University of Arkansas Department of Mathematical Sciences

46th Spring Lecture Series

David Keyes

Extreme Computing Research Center

King Abdullah University of Science and Technology

5-9 April 2021

Lecture 5

The Convergence of Big Data and Extreme Computing

Greetings from KAUST's President

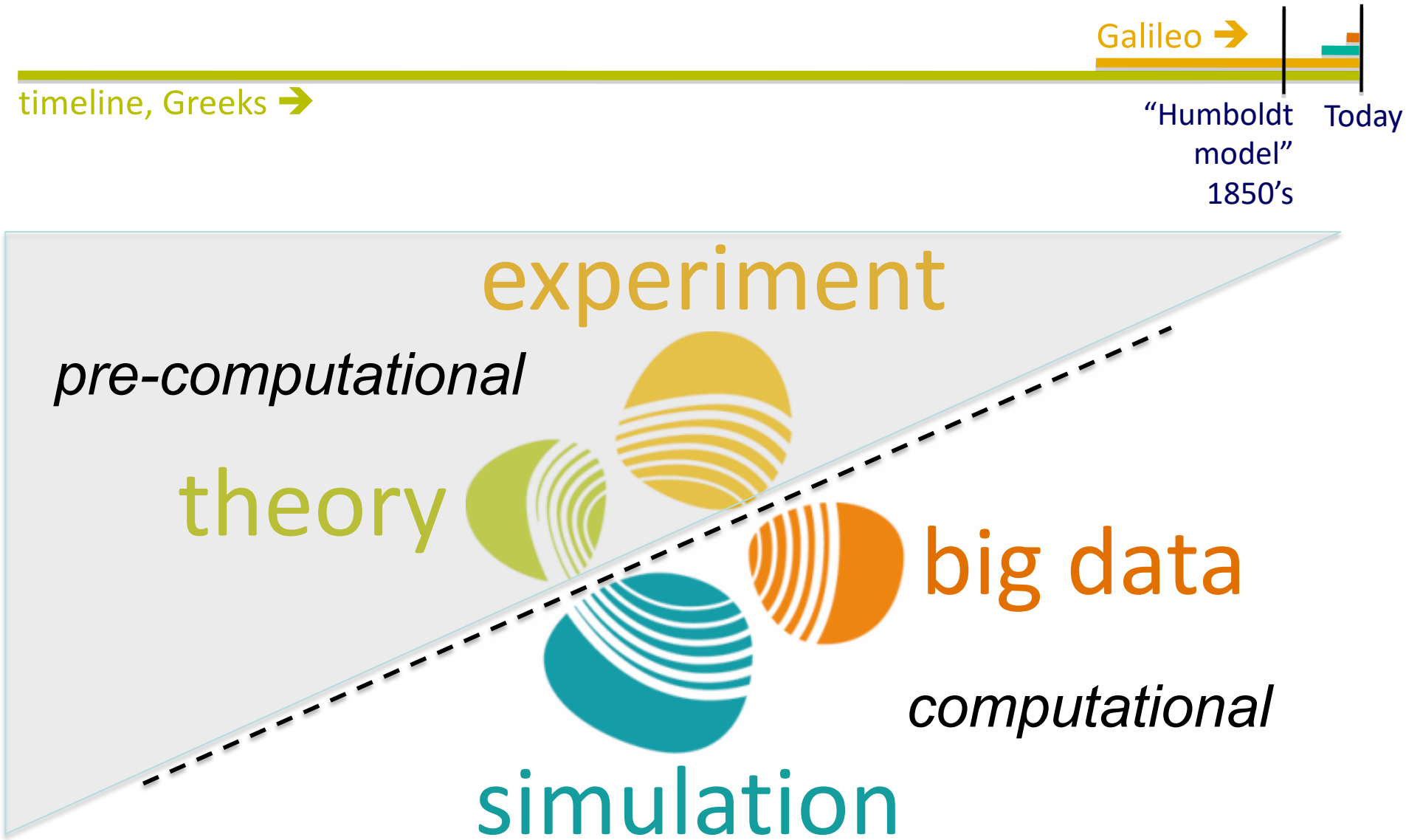
Tony Chan

- **Member, NAE**
- **Fellow of SIAM, IEEE, AAAS**
- **ISI highly cited, imaging sciences, numerical analysis**

Formerly:

- **President, HKUST**
- **Director, Div Math & Phys Sci, NSF**
- **Dean, Phys Sci, UCLA**
- **Chair, Math, UCLA**
- **Co-founder, IPAM**

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

***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)

Three 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 reducing raw data transmission

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., treaties for nuclear weapons testing and climate)
  - recognized for decision support (e.g., oil drilling, therapy planning)
- **HDA: high-end data analytics**
  - grew up around open source tools (e.g., Google MapReduce, Apache 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

Trillion dollar market? Yes.

Rank	Name	Market Cap	Price	Today	Price (30 days)	Country
1	Apple AAPL	$2.187 T	$130.28	1.86%		USA
^1 2	Microsoft MSFT	$1.910 T	$253.22	1.33%		USA
v1 3	Saudi Aramco 2222.SR	$1.893 T	$9.47	0.14%		S. Arabia
4	Amazon AMZN	$1.670 T	$3,317	1.15%		USA
5	Alphabet (Google) GOOG	$1.524 T	$2,266	0.70%		USA
6	Facebook FB	$893.36 B	$313.72	0.20%		USA
7	Tencent TCEHY	$798.67 B	$80.57	2.94%		China
8	Tesla TSLA	$656.75 B	$684.22	1.97%		USA
9	Alibaba BABA	$628.90 B	$228.85	1.52%		China

- **The market capitalization of the 7 highlighted IT companies from sums to $9.6T today**
- **Annual revenues of these same companies for 2021 is projected to be approximately $2T**

https://companiesmarketcap.com/ [downloaded 8 April 2021]

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, *our concerns are the same*:**
  - **energy efficiency**
  - **limited memory per core**
  - **limited memory bandwidth per core**
  - **cost of moving data "horizontally" and "vertically"**

Pressure on HDA

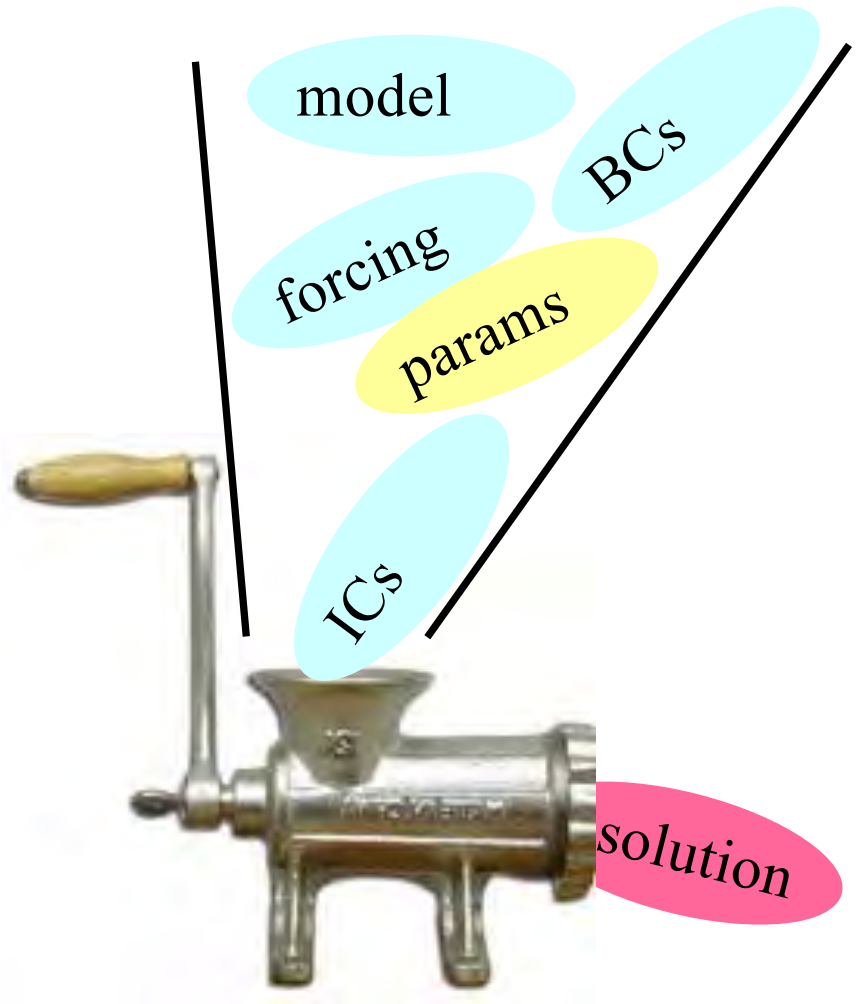
- Since the beginning of the big data age, data has been moved over "stateless" networks
  - routing is based on address bits in the data packets
  - no system-wide coordination of data sets or buffering
- Workarounds coped with volume but are now creaking
  - ftp mirror sites, web-caching (e.g., Akamai 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 (2017) report findings

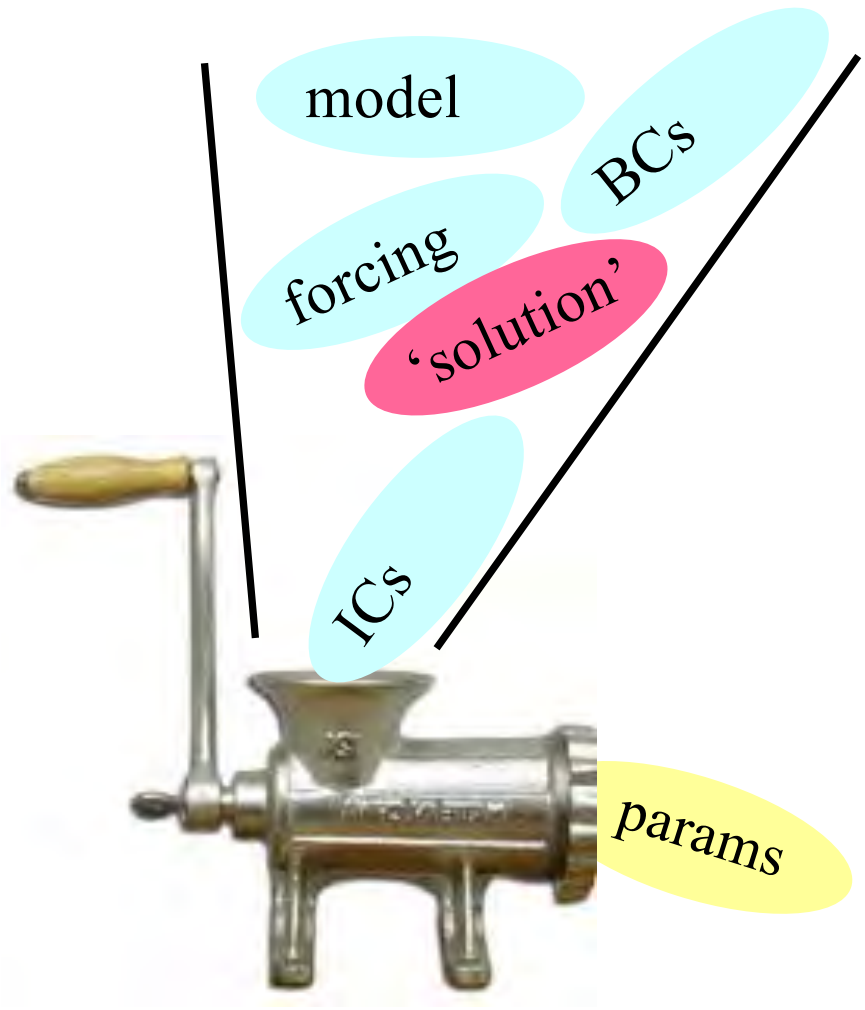
- **Many motivations to bring together large-scale simulation and big data analytics ("convergence")**
- **Should be combined *in situ***
  - **pipelining between simulation and analytics through disk files with sequential applications leaves too many benefits "on the table"**
- **Many hurdles to convergence 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!**

