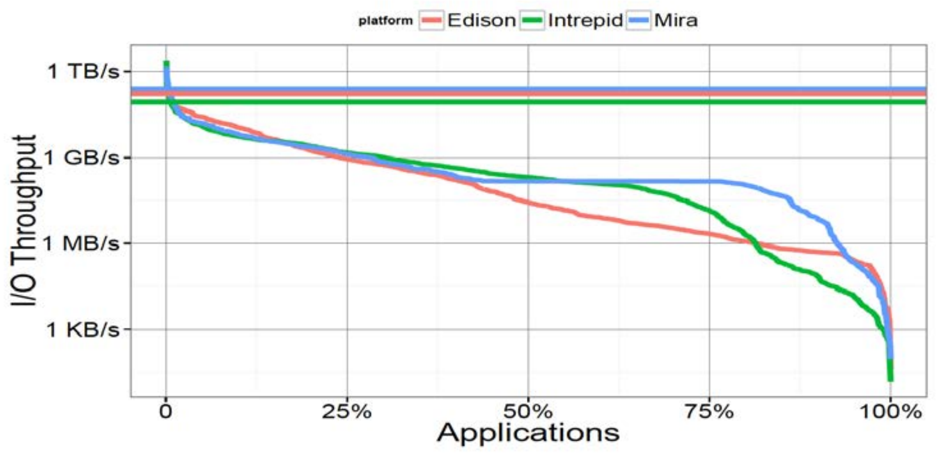
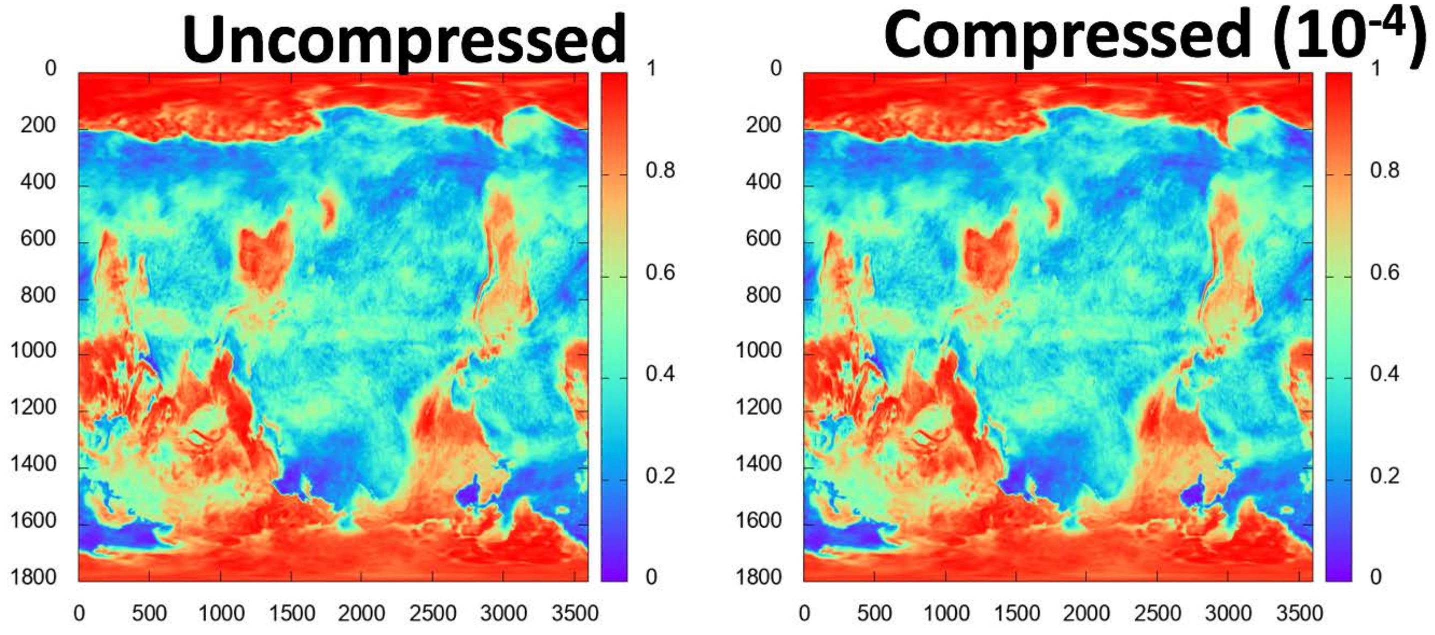


Figure 6: Maximum I/O throughput of each app across all its jobs on a platform, and platform peak I/O throughput.

c/o W. Gropp, UIUC

Bonus convergence benefit: Data center economy

Reduce the space burden of I/O



SZ Compression factor: 6.4

(1.4 with GZIP)

c/o F. Cappello, Argonne

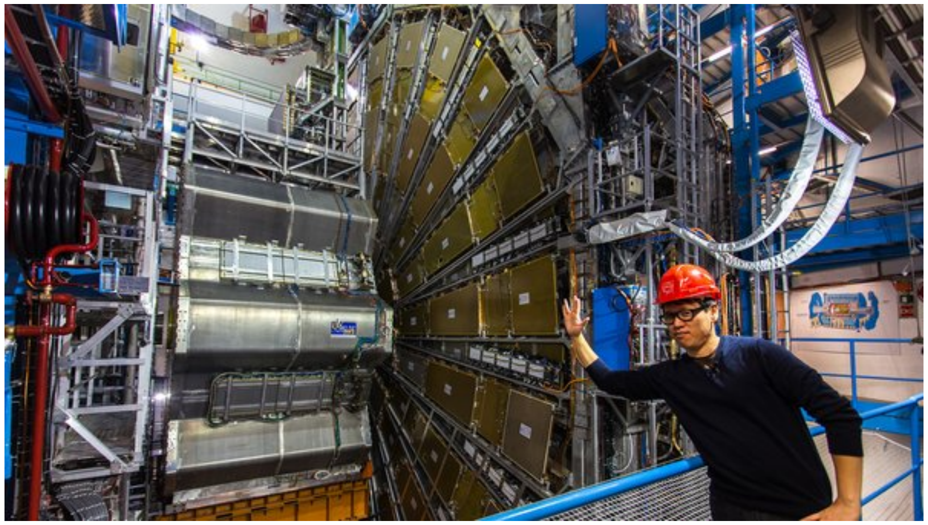
Summary observations: convergence

- “Convergence” began as an architectural imperative due to market size, but flourishes as a stimulus to both simulation science and data science
- However, the two distinct ecosystems require blending
- In standalone modes, architectures, operations, software, and data characteristics often strongly contrast
- Must be overcome since standalone mode may not be competitive

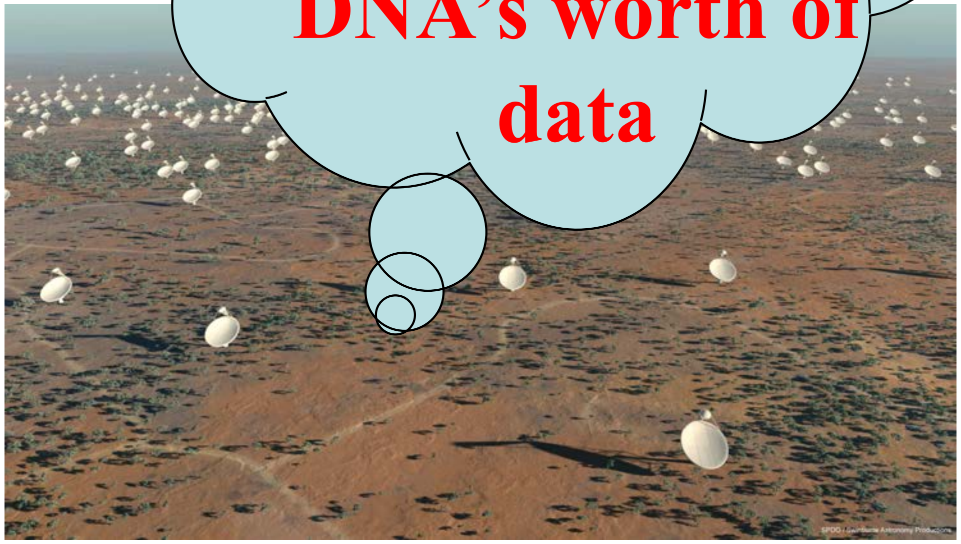
Giving convergence the “edge”