A traditional combination of 3rd/4th paradigms: from forward to inverse problems

forward problem



inverse problem



+ regularization

A traditional combination of 3rd/4th paradigms: data assimilation

Theory

Fully Nonlinear Filters

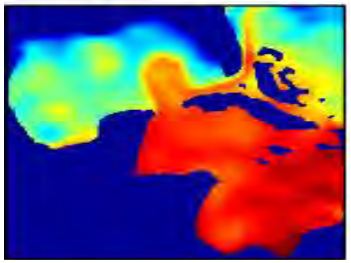
Dual Filters Coupled Models

Robust Ensemble Filters

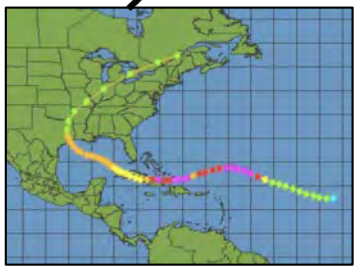
Hybrid Adjoint-Ensemble Filters

Bayesian Filtering
Data Assimilation

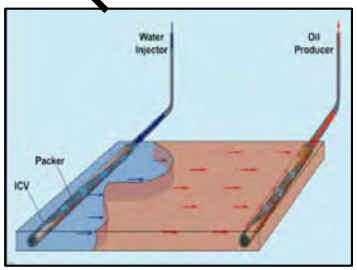
Applications



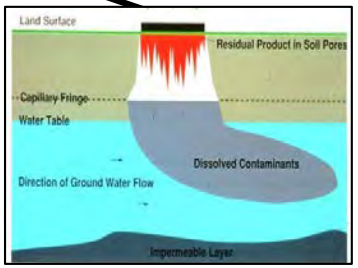
Ocean Circulation



Storm Surge Prediction



Reservoir Exploitation



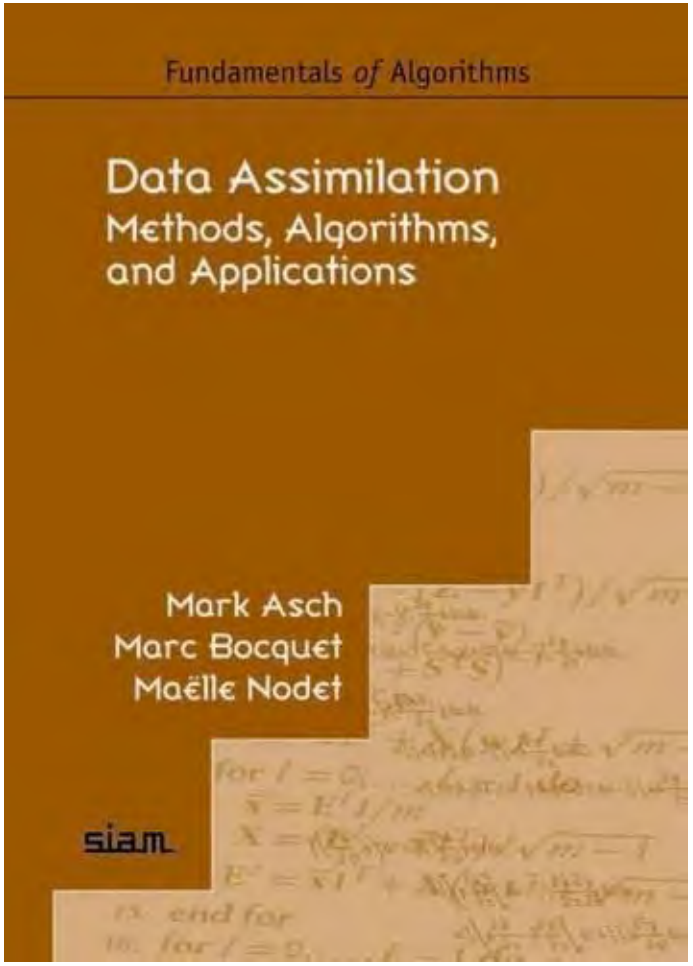
Contaminant Transport

c/o I. Hoteit, KAUST

My definition of data assimilation

"When two ugly parents have a beautiful child"

A beautiful book



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

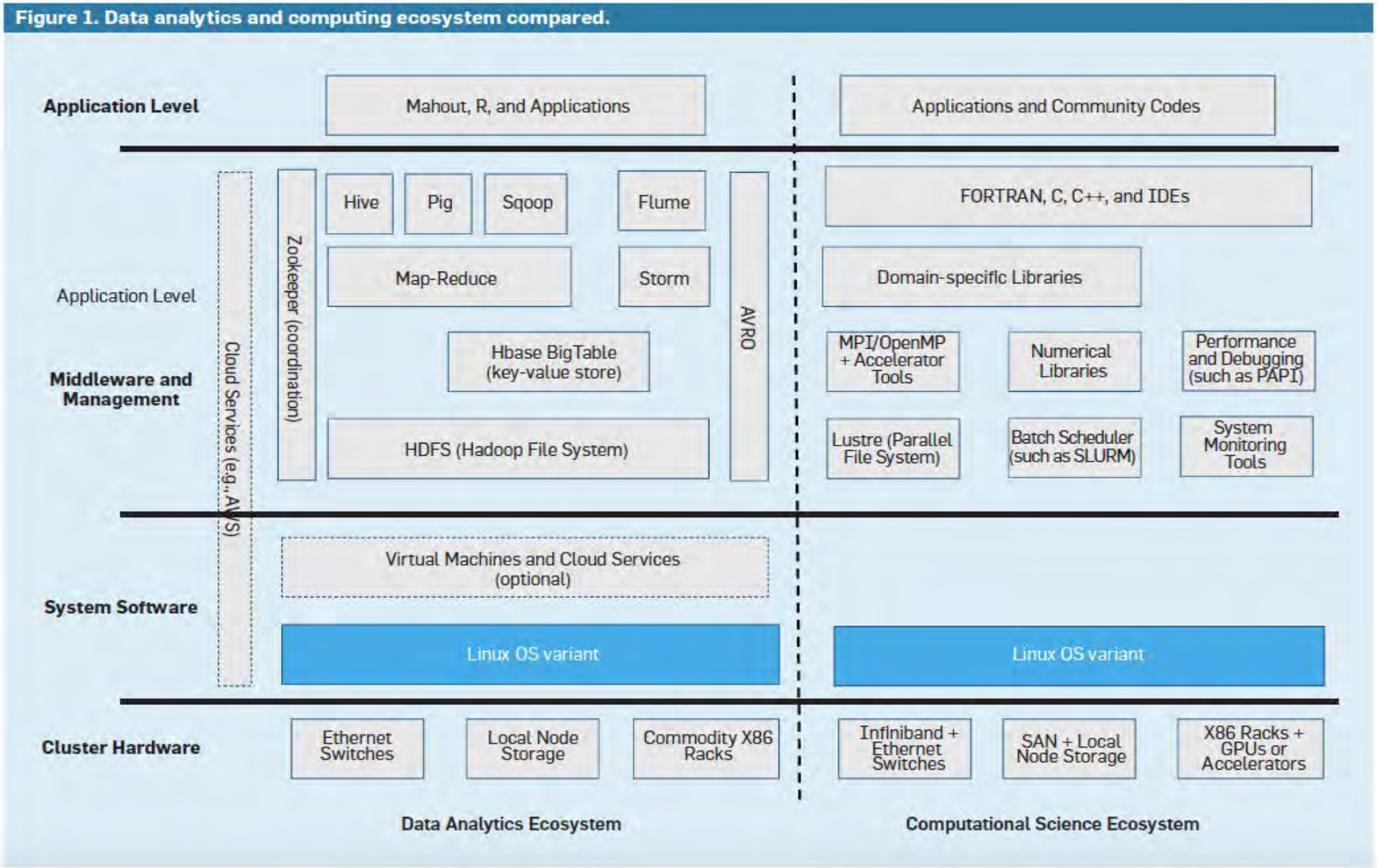
And If

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



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	EC 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	EC 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	EC 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	EC 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***

c/o BDEC break-out work product, 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 viz is 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

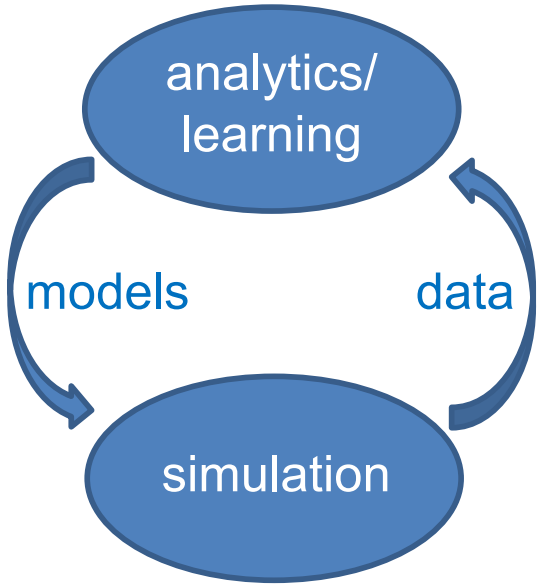
AI classification (unconventional)

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

after Eng Lim Goh (Chief Technologist, HPE)

Simulation and analytics: a cute couple

- **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

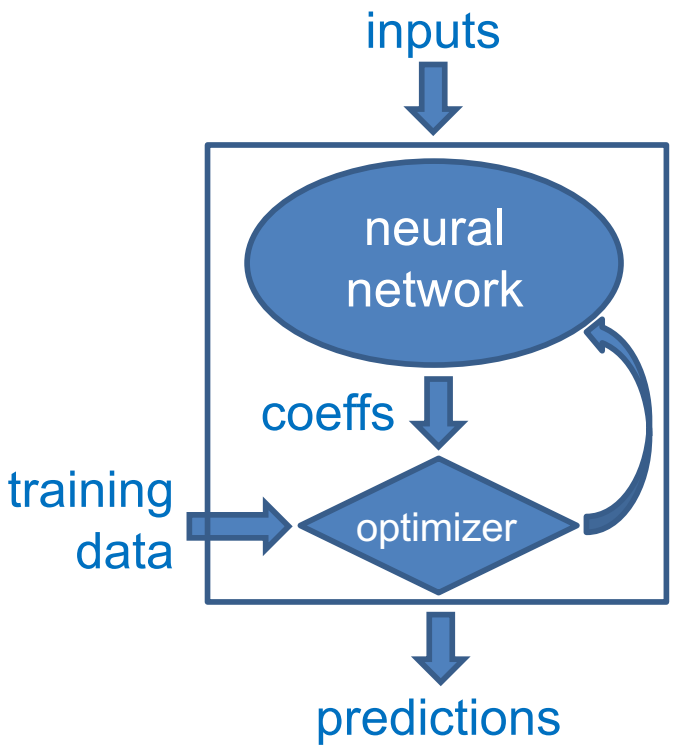


Simulation and learning: difference

- **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**

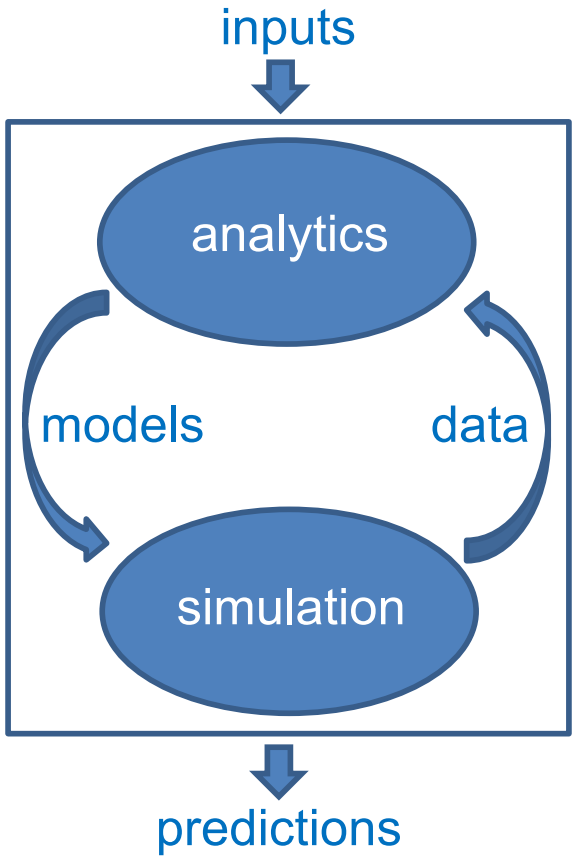
Simulation and learning: difference

- 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!

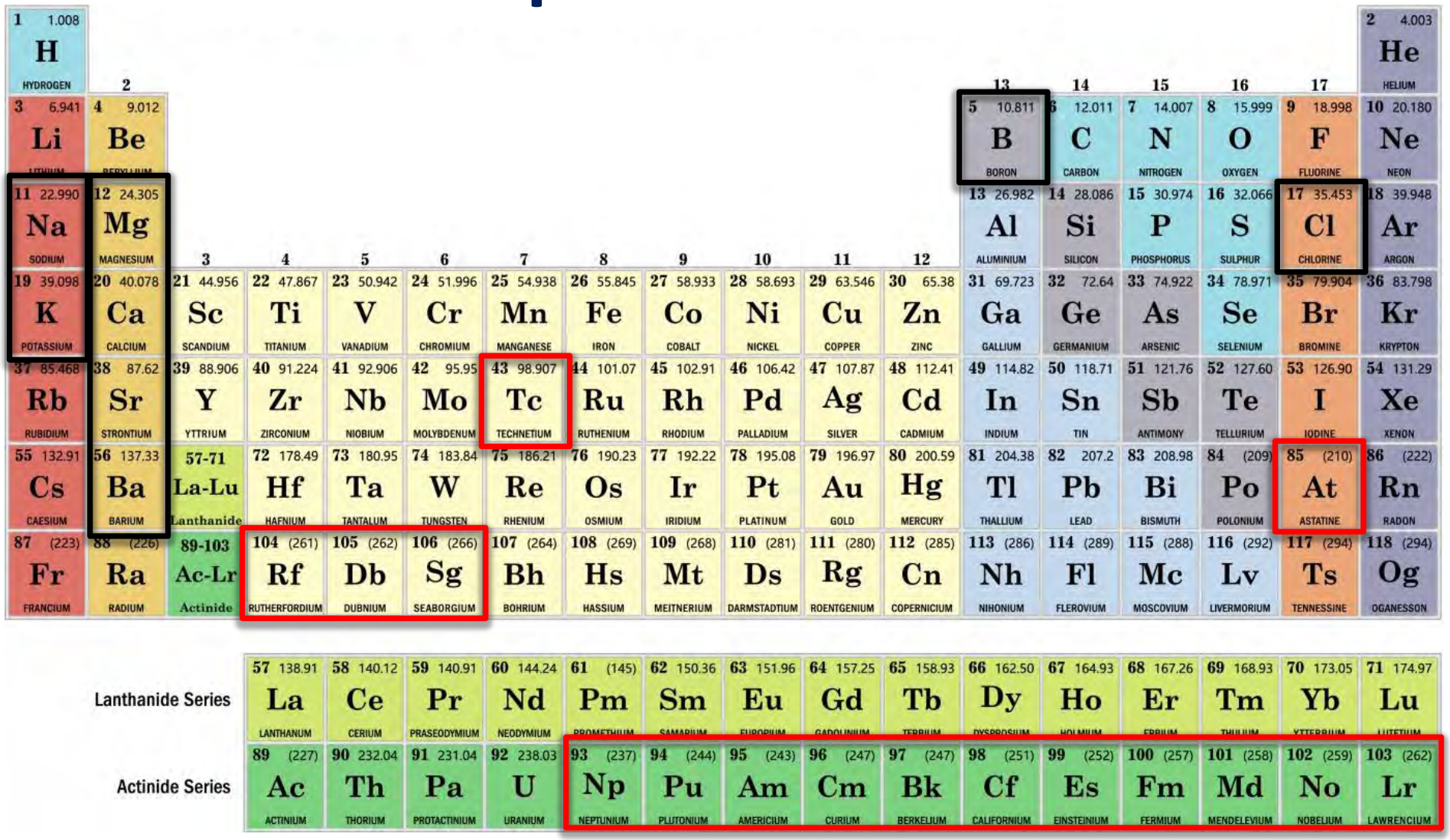
"Nothing tends so much to the advancement of knowledge as the application of a new instrument.

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)

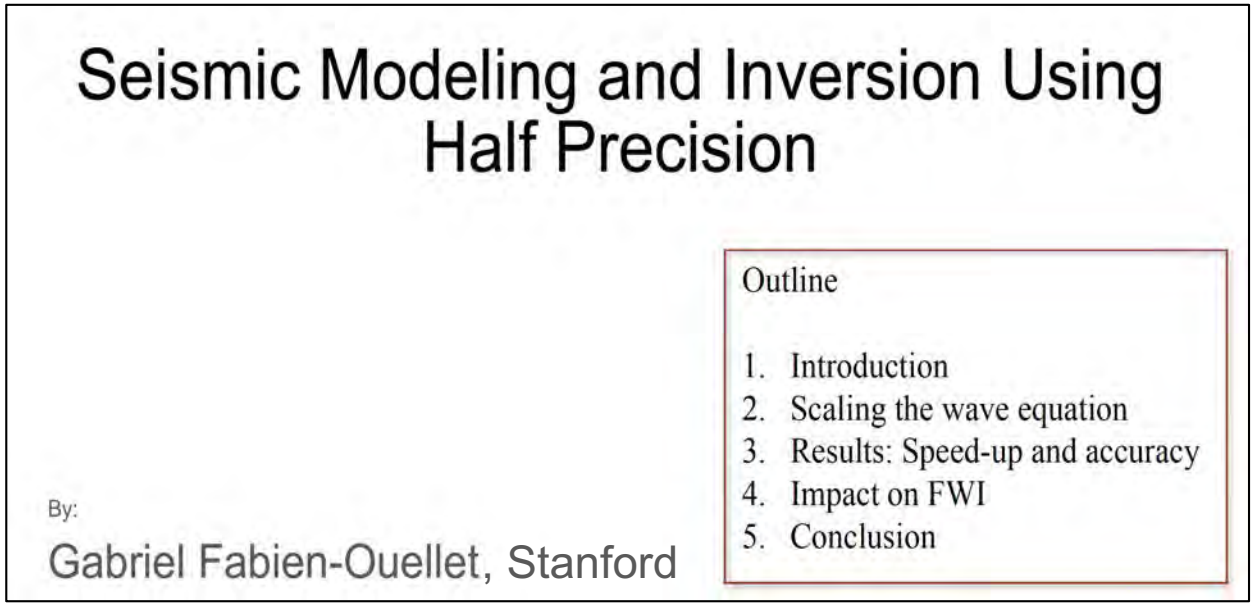
Inventor of electrochemistry (1802)
Discoverer of K, Na, Mg, Ca, Sr, Ba, B, Cl (1807-1810)

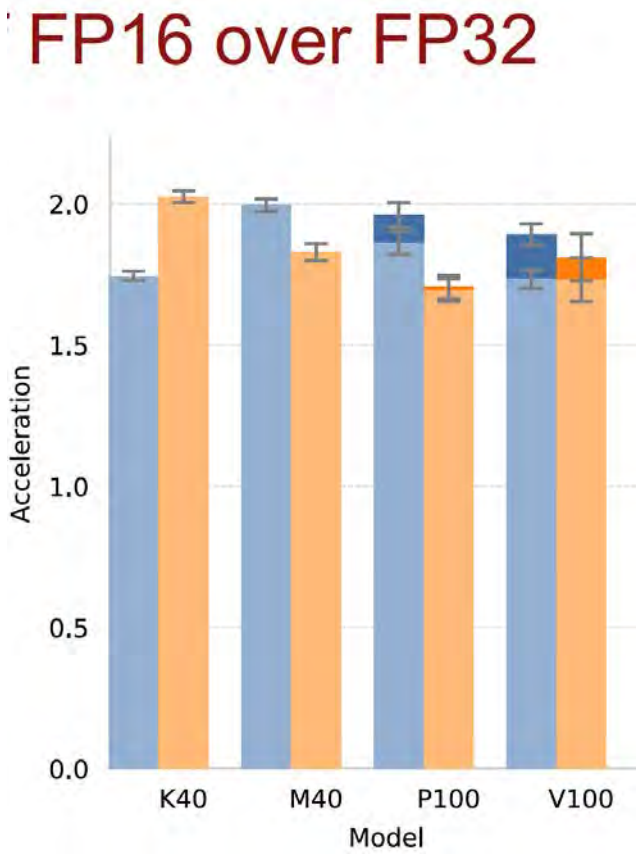
Davy's 1807-1010 "sprint" through the periodic table