- Currently, data from “edge” devices is sent to the cloud to learn from
- Inference model is set back to th
- Need lightweight machine lea
  downsize the data

SKA will produce annually about 6 *global* human DNA’s worth of data



CERN (ATLAS pictured)
25 GB/s, 780 PB/yr

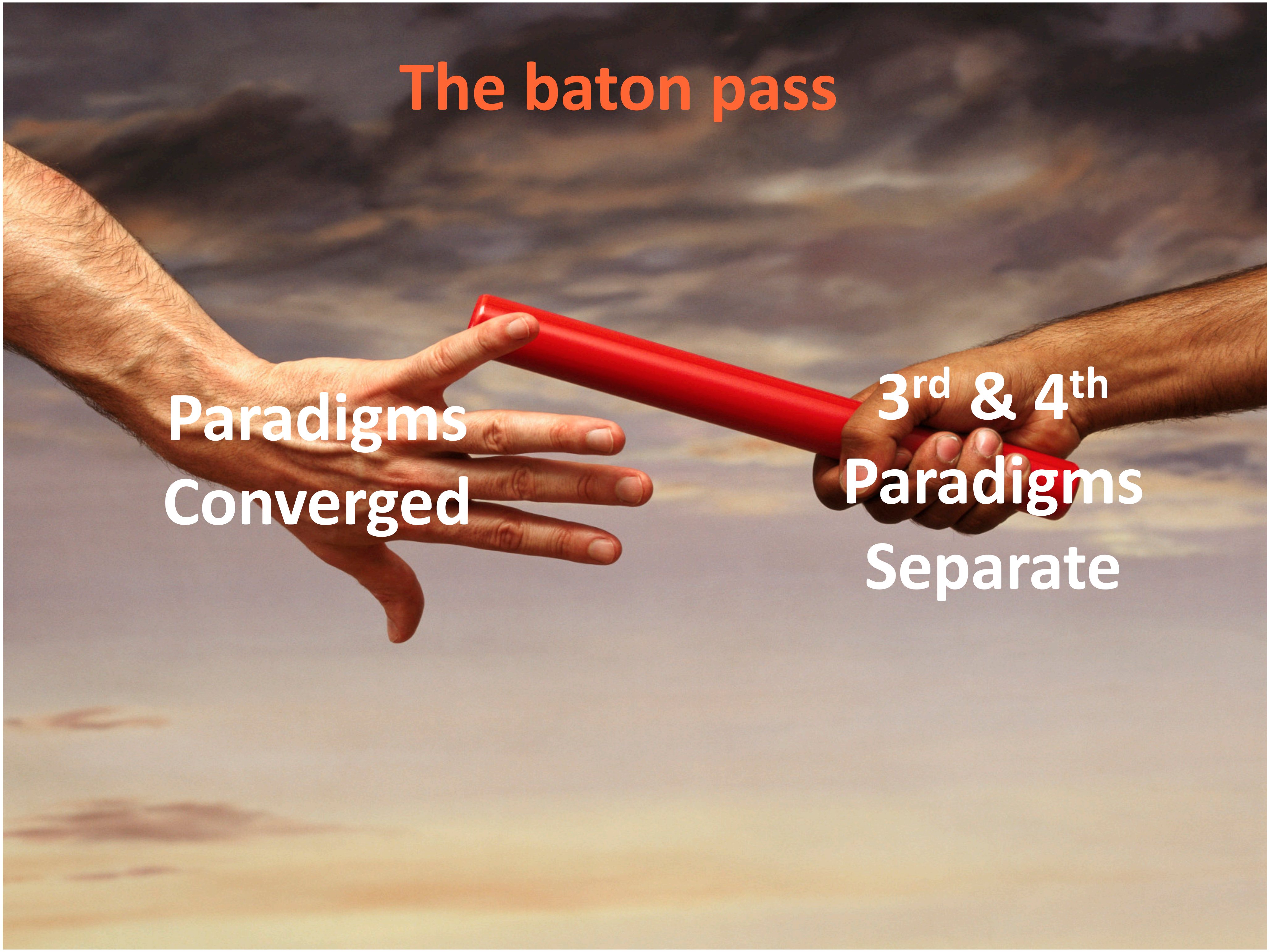


SKA (dishes pictured)
1 TB/s, 31 EB/yr, red to 3 EB/yr

Extending BDEC to the edge

- Mar 2018
  Chicago
- Nov 2018
  Indianapolis
- Feb 2019
  Kobe
- May 2019
  Poznan

The baton pass



2011 Roadmap report

The International Exascale Software Roadmap

J. Dongarra, et al., *International Journal of High Performance Computer Applications* **25**:3-60, 2011

Jack Dongarra, Pete Beckman, Terry Moore, Patrick Aerts, Giovanni Aloisio, Jean-Claude Andre, David Barkai, Jean-Yves Berthou, Taisuke Boku, Bertrand Braunschweig, Franck Cappello, Barbara Chapman, Xuebin Chi, Alok Choudhary, Sudip Dosanjh, Thom Dunning, Sandro Fiore, Al Geist, Bill Gropp, Robert Harrison, Mark Hereld, Michael Heroux, Adolfy Hoisie, Koh Hotta, Yutaka Ishikawa, Zhong Jin, Fred Johnson, Sanjay Kale, Richard Kenway, David Keyes, Bill Kramer, Jesus Labarta, Alain Lichnewsky, Thomas Lippert, Bob Lucas, Barney Maccabe, Satoshi Matsuoka, Paul Messina, Peter Michielse, Bernd Mohr, Matthias Mueller, Wolfgang Nagel, Hiroshi Nakashima, Michael E. Papka, Dan Reed, Mitsuhisa Sato, Ed Seidel, John Shalf, David Skinner, Marc Snir, Thomas Sterling, Rick Stevens, Fred Streitz, Bob Sugar, Shinji Sumimoto, William Tang, John Taylor, Rajeev Thakur, Anne Trefethen, Mateo Valero, Aad van der Steen, Jeffrey Vetter, Peg Williams, Robert Wisniewski, and Kathy Yelick

Exascale architectural drivers

- Clock rates cease to increase while arithmetic capability **continues to increase** dramatically with concurrency consistent with Moore's Law
- Memory storage capacity **fails to keep up** with arithmetic capability
- Transmission capability (memory bandwidth, network bandwidth) **fails to keep** up with arithmetic capability

➔ Billions of € £ $ ¥ of scientific applications worldwide hang in the balance until algorithms better span the growing **architecture-applications gap**