+ Berkeley cyclotron (1931) elements

Bonus convergence benefit: Rethinking HPC in HDA datatypes





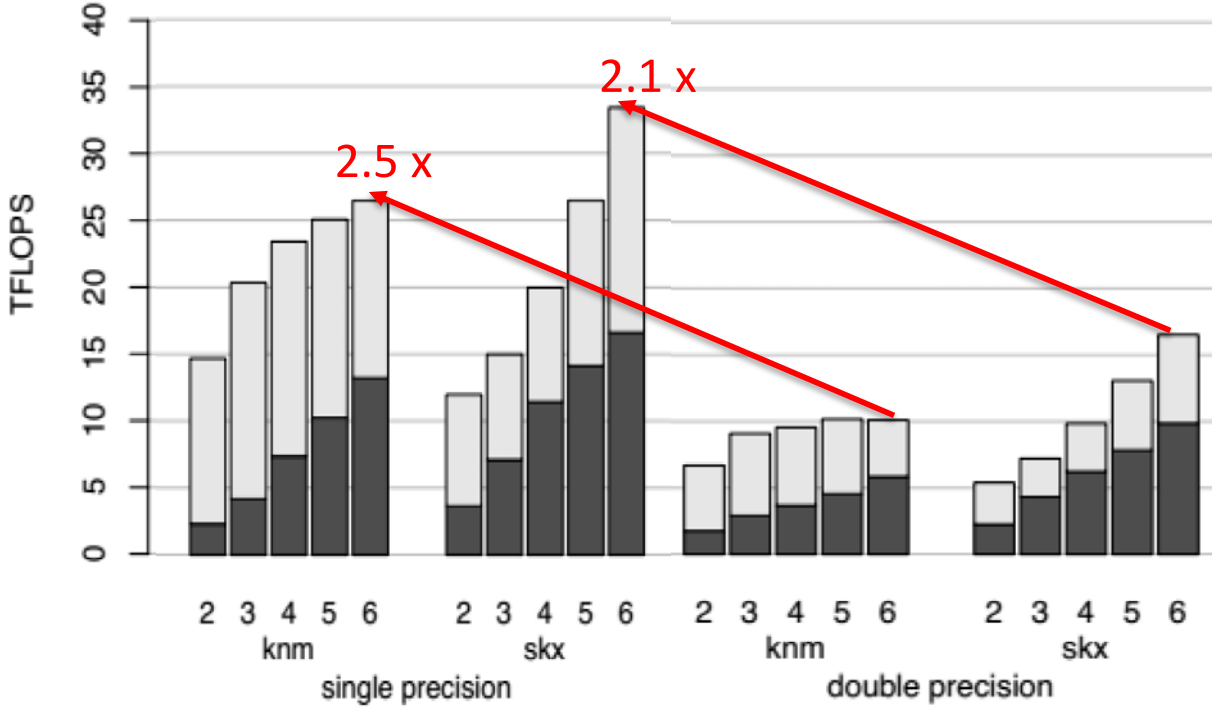
Fully acceptable accuracy in seismic imaging from single to half precision!

GTC 2018 Santa Clara

Bonus convergence benefit: Rethinking HPC in HDA datatypes

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

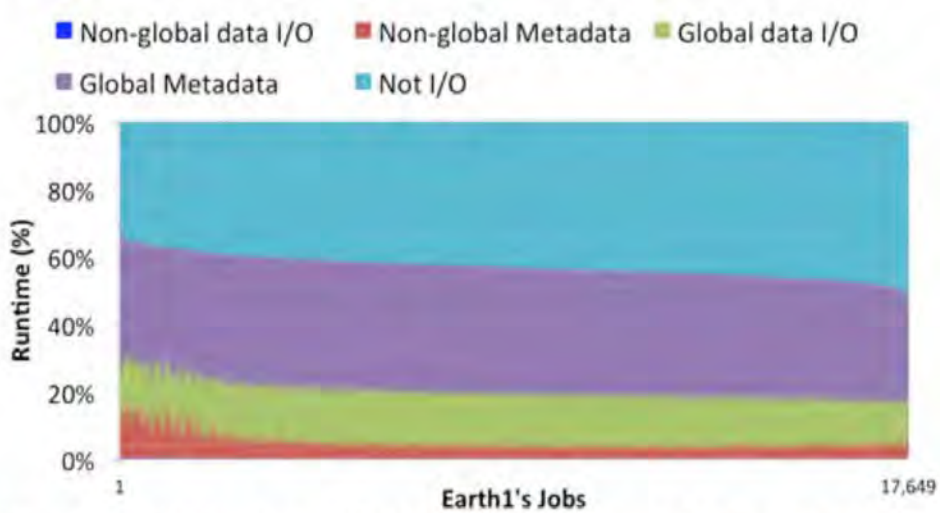


Figure 4: Breakdown of total run time for each Earth1 job.

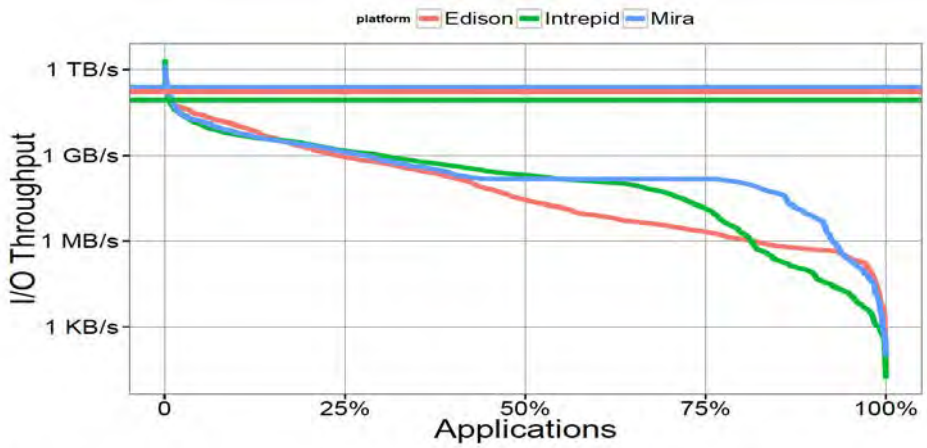
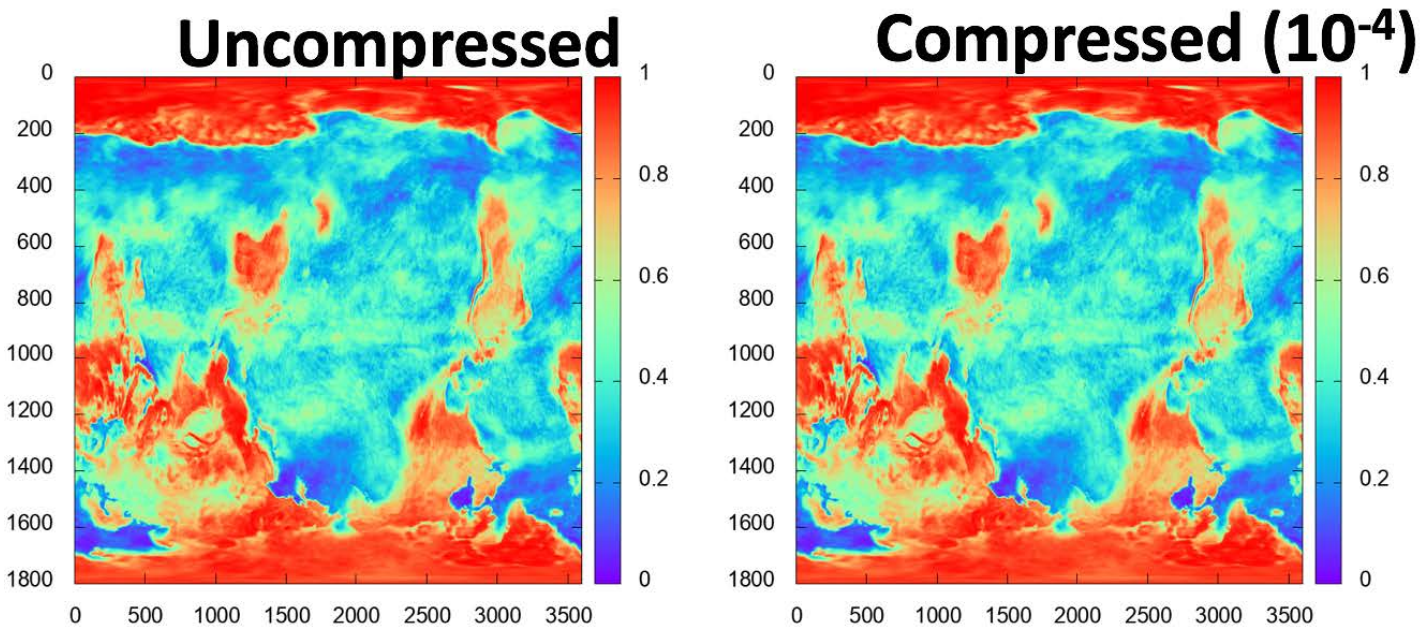


Figure 6: Maximum I/O throughput of each app across all its jobs on a platform, and platform peak I/O throughput.

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 on 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

Motivations for convergence

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

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 your desk – *if you can use it*." [me to you, 2021]**

- **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]**

Just two decades of evolution

1997

ASCI Red at Sandia

1.3 TF/s, 850 KW

2017

Cavium ThunderX2

~ 1.1 TF/s, ~ 0.2 KW

3.5 orders of magnitude

A vision for BDEC 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

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

Some additional attribute dimensions

- Real applications are often combinations of these three types of edge, federated, and combined
- Types of services used:
  - simulation, analytics, learning, sensing, actuation
- Off-line and real-time
- Open-loop and closed-loop
  - prediction vs. control
- Physical space-time environment and virtual space-time environment
- Human-in-the-loop and automated adaptation

Some goals for big data apps

- Simulation & learning to predict
- Simulation & learning to intervene
  - experimental or production automation
  - emergency response
- Assimilation of data in simulations to improve accuracy
  - minimize resources (e.g., # of simulations, amount of data transmitted) while achieving given predictive power

Services to compose in developing apps

- Simulation
  - PDEs, SVDs, Molecular dynamics, Lattice Boltzmann, Cellular Automota, agents, etc.
- Assimilation
  - Ensemble Kalman filters
- Optimization
  - Design, Control, Identification
- Uncertainty Quantification
- Reduced-order Modeling
- Digital Twins (to complete system definition)
- Observation
  - Microscopy, telescopy, satellites, ground penetrating radar, light sources, etc.
- Analytics
  - Data base queries
  - Image or sonic segmentation
  - Visualization
  - Regression
- Learning
  - Classification (supervised)
  - Clustering (unsupervised)

Some expected benefits of apps R&D

- **Provide direction to hardware architects (co-design for system balance)**
  - typical combinations (often multiply nested) of services
  - storage requirements
  - transmission requirements
- **Find cross-cutting applications of common tools**
  - for example, microscopy and satellite imagery
  - both are 2D image processing requiring segmentation, registration, automated identification, etc.