Two decades of evolution

1997 2017

ASCI Red at Sandia

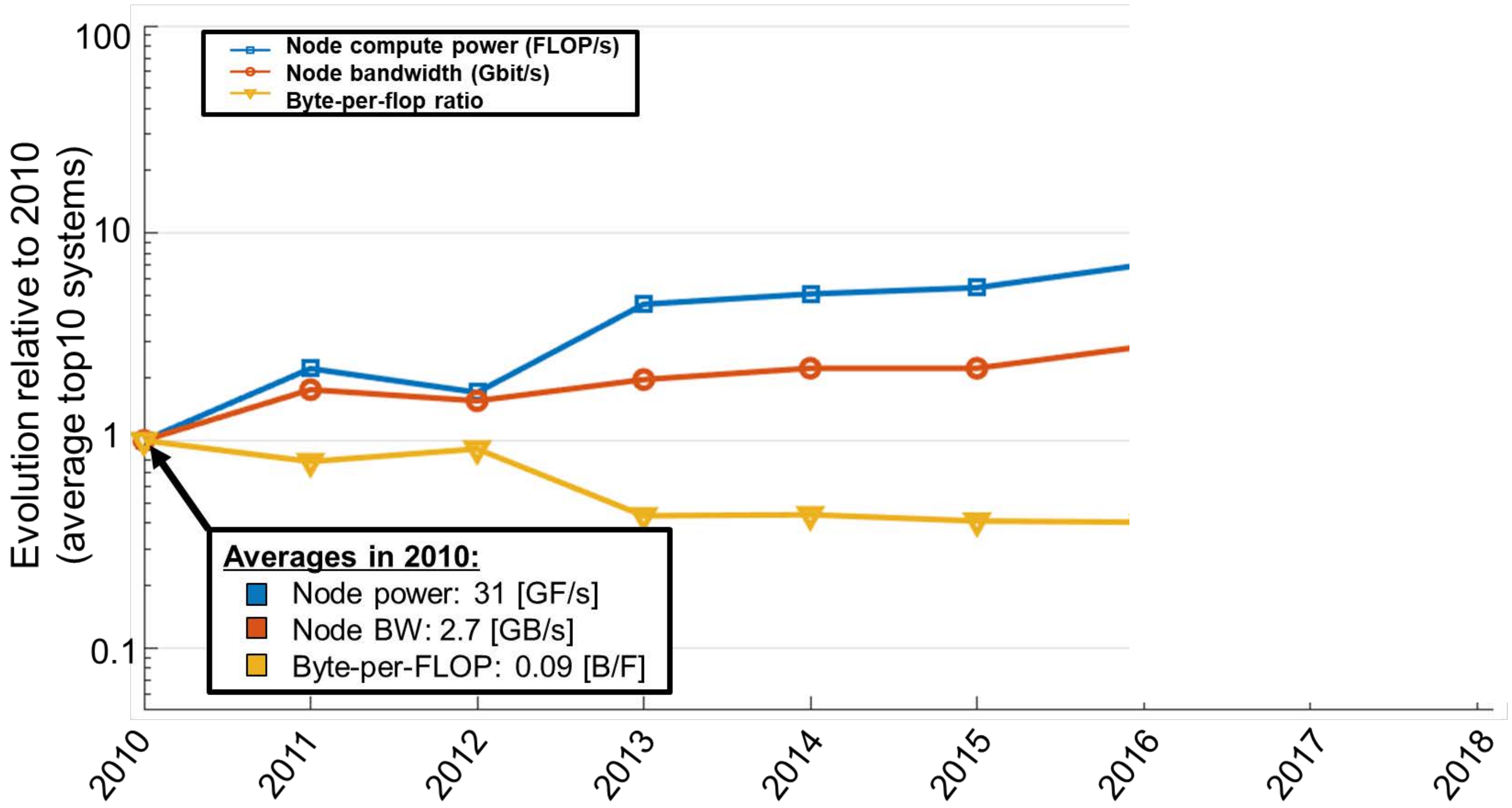
1.3 TF/s, 850 KW

Cavium ThunderX2

~ 1.1 TF/s, ~ 0.2 KW

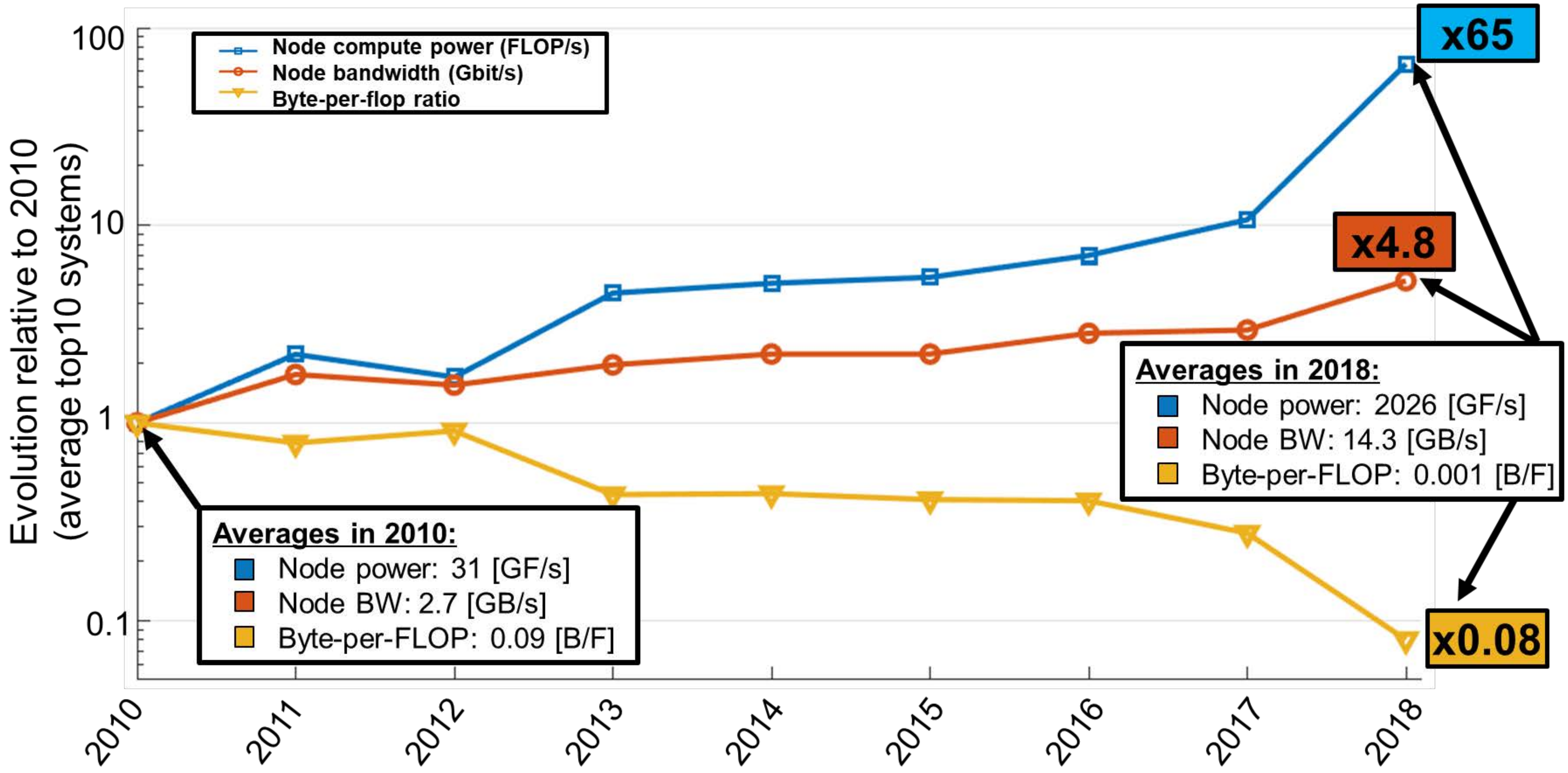
3.5 orders of magnitude

Top 10 architecture trends, 2010-2018



c/o Keren Bergman (Columbia, ISC'18)

Top 10 architecture trends, 2010-2018



Sunway TaihuLight (Nov 2017) B/F = 0.004;

Summit HPC (June 2018) B/F = 0.0005

8x deterioration in 2018

c/o Keren Bergman (Columbia, ISC’18)

It’s not just bandwidth; it’s energy

• Access SRAM (registers, cache)	~ 10 fJ/bit
• Access DRAM on chip	~ 1 pJ/bit
• Access HBM (few mm)	~ 10 pJ/bit
• Access DDR3 (few cm)	~ 100 pJ/bit

~ 10^4 advantage in energy for staying in cache!

similar ratios for *latency* as for *bandwidth* and *energy*

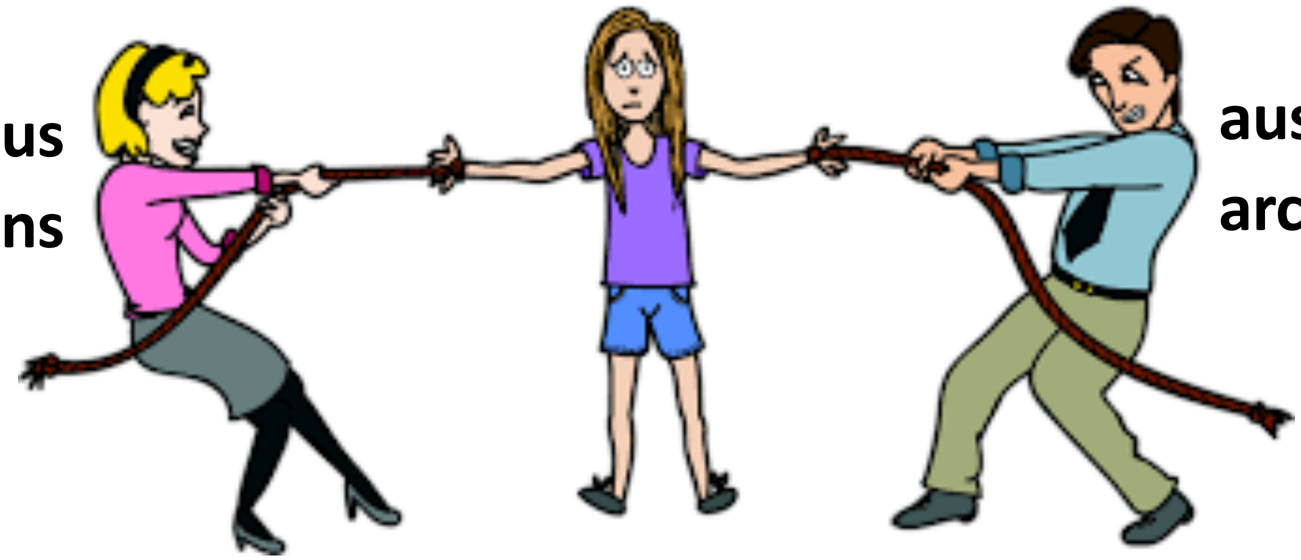
Algorithmic philosophy

Algorithms must span a widening gulf …

adaptive algorithms

ambitious applications

austere architectures



A full employment program for algorithm developers ☺

➔ **Billions of**

$ € £ ¥

of scientific software worldwide hangs in the balance until our algorithmic infrastructure evolves to span the architecture-applications gap

Required software

Model-related

- Geometric modelers
- Meshers
- Discretizers
- Partitioners
- Solvers / integrators
- Adaptivity systems
- Random no. generators
- Subgridscale physics
- Uncertainty quantification
- Dynamic load balancing
- Graphs and combinatorial algs.
- Compression

Development-related

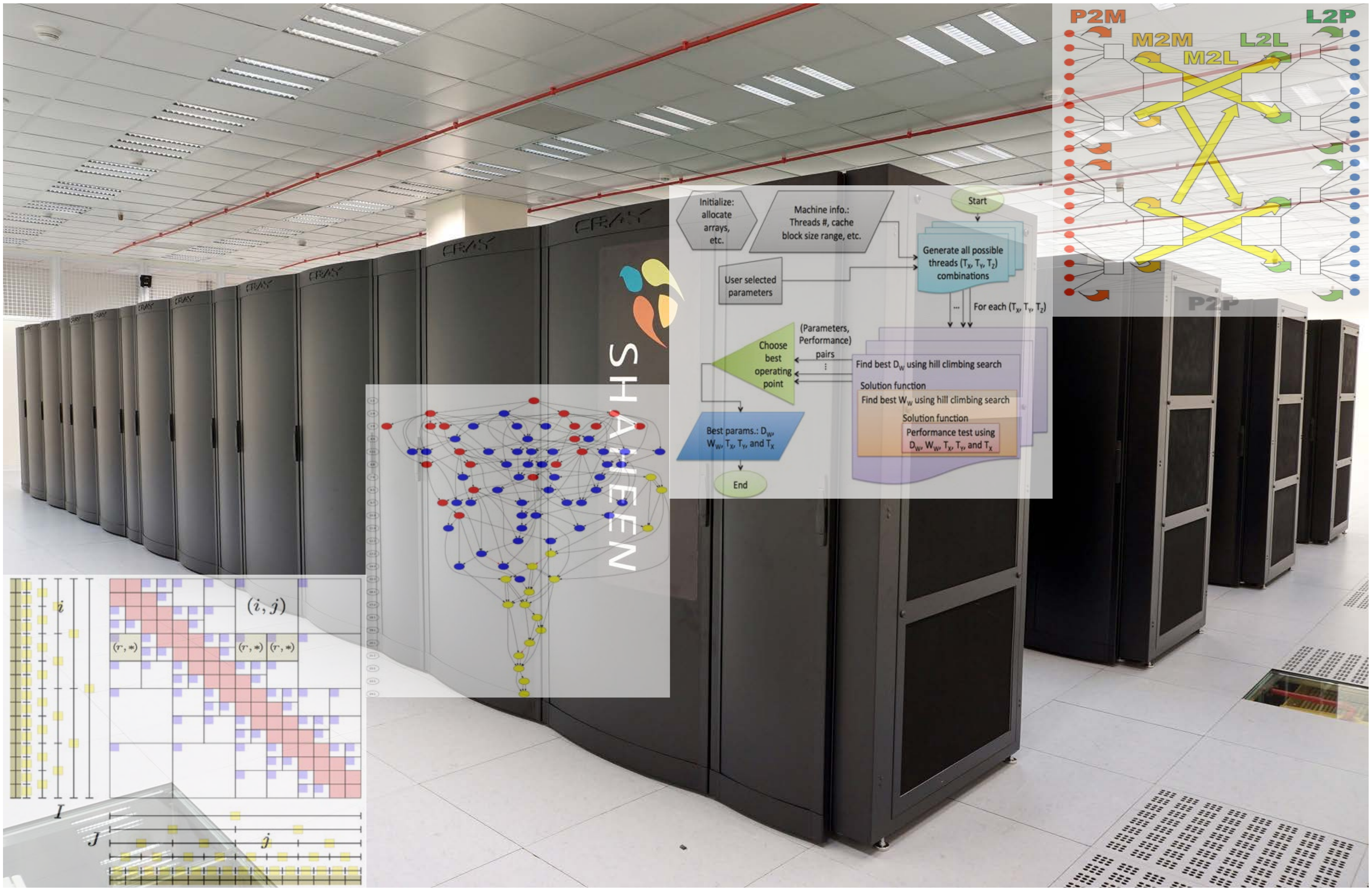
- Configuration systems
- Source-to-source translators
- Compilers
- Simulators
- Messaging systems
- Debuggers
- Profilers

High-end computers come with little of this. Most is contributed by the user community.