Promoting “natural” disruptions

- **“What we are doing now” is important, but…**
- **“What we really want to do” is more important**
- **As the custodians of the applications, we should define the terms *we* need and not simply “eat the crumbs” of commercial computing, so…**

Examples of disruptive questions

- **What in current HPC system job scheduling inhibits the campaigns we want to run**
  - **e.g., with persistently mounted databases?**
- **What services do we need to compose that we now have to pipeline through slow disk I/O, or worse?**
- **How can we transfer data between representations fluidly to exploit newly available techniques, e.g., for this data pipeline:**
  - **create visualizations of simulated materials to**
  - **apply image-oriented machine learning to**
  - **design beamline experiments for real materials**

Desired dimensions of a survey of apps

- **Find a minimum "basis set" that will suggest all of the required software architecture capabilities**
  - **A few deeply specified representative apps rather than a full but shallow shopping list**
- **Then find a comprehensive list of apps that will indicate where the activity is dense and the potential stakeholder payoff is greatest**
  - **Many (perhaps shallowly specified) apps that will leverage investment**

Domains of candidate applications

- Basic science
- Medical science
- Geospatial monitoring
- Engineering
- Manufacturing
- Societal infrastructure

Examples of composed applications

- **Precision agriculture merged with weather prediction**

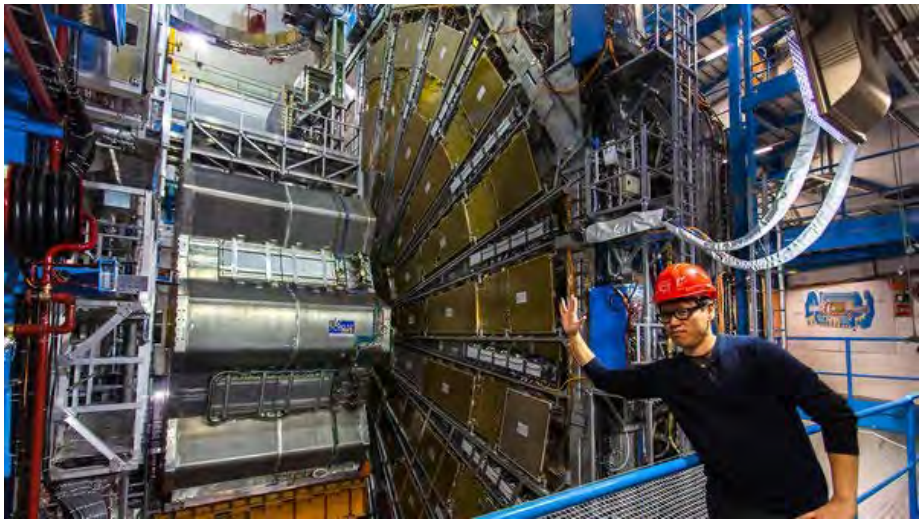
- **Windfarm power grid management merged with weather prediction**

- **Wildfire fighting merged with overhead imagery and weather prediction**

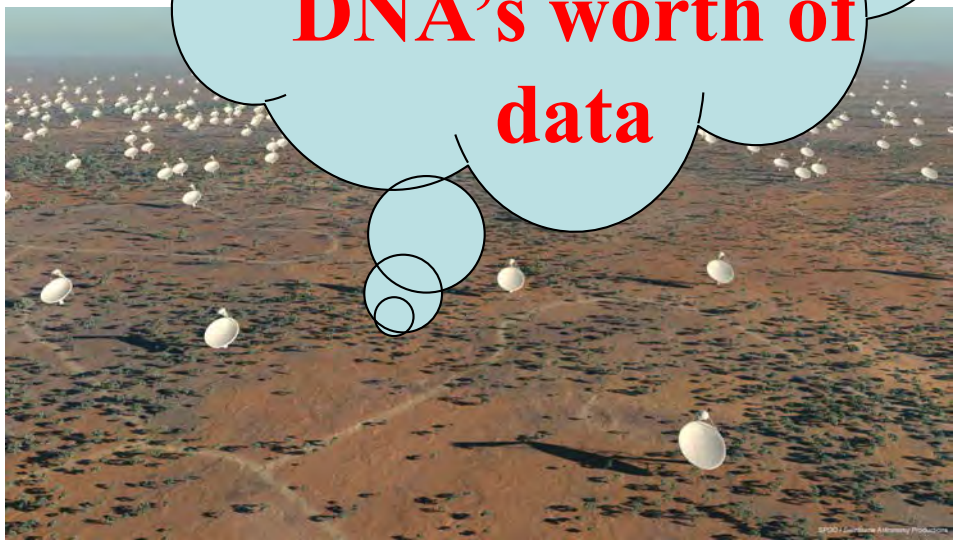
Extending convergence to the "edge"

- 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

The computing continuum

IoT/Edge — Fog — HPC/Cloud/Instrument

Size	Nano	Micro	Milli	Server	Fog	Campus	Facility
Example	Adafruit Trinket	Particle.io Boron	Array of Things	Linux Box	Co-located Blades	1000-node cluster	Datacenter
Memory	0.5K	256K	8GB	32GB	256G	32TB	16PB
Network	BLE	WiFi/LTE	WiFi/LTE	1 GigE	10GigE	40GigE	N*100GigE
Cost	$5	$30	$600	$3K	$50K	$2M	$1000M

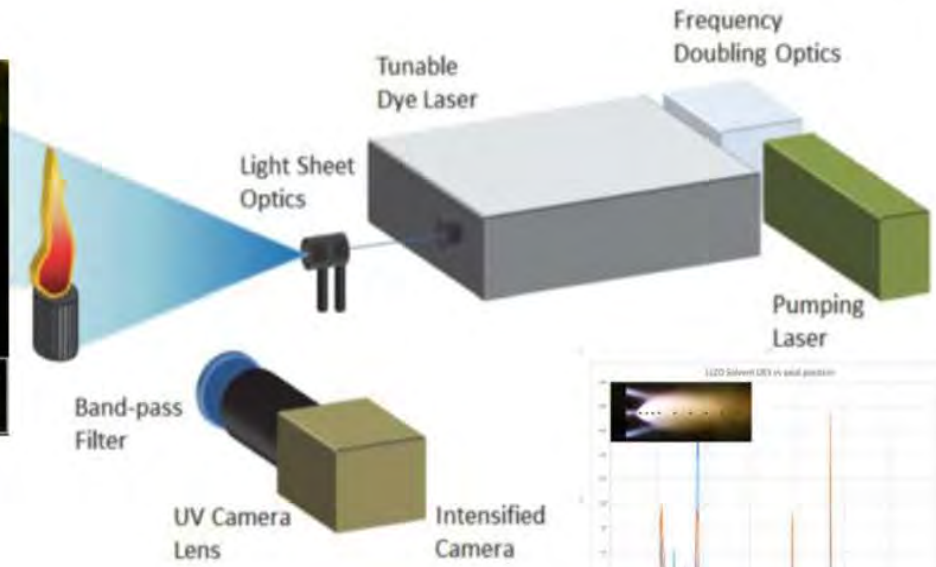
Count = 10^9
Size = 10^1

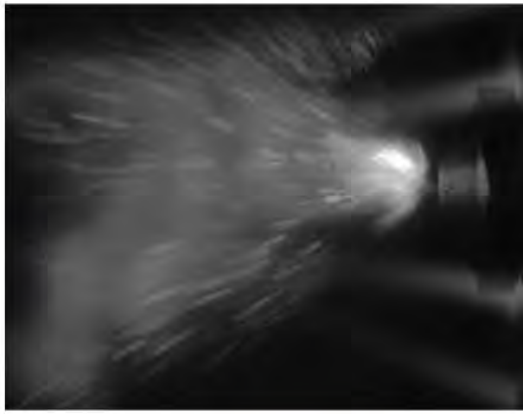
Count = 10^1
Size = 10^9

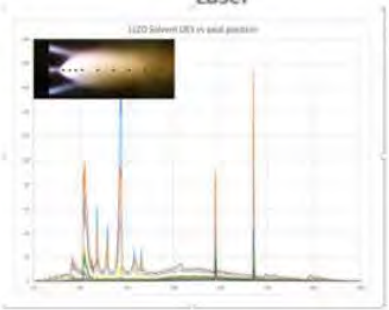
Figure 1: The Computing Continuum: Cyberinfrastructure that spans every scale. Components vary from small, inexpensive devices with limited computer resources (IoT) to modest priced servers with mid-range resources to expensive high performance computers with extensive compute, storage and network capabilities. This range of capabilities, cost, and numbers forms a continuum.

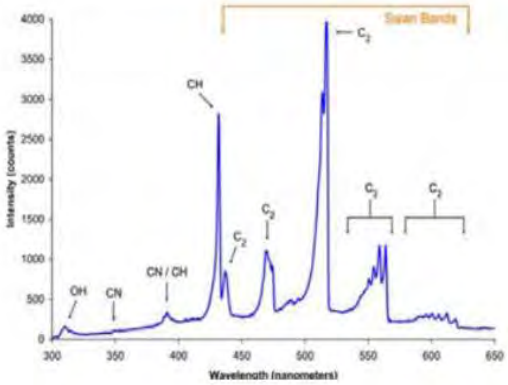
Edge computing in manufacturing

Example manufacturing process: Flame Spray Pyrolysis for functional nanostructured materials







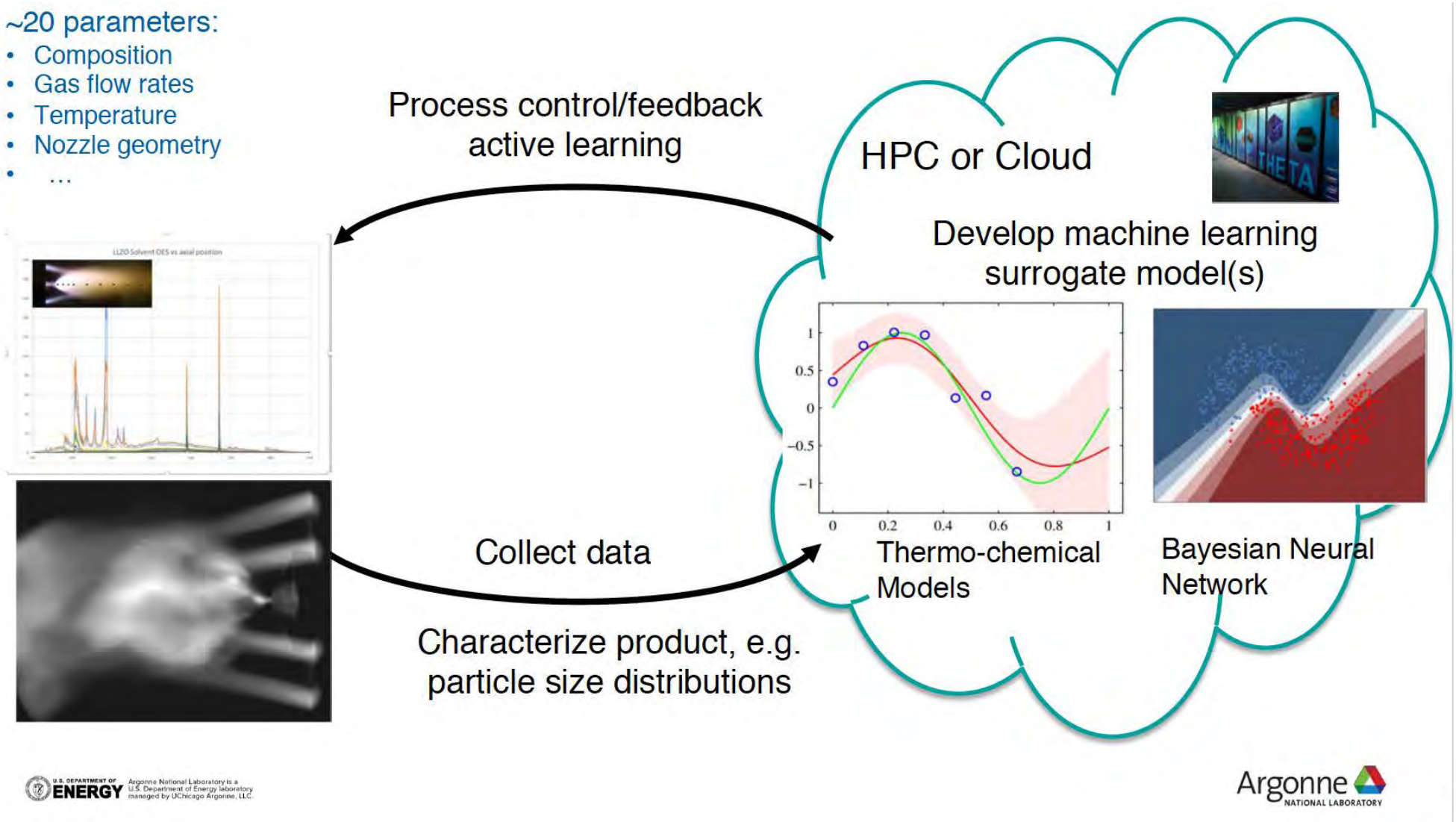


- Use data collected to date to develop ML/DL models
- Relate process parameters to output measures
  - Particle size analyzer, flame color, flame volume, optical emission spectrometer, Laser PLIF
- Optimize process