Production-related

- Visualization systems
- Dynamic resource management
- Dynamic performance optimization
- Authenticators
- I/O systems
- Workflow controllers
- Frameworks
- Data miners
- Fault monitoring, reporting, and recovery

Embracing the opportunities of exascale

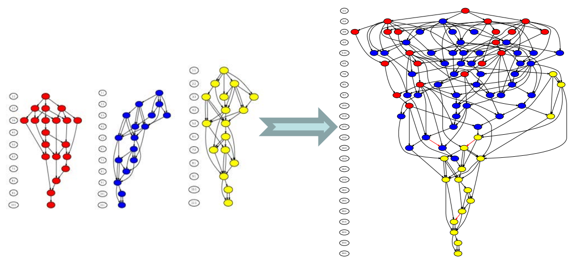

P2M
M2M
M2L
L2L
L2P
P2P
Initialize: allocate arrays, etc.
Machine info.: Threads #, cache block size range, etc.
Start
Generate all possible threads (T_X, T_Y, T_Z) combinations
User selected parameters
For each (T_X, T_Y, T_Z)
(Parameters, Performance) pairs
Choose best operating point
Find best D_W using hill climbing search
Solution function
Find best W_W using hill climbing search
Solution function
Performance test using D_W, W_W, T_X, T_Y, and T_X
Best params.: D_W, W_W, T_X, T_Y, and T_X
End
SHAHEEN
CRAY
(i, j)
i
$(r, *)$
$(r, *)$
$(r, *)$
I
J
j

Architectural imperatives for algorithms

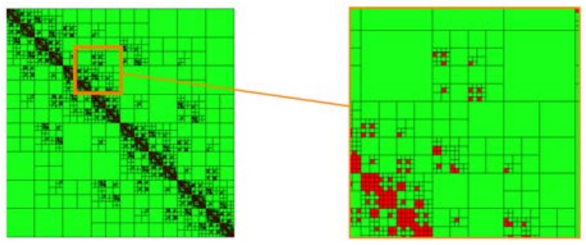
- **Reduce synchrony**
  - in frequency or span or both
  - cannot afford to synchronize a billion imbalanced cores
- **Reside “high” on the memory hierarchy**
  - as close as possible to the processing elements
  - latency to DRAM may be a thousand cycles
  - moving data is orders of magnitude more costly in energy than computing
- **Increase SIMT/SIMD-style shared-memory concurrency**
  - one instruction can trigger 8 (AVX 512) to 64 (tensor core) operations

Exascale algorithmic strategies

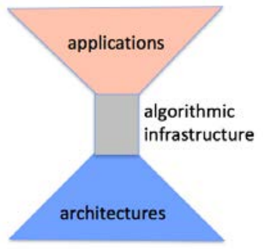
- Employ dynamic runtime systems based on directed acyclic task graphs (DAGs)
  - e.g., ADLB, Argo, Charm++, HPX, Legion, OmpSs, Quark, STAPL, StarPU, OpenMP



- Exploit hierarchical low-rank data sparsity
  - meet "curse of dimensionality" with "blessing of low rank"



- Code to the architecture, but present an abstract API
  - "hourglass model" of IP/TCP for processors



1) Taskification based on DAGs

- Advantages
  - remove artifactual synchronizations in the form of subroutine boundaries
  - remove artifactual orderings in the form of pre-scheduled loops
  - expose more concurrency
- Disadvantages
  - pay overhead of managing task graph
  - potentially lose some memory locality

2) Hierarchically low-rank operators

- Advantages
  - shrink memory footprints to live higher on the memory hierarchy
    - higher means quick access (↑ arithmetic intensity)
  - reduce operation counts
  - tune work to accuracy requirements
    - e.g., preconditioner versus solver
- Disadvantages
  - pay cost of compression
  - not all operators compress well

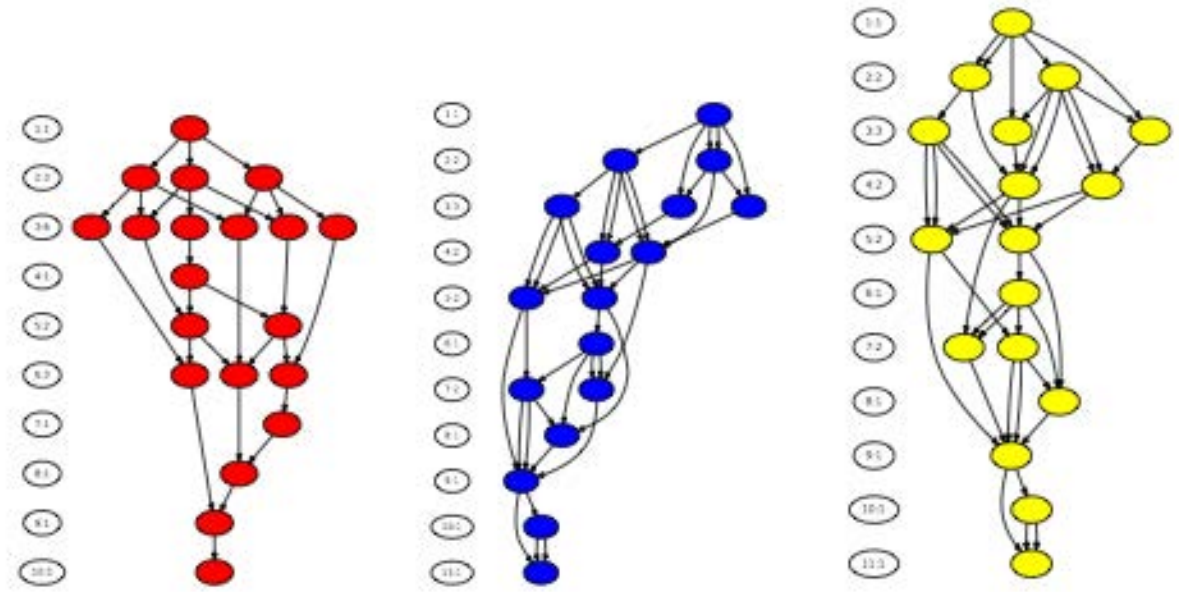
3) Code to the architecture

- Advantages
  - tiling and recursive subdivision create large numbers of small problems that can be marshaled for batched operations on GPUs and MICs
    - amortize call overheads
    - polyalgorithmic approach based on block size
  - non-temporal stores, coalesced memory accesses, double-buffering, etc. reduce sensitivity to memory
- Disadvantages
  - code is more complex
  - code is architecture-specific at the bottom

1) Reduce over-ordering and synchronization through DAGs, ex.: generalized eigensolver

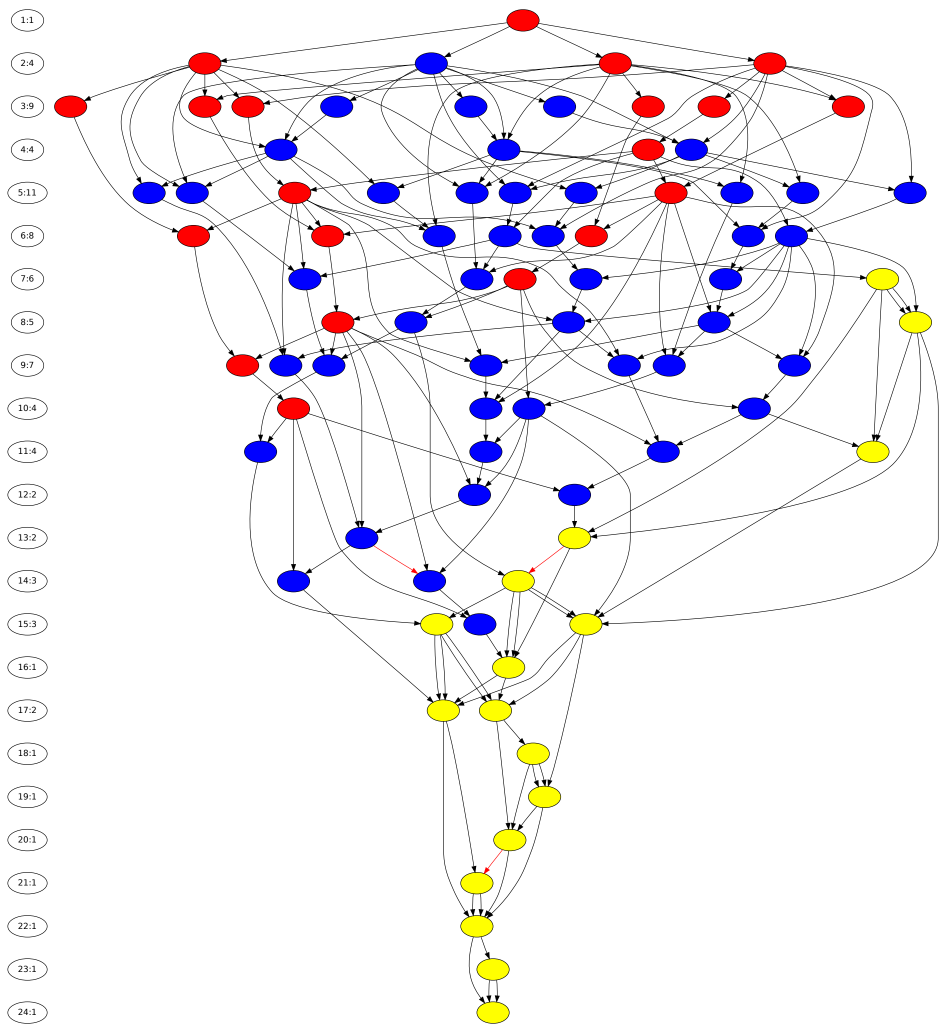
$$Ax = \lambda Bx$$

	Operation	Explanation	LAPACK routine name
1	$B = L \times L^T$	Cholesky factorization	POTRF
2	$C = L^{-1} \times A \times L^{-T}$	application of triangular factors	SYGST or HEGST
3	$T = Q^T \times C \times Q$	tridiagonal reduction	SYEVD or HEEVD
4	$Tx = \lambda x$	QR iteration	STERF



Loop nests and subroutine calls, with their over-orderings, can be replaced with DAGs

- Diagram shows a dataflow ordering of the steps of a 4 × 4 symmetric generalized eigensolver
- Nodes are tasks, color-coded by type, and edges are data dependencies
- Time is vertically downward
- Wide is good; short is good



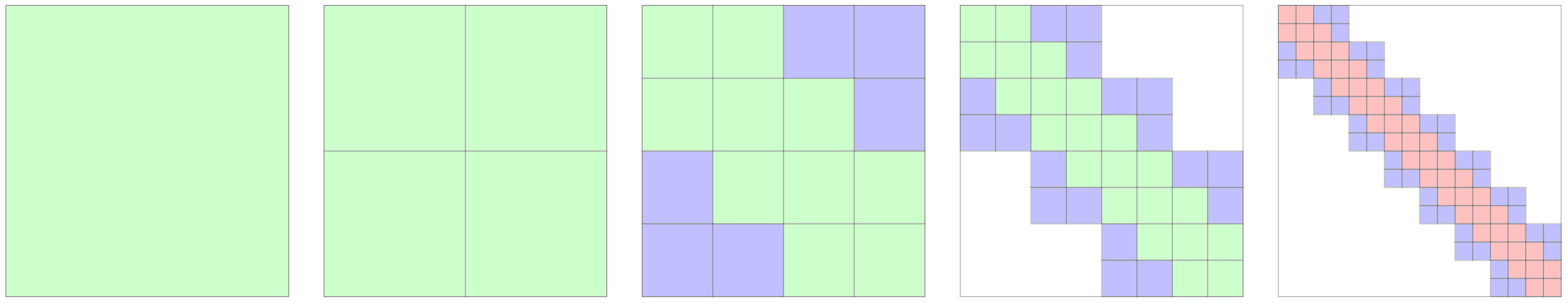
2) Reduce memory footprint and operation complexity with low rank

- Replace dense blocks with hierarchical representations when they arise during matrix operations
  - use high accuracy (high rank, but typically less than full) to build “exact” solvers
  - use low accuracy (low rank) to build preconditioners
- Tune block structure and rank parameters to variety of hardware configurations

Key tool: hierarchical matrices

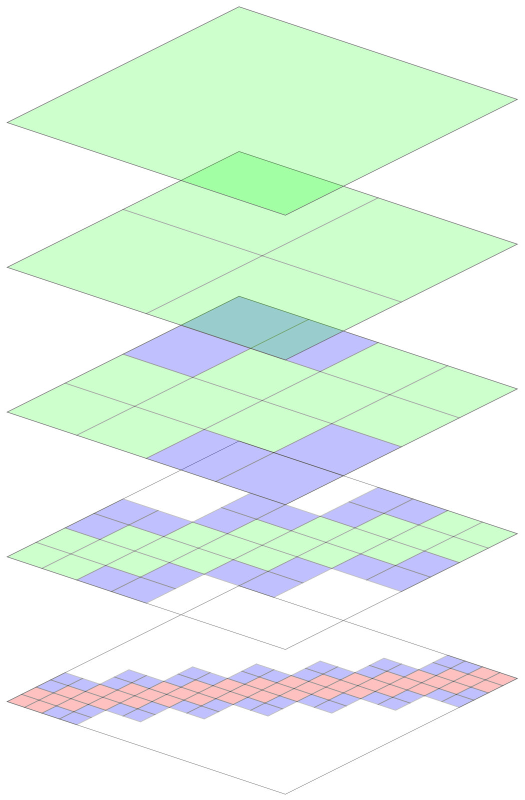
- [Hackbusch, 1999] **: off-diagonal blocks of typical differential and integral operators have low effective rank**
- **By exploiting low rank, *k* , memory requirements and operation counts approach optimal in matrix dimension *n*:**
  - **polynomial in *k***
  - **lin-log in *n***
  - **constants carry the day**
- **Such hierarchical representations navigate a compromise**
  - **fewer blocks of larger rank (“weak admissibility”) or**
  - **more blocks of smaller rank (“strong admissibility”)**

Recursive construction of an $\mathcal{H}$-matrix



Specify two parameters:

- Block size acceptably small to handle densely
- Rank acceptably small to represent block