With J.Libera & S. Chaudhuri, Materials Engineering Research Facility, ANL

c/o Nicola Ferrier, Senior Computer Scientist, Argonne

Edge computing in manufacturing



c/o Nicola Ferrier, Senior Computer Scientist, Argonne

Big app: NEOM's "Cognitive City"


A baton pass
Paradigms Converged
3rd & 4th Paradigms Separate

References to the community reports

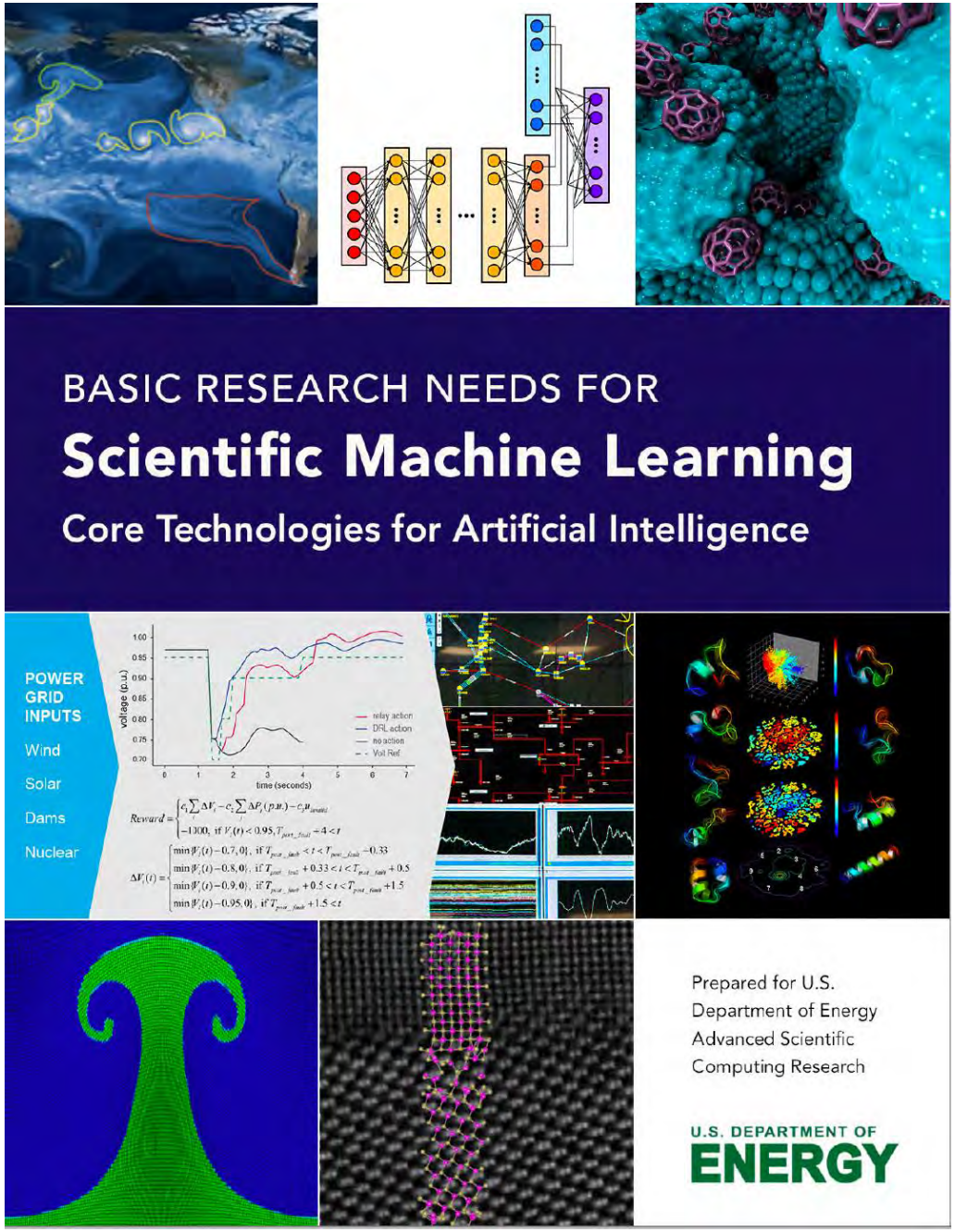
- **exascale.org/bdec**
  - http://www.exascale.org/bdec/sites/www.exascale.org.bdec/files/whitepapers/bdec2017pathways.pdf
  - "Big Data and Extreme-scale Computing: Pathways to Convergence," M. Asch, et al., *Int. J. High Perf. Comput. Applics.* **32**:435-479, 2018
- **exascale.org/iesp**
  - http://www.exascale.org/mediawiki/images/2/20/IESP-roadmap.pdf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

DOE report "SciML"

Feb 2019

460 references

109 pages



https://www.osti.gov/servlets/purl/1478744

Questions addressed in SciML report

- How should domain knowledge be modeled and represented in scientific ML?
- How can reproducibility be implemented in applications of scientific ML?
- Under what conditions is a scientific ML algorithm “well-posed”?
- How should robustness, performance, and quality of scientific ML be assessed?
- How can robust scientific ML be achieved with noisy data?
- How can ML be used to enable adaptive scientific computing?
- How can scientific computing expertise help scientific ML?
- How should ML be used to guide data acquisition?

An irony of the success of convergence

March 2021 *Nature Computational Science*

modeling is articulately defended with respect to machine learning ☺

The imperative of physics-based modeling and inverse theory in computational science

To best learn from data about large-scale complex systems, physics-based models representing the laws of nature must be integrated into the learning process. Inverse theory provides a crucial perspective for addressing the challenges of ill-posedness, uncertainty, nonlinearity and under-sampling.

Karen E. Willcox, Omar Ghattas and Patrick Heimbach

The notions of 'artificial intelligence (AI) for science' and 'scientific machine learning' (SciML) are gaining widespread attention in the scientific community. These initiatives target development and adoption of AI approaches in scientific and engineering fields with the goal of accelerating research and development breakthroughs in energy, basic science, engineering, medicine and national security. For the past six decades, these fields have been advanced through the synergistic and principled use of theory, experiments and physics-based simulations. Our increased ability to sense and acquire data is clearly a game-changer in these endeavors. Yet, in our excitement to define a new generation of data-centric approaches, we must be careful not to chart our course based entirely on the successes of data science and machine learning in the vastly different domains of social media, online entertainment, online retail, image recognition, machine translation and natural language processing — domains for which data are plentiful and physics-based models do not exist. In contrast, many of today's scientific grand challenges suffer from the lack of adequate sampling of the processes underlying the complex, large-scale systems. Yet, for many of these systems, a great deal is known regarding the underlying physical principles or governing equations; we must continue to appeal to computational science to unleash this information. As Coveney et al. argue elegantly[1], big data need big theory — and big physics-based simulation models — too.

The unreasonable effectiveness of physics-based models

But what are physics-based models and why are they indispensable? A physics-based model is a representation of the governing laws of nature that innately embeds the concepts of time, space, causality and generalizability. These laws of nature define how physical, chemical, biological and geological processes evolve. Physics-based models typically encode knowledge in the form of conservation and constitutive laws, often based on decades if not centuries of theoretical development and experimental validation. These laws often manifest as systems of differential equations that are solved numerically with high-performance computing (HPC).

In his famous 1960 article, Eugene Wigner wrote about 'The unreasonable effectiveness of mathematics in the natural sciences'[2], pointing to "the 'laws of nature' being of almost fantastic accuracy but of strictly limited scope." As Wigner discusses, physics-based modeling is powerful and effective because it gives us a predictive window into the future based on understanding. It achieves this because any particular model limits its scope to a particular class of physical systems or processes, building a universal representation within that class. Armed with that universal representation, physics-based modeling is a way to simulate 'what if' scenarios and to issue predictions that have explanatory power or projections with quantified uncertainties that go beyond the current state and available data. For example, in our modern world, physics-based models are used to issue predictions about the future evolution of a cancer patient's tumor, or about the loads that a yet to be built aircraft may find itself experiencing under different operating conditions. They enable predictions of weather over the next five to ten days, or scenario-based projections about the future state of the Earth's climate in the decades to come.

The role of inverse theory in learning from data

As attention turns from simulation to learning from data (that is, from the forward problem to the inverse problem), we must bring these learned lessons — the big theory and the big physics-based simulation models — with us. Without physical constraints, purely data-driven approaches are unlikely to be predictive, no matter how expressive the underlying representation. Even when physical models are not well-established (such as for many biological processes, in constitutive laws for complex materials, or in subgrid scale models for unresolved physics), we know that certain universal properties and relationships must hold, such as conservation properties, material frame indifference, objectivity, symmetries, or other invariants. The learning-from-data problem is fundamentally an inverse problem that merges the partial knowledge reservoir of data with that of physics-based models in a systematic and rigorous way, and in a way that exploits the complementary and mutually reinforcing aspects of both data and models.

Data and models invariably come with uncertainties. Data are often noisy, sparsely and heterogeneously sampled, and representative of disparate observables. Experiments and data gathering are costly, time-consuming, and sometimes dangerous or impossible. Often data are hardest to acquire and are thus sparsest in the most decision-critical regions (for example, failure, instability, extreme environments). Even if it is possible to generate more data (for example, via simulation) a fundamental challenge remains: due to information loss in the forward problem and resulting ill-posedness of the inverse problem, data often contain only low-dimensional information about the physics, even when the data are large-scale[3].