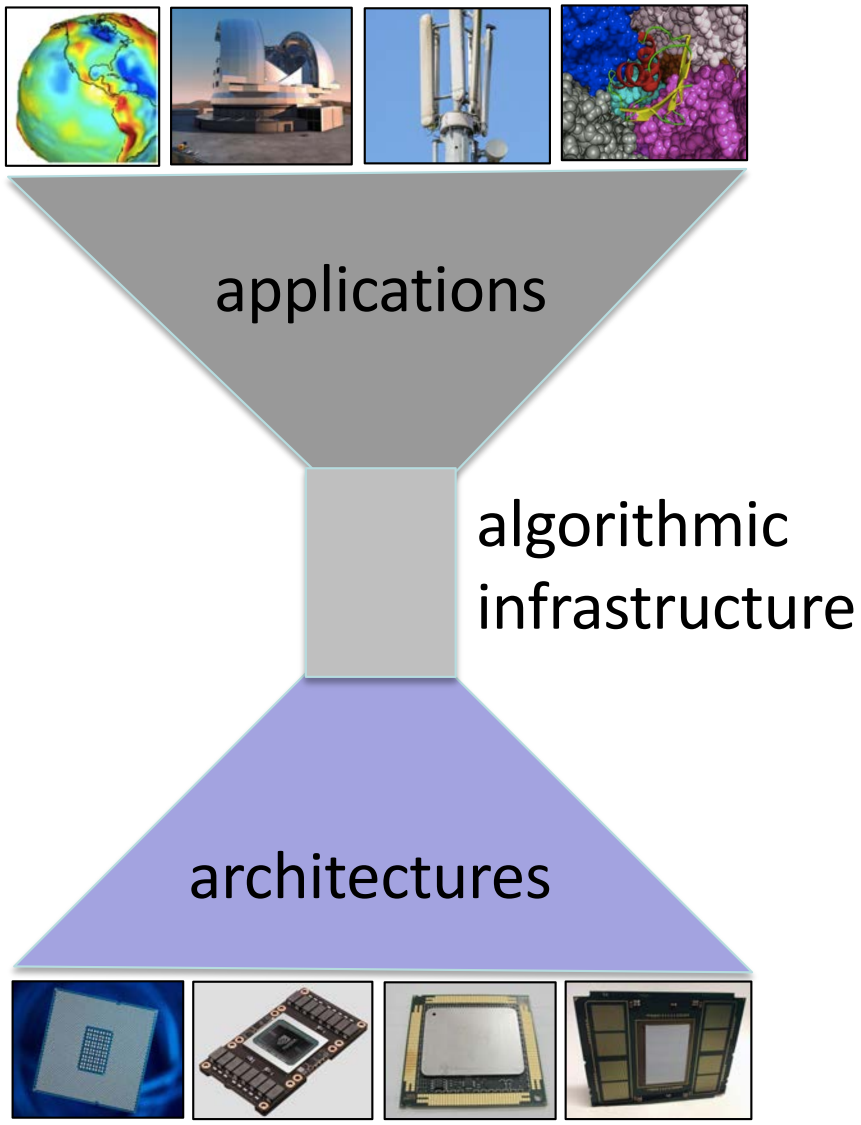


Until each block is acceptably small:

- Is rank acceptably small?
- If not, subdivide block

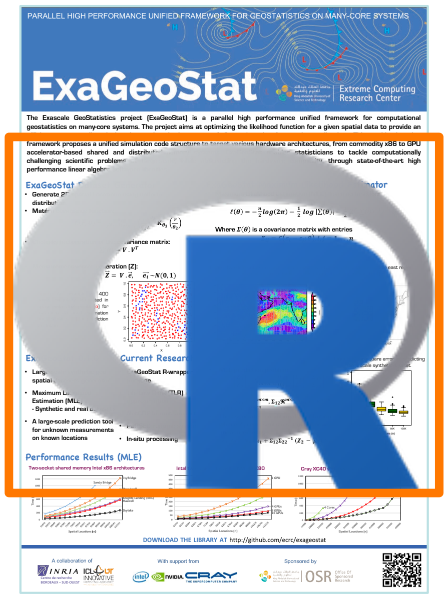
Take union of leaf blocks

3) "Hourglass" model for algorithms (borrowed from internet protocols)

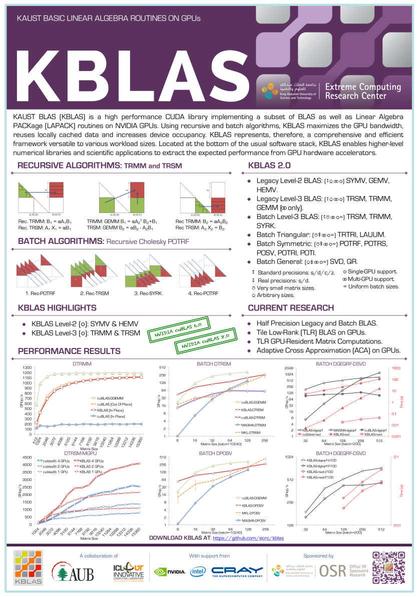


Software implementing these strategies

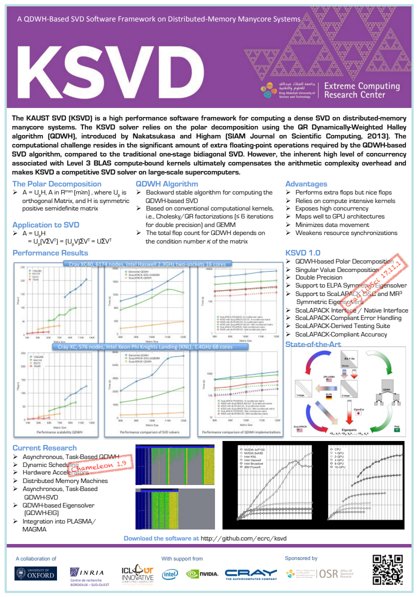
A HIGH PEFORMANCE MULTI-OBJECT ADAPTIVE OPTICS FRAMEWORK FOR GROUND-BASED ASTRONOMY
MOAO
Extreme Computing Research Center


PARALLEL HIGH PERFORMANCE UNIFIED FRAMEWORK FOR GEOSTATISTICS ON MANY-CORE SYSTEMS
ExaGeoStat
Extreme Computing Research Center

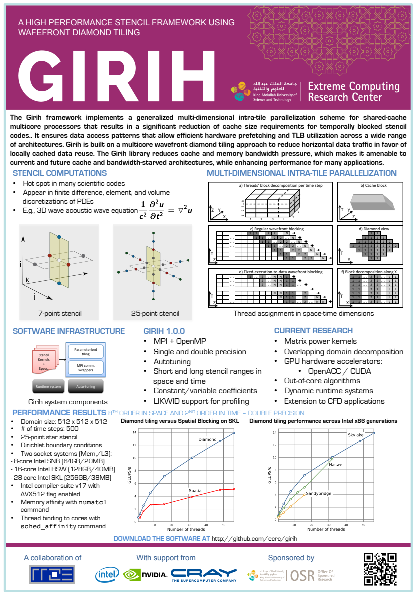
in NVIDIA cuBLAS

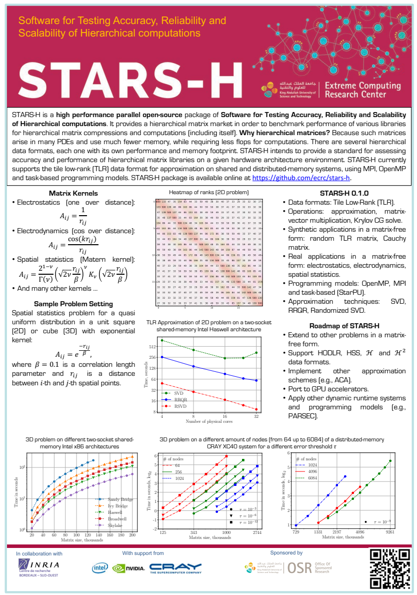

KAUST BASIC LINEAR ALGEBRA ROUTINES ON GPUs
KBLAS
Extreme Computing Research Center