In turn, physics-based models are typically characterized by uncertain parameters, which may include initial and boundary conditions, sources, material properties, geometry and model structure, all of which can be heterogeneous in space or time. In this setting, rather than ignore known physics, we must employ them to define the maps from parameters to observables, and invert them to project

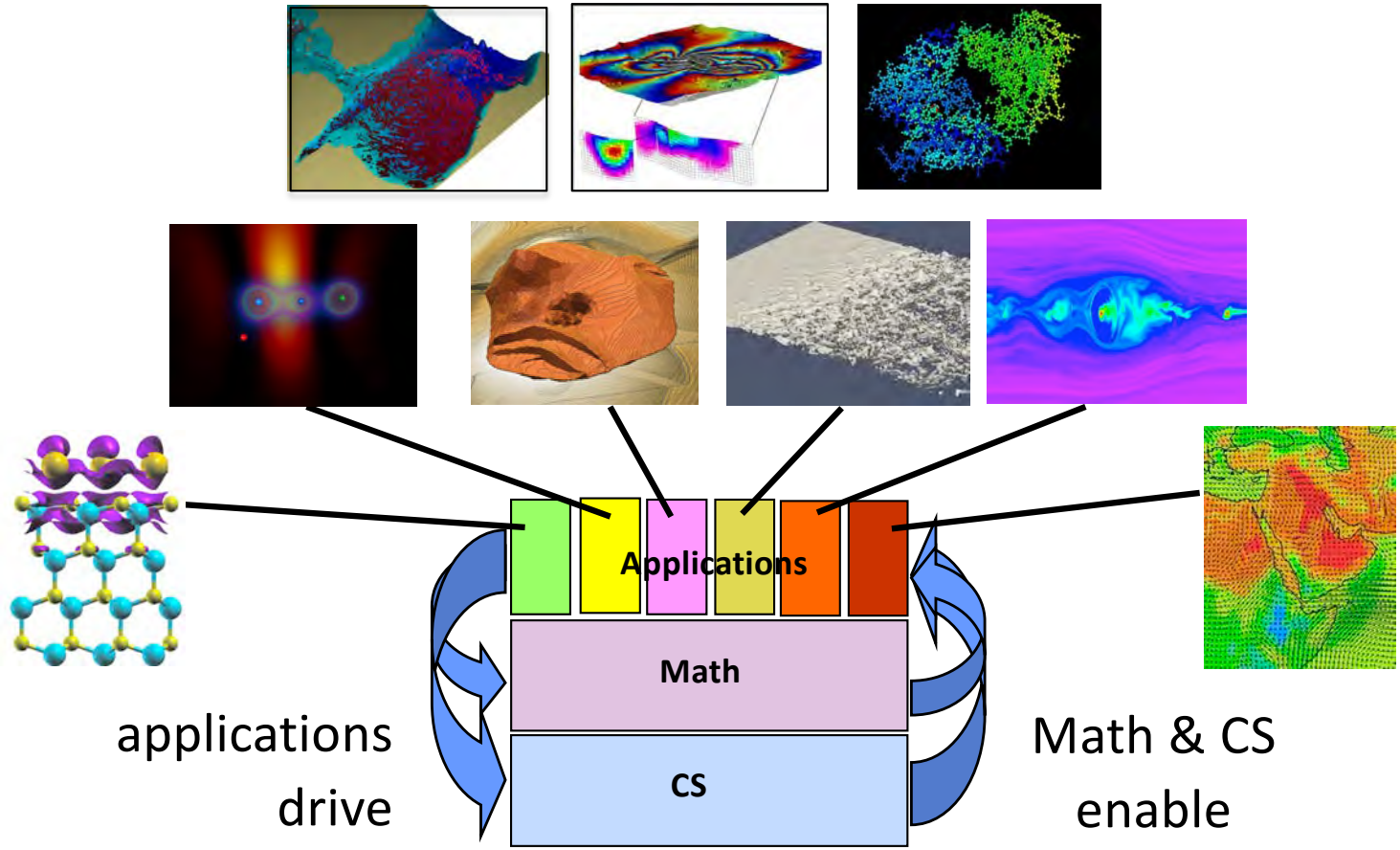
Summary convergence prediction

- **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**

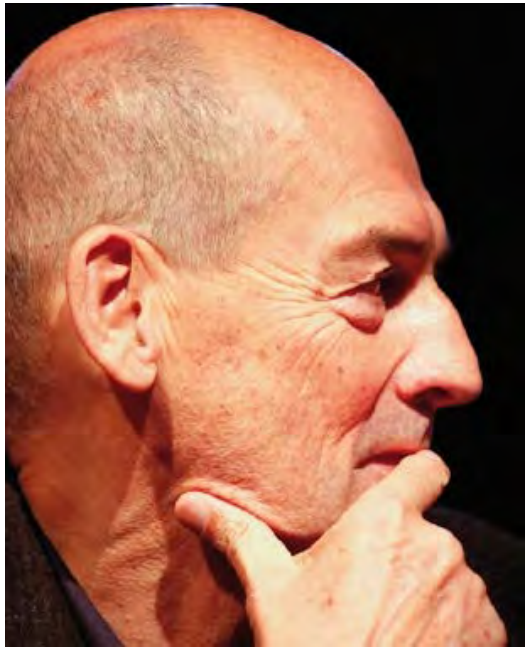
Overall motivations for series

- **Mathematical aesthetic**
  - Exascale algorithmics is beautiful
- **Engineering aesthetic**
  - Exascale algorithms tune *storage* and *work* to accuracy requirements
- **Software engineering aesthetic**
  - Cool stuff finds new important roles: direct and randomized floating point kernels, tree-traversal from FMM, task-based programming, etc.
- **Computer architecture requirement**
  - Emerging architectures are met on their terms: limited fast memory per core, SIMT instructions, etc.
- **Application opportunities (as cited)**
  - In simulation, big data analytics, machine learning and their combination

Applications are the visible impact



We are in the business of infrastructure



“Infrastructure is much more important than architecture.”

Rem Koolhaas (1944 –), architect

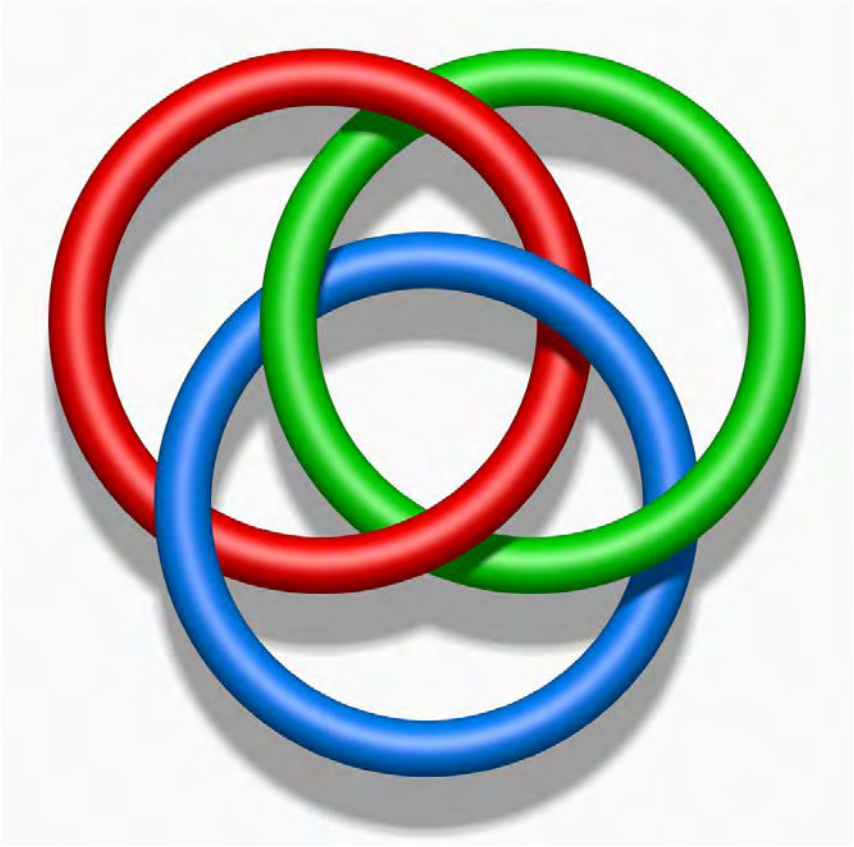
“The essential is invisible to the eyes.”

Antoine de Saint-Exupéry (1900 – 1944), author

Borromean Rings: A^3

Exascale computing is an interplay of

- **Applications**
- **Algorithms**
- **Architectures**
  - **Hardware**
  - **Software**



Remove any one ring and the others become unlinked

A “perfect storm” for exascale

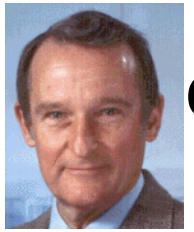
(dates are symbolic)

1686

scientific models

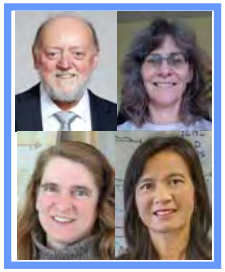
1947

numerical algorithms



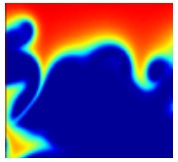
1976

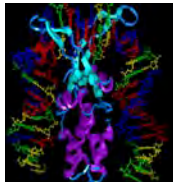
computer architecture

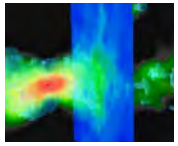


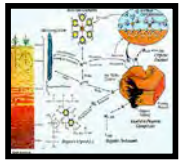
1992

scientific software engineering









The second baton pass

Bad news/good news

- **Must explicitly control more of the data motion**
  - **carries the highest energy and time cost in the exascale computational environment**
- **More opportunities to control the *vertical* data motion**
  - ***horizontal* data motion under control of users already**
  - **but vertical replication into caches and registers was (until recently) mainly scheduled and laid out by hardware and runtime systems, mostly invisibly to users**

Bad news/good news

- **Use of uniform high precision in nodal bases on dense grids may decrease, to save storage and bandwidth**
  - **representation of a smooth function in a hierarchical basis or on sparse grids or a kernel-based operator in hierarchical low rank requires fewer bits than storing its elemental values, for adequate accuracy**
- **We may compute and communicate "deltas" between states rather than the full state quantities**
  - **as when double precision was once expensive (e.g., iterative correction in linear algebra)**
  - **a generalized "combining network" node or a smart memory controller may remember the last address and the last value, and forward just the delta**
- **Equidistributing errors properly to minimize resource use will lead to innovative error analyses in numerical analysis**

Bad news/good news

- **Fully deterministic algorithms may come to be regarded as too synchronization-vulnerable**
  - beyond unrolling into task graphs, rather than wait for missing data we may predict it using various means and continue
  - we do this with increasing success in problems without models ("big data")
  - should be fruitful in problems coming from continuous models
  - "apply machine learning to the simulation machine"
- **A rich numerical analysis of algorithms that make use of statistically inferred "missing" quantities may emerge**
  - future sensitivity to poor predictions can often be estimated
  - numerical analysts will use statistics, signal processing, ML, etc.