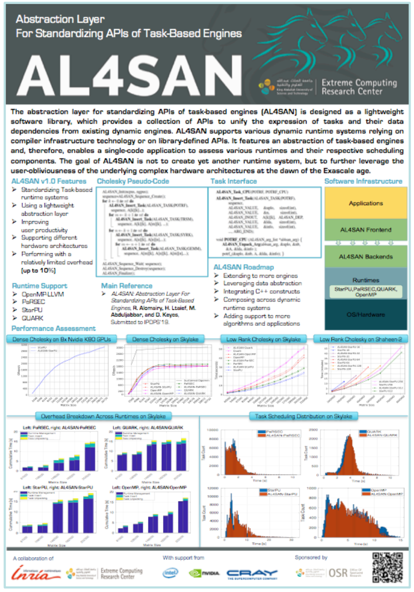
in Cray LibSci

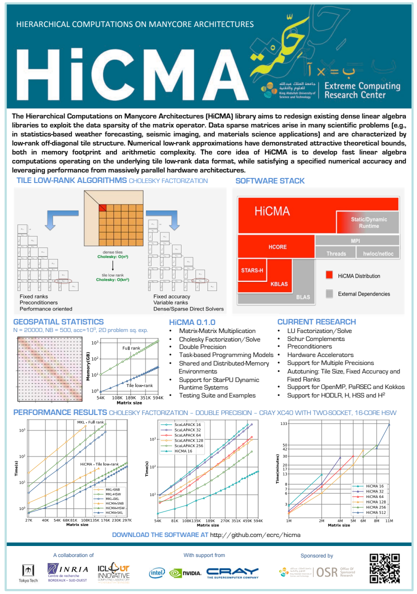

A QDWH-Based SVD Software Framework on Distributed-Memory Manycore Systems
KSVD
Extreme Computing Research Center

Intel s/w for Aramco


A HIGH PERFORMANCE STENCIL FRAMEWORK USING WAFEFRONT DIAMOND TILING
GIRIH
Extreme Computing Research Center

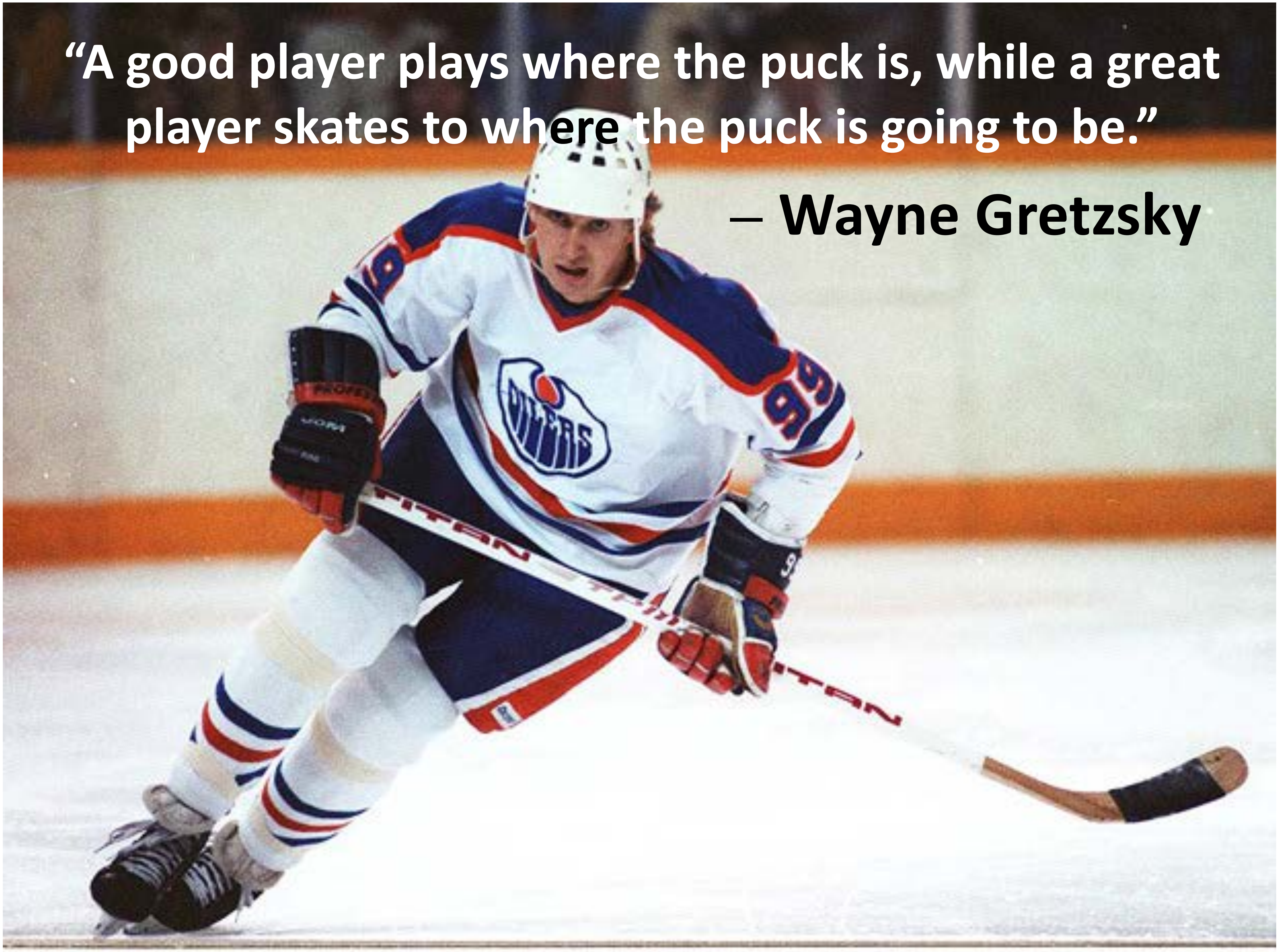

Software for Testing Accuracy, Reliability and Scalability of Hierarchical computations
STARS-H
Extreme Computing Research Center


Abstraction Layer For Standardizing APIs of Task-Based Engines
AL4SAN
Extreme Computing Research Center


HIERARCHICAL COMPUTATIONS ON MANYCORE ARCHITECTURES
HiCMA
Extreme Computing Research Center

“A good player plays where the puck is, while a great player skates to where the puck is going to be.”

– Wayne Gretzsky



The second baton pass

Architectural “trickles”

- HPC hardware architecture has “trickle down” benefits
  - “Petascale in the machine room means terascale on the node.” [Petaflops Working Group, 1990s]
  - Extrapolating: exascale on the machine room floor means petascale under the desk.
- HDA software architecture has “trickle back” benefits
  - “Google is living a few years in the future and sends the rest of us messages.” [Doug Cutting, Hadoop founder]

Motivations for convergence

- Scientific and engineering advances
  - tune physical parameters in simulations for predictive performance
  - tune algorithmic parameters of simulations for execution performance
  - filter out nonphysical candidates in learning
  - provide data for learning
- Economy of data center operations
  - obviate I/O
  - obviate computation!
- Development of a competitive workforce
  - leaders in adopting disruptive tools have advantages in capability and in recruiting

References to the community reports

- **exascale.org/bdec**
  - http://www.exascale.org/bdec/sites/www.exascale.org.bdec/files/whitepapers/bdec2017pathways.pdf
  - "Big Data and Extreme-scale Computing: Pathways to Convergence," M. Asch, et al., *Int. J. High Perf. Comput. Applics.* **32**:435-479, 2018
- **exascale.org/iesp**
  - http://www.exascale.org/mediawiki/images/2/20/IESP-roadmap.pdf
  - "The International Exascale Software Roadmap," J. Dongarra, et al., *Int. J. High Perf. Comput. Applics.* **25**:3-60, 2011

Concluding prediction

- No need to force a “shotgun” marriage of “convergence” between 3rd and 4th paradigms
  - a love-based marriage is inevitable in the near future
- Driver will be opportunity for both 3rd and 4th paradigm communities to address their own traditional concerns in a superior way in mission-critical needs in scientific discovery and engineering design

جامعة الملك عبدالله للعلوم والتقنية
King Abdullah University of Science and Technology

Thank you!

david.keyes@kaust.edu.sa