Bad news/good news

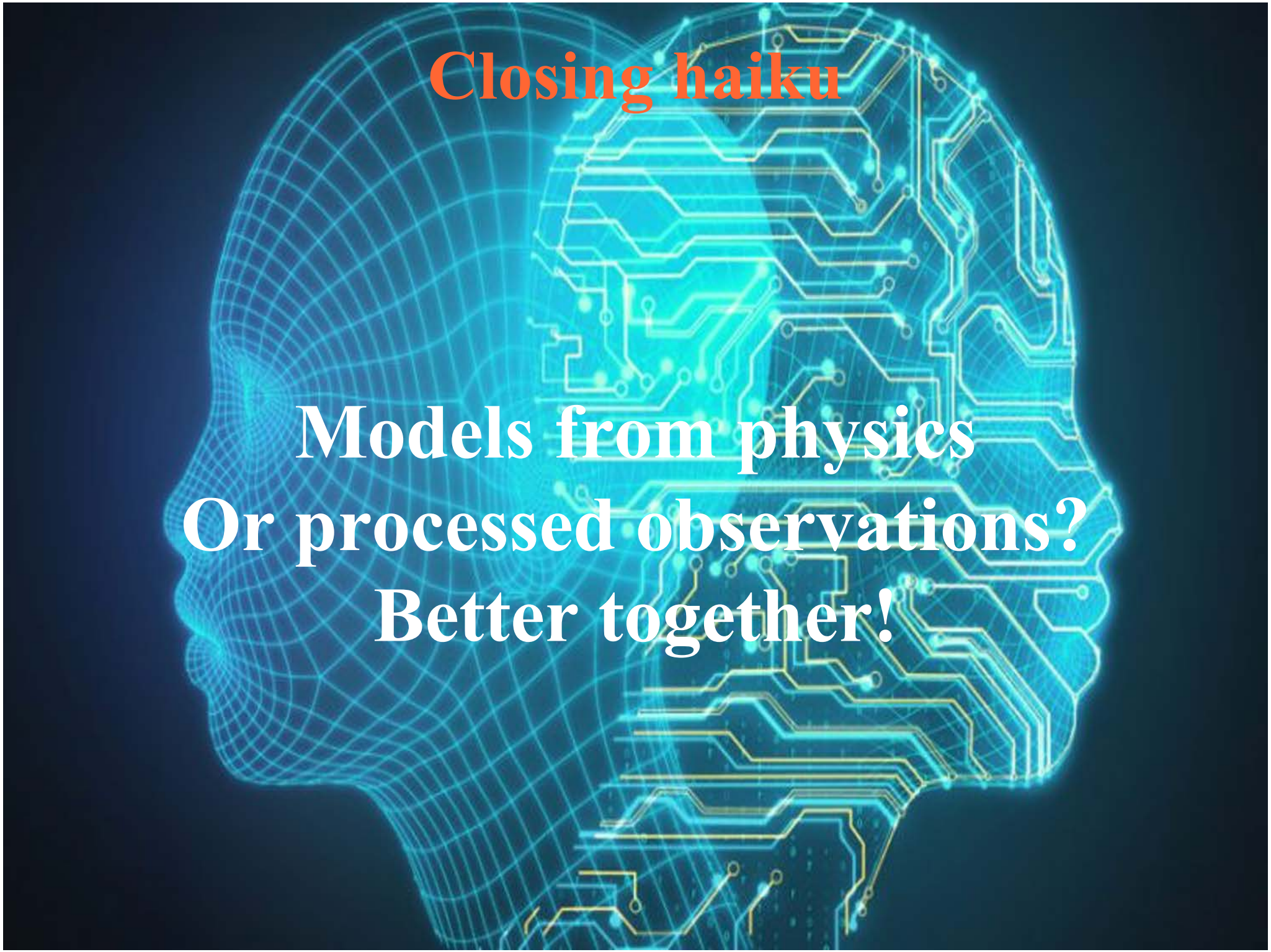
- Fully hardware-reliable executions may be regarded as too costly
- Algorithmic-based fault tolerance will be cheaper than hardware and OS-mediated reliability
  - developers will partition their data and their program units into two sets
    - a small set that must be done reliably (with today's standards for memory checking and IEEE ECC)
    - a large set that can be done fast and unreliably, knowing the errors can be either detected, or their effects rigorously bounded
- Many examples in direct* and iterative** linear algebra
- Anticipated by Von Neumann, 1956 ("Synthesis of reliable organisms from unreliable components")

---

*e.g., using checksums to detect

** e.g., using FGMRES to repair

Closing haiku

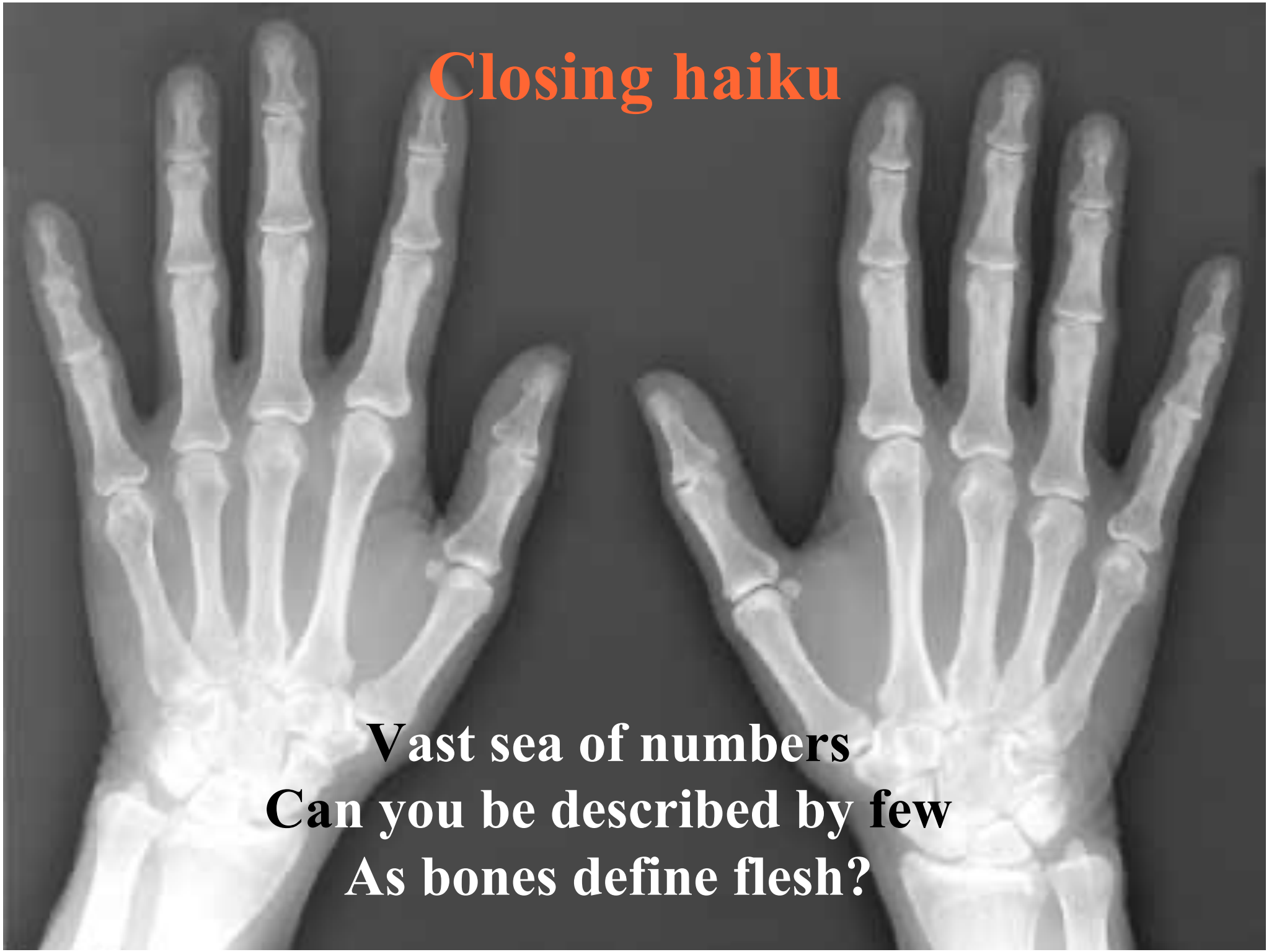


Models from physics
Or processed observations?
Better together!

Closing haiku

Covariances
In the billions require
ExaGeoStat

Closing haiku



Vast sea of numbers
Can you be described by few
As bones define flesh?

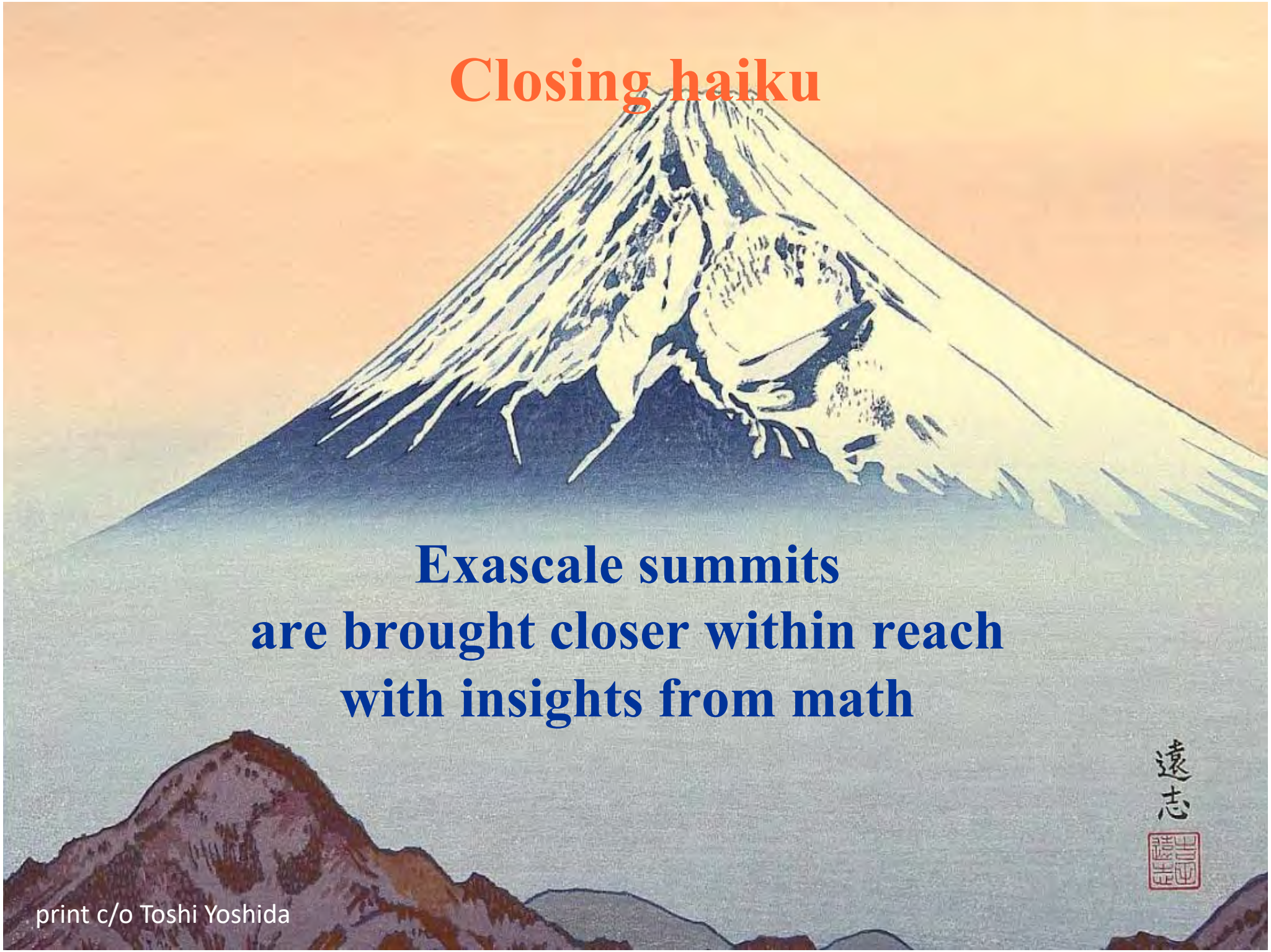
Closing haiku

Curse of dimension,

Can you be mitigated

By low rank's blessing?

Closing haiku



Exascale summits
are brought closer within reach
with insights from math

print c/o Toshi Yoshida

Thank you!

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

شكرا

david.keyes@kaust.edu.sa