Lecture 05: The Convergence of Big Data and Extreme Computing

David Keyes
King Abdullah University of Science and Technology, david.keyes@kaust.edu.sa

Follow this and additional works at: https://scholarworks.uark.edu/mascsls

Part of the Artificial Intelligence and Robotics Commons, Harmonic Analysis and Representation Commons, Numerical Analysis and Computation Commons, Numerical Analysis and Scientific Computing Commons, and the Systems Architecture Commons

Citation

This Video is brought to you for free and open access by the Mathematical Sciences at ScholarWorks@UARK. It has been accepted for inclusion in Mathematical Sciences Spring Lecture Series by an authorized administrator of ScholarWorks@UARK. For more information, please contact scholar@uark.edu.
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

- **pre-computational**
- **theory**
- **simulation**
- **big data**
- **computational**

Timeline:
- **Galileo**
- **“Humboldt model” 1850’s**
- **Today**

Greek and computational paradigms evolution.
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 3\textsuperscript{rd} and 4\textsuperscript{th} paradigms

\textit{Big Data and Extreme Computing: Pathways to Convergence} (2017)
downloadable at exascale.org

successor to the 2011 \textit{International Exascale Software Roadmap}

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.

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

- 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 \textit{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 3\textsuperscript{rd}/4\textsuperscript{th} paradigms: from forward to inverse problems

forward problem

- model
- forcing
- ICs
- BCs

inverse problem

- model
- forcing
- ICs
- BCs

- 'solution'
- plus regularization
- params
A traditional combination of $3^{rd}/4^{th}$ paradigms: data assimilation

Theory

- Fully Nonlinear Filters
- Dual Filters Coupled Models
- Robust Ensemble Filters
- Hybrid Adjoint-Ensemble Filters

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

Photo credit: Publicis
Coming interactions between paradigms
opportunities of *in situ* convergence

<table>
<thead>
<tr>
<th></th>
<th>To Simulation</th>
<th>To Analytics</th>
<th>To Learning</th>
</tr>
</thead>
<tbody>
<tr>
<td>3rd</td>
<td>Simulation provides</td>
<td>—</td>
<td></td>
</tr>
<tr>
<td>4th (a)</td>
<td>Analytics provides</td>
<td></td>
<td>—</td>
</tr>
<tr>
<td>4th (b)</td>
<td>Learning provides</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Coming interactions between paradigms opportunities of *in situ* convergence

<table>
<thead>
<tr>
<th></th>
<th>To Simulation</th>
<th>To Analytics</th>
<th>To Learning</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>3rd</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Simulation provides</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>4th (a)</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Analytics provides</td>
<td>Steering in high dimensional parameter space; <em>In situ</em> processing</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>4th (b)</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Learning provides</td>
<td>Smart data compression; Replacement of models with learned functions</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Coming interactions between paradigms
opportunities of *in situ* convergence

<table>
<thead>
<tr>
<th></th>
<th>To Simulation</th>
<th>To Analytics</th>
<th>To Learning</th>
</tr>
</thead>
<tbody>
<tr>
<td>3rd</td>
<td>Simulation provides</td>
<td></td>
<td>Physics-based “regularization”</td>
</tr>
<tr>
<td>4th (a)</td>
<td>Analytics provides</td>
<td>Steering in high dimensional parameter space; <em>In situ</em> processing</td>
<td></td>
</tr>
<tr>
<td>4th (b)</td>
<td>Learning provides</td>
<td>Smart data compression; Replacement of models with learned functions</td>
<td></td>
</tr>
</tbody>
</table>
## Coming interactions between paradigms

opportunities of *in situ* convergence

<table>
<thead>
<tr>
<th></th>
<th>To Simulation</th>
<th>To Analytics</th>
<th>To Learning</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>3rd</strong></td>
<td>Simulation provides</td>
<td>—</td>
<td>Physics-based “regularization”</td>
</tr>
<tr>
<td><strong>4th (a)</strong></td>
<td>Analytics provides</td>
<td>Steering in high dimensional parameter space; <em>In situ</em> processing</td>
<td>—</td>
</tr>
<tr>
<td><strong>4th (b)</strong></td>
<td>Learning provides</td>
<td>Smart data compression; Replacement of models with learned functions</td>
<td>Imputation of missing data; Detection and classification</td>
</tr>
</tbody>
</table>
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 3\textsuperscript{rd} and 4\textsuperscript{th} paradigms

Figure 1. Data analytics and computing ecosystem compared.

Application Level
- Mahout, R, and Applications
- Applications and Community Codes

Zookeeper (configuration)
- Hive
- Pig
- Sqoop
- Flume
- Storm

Middleware and Management
- Map-Reduce
- Hbase BigTable (key-value store)
- HDFS (Hadoop File System)

AVRO
- 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

Cluster Hardware
- Ethernet Switches
- Local Node Storage
- Commodity X86 Racks
- Intrahub + Ethernet Switches
- SAN + Local Node Storage
- X86 Racks + GPUs or Accelerators

Data Analytics Ecosystem

Computational Science Ecosystem

\textit{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
  - HPC: exclusive space
  - HDA: interactive
  - 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

Comparing Architecture

<table>
<thead>
<tr>
<th>Big Data</th>
<th>BDEC Extreme Computing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cost in memory and interconnect bandwidth</td>
<td>Significant Cost in memory and interconnect bandwidth</td>
</tr>
<tr>
<td>Little Cost for resilient hardware in data storage</td>
<td>Significant Cost in resilient hardware in shared file system</td>
</tr>
<tr>
<td>Little Cost for hardware to support system-wide resilience</td>
<td>Significant Cost in resilience hardware to reduce whole-system MTTI</td>
</tr>
<tr>
<td>Significant Cost: increased aggregate IOPs</td>
<td>Significant Cost: cutting-edge CPU performance features</td>
</tr>
<tr>
<td>Often trades performance for capacity</td>
<td>Often trades capacity for performance</td>
</tr>
</tbody>
</table>

Comparing Operations

<table>
<thead>
<tr>
<th>Big Data</th>
<th>BDEC Extreme Computing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Continuous access to long-lived &quot;services&quot; created by science community</td>
<td>Periodic access to compute resources via job submitted to scheduler and queue</td>
</tr>
<tr>
<td>Time-shared access to elastic resources</td>
<td>Space-shared compute resources for exclusive access during jobs</td>
</tr>
<tr>
<td>New hardware capacity purchased incrementally</td>
<td>New tightly integrated system purchased every 4 years</td>
</tr>
<tr>
<td>Users charged for all resources (storage, CPU, networking)</td>
<td>Users charged for CPU hours, storage and networking is free</td>
</tr>
</tbody>
</table>

Comparing Software

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

Comparing Data

<table>
<thead>
<tr>
<th>Scientific Big Data</th>
<th>BDEC Extreme Computing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inputs arrive continuously, streaming workflows</td>
<td>Inputs arrive infrequently, buffering carefully managed</td>
</tr>
<tr>
<td>Data is unrepeatable snapshot in time</td>
<td>Data often reproducible (repeat simulation)</td>
</tr>
<tr>
<td>Data generated by sensors</td>
<td>Data generated from simulation (error: from simulation)</td>
</tr>
<tr>
<td>Data rate limited by sensors</td>
<td>Data rate limited by platform</td>
</tr>
<tr>
<td>Data often shared and curated by community</td>
<td>Data often private</td>
</tr>
<tr>
<td>Often unstructured</td>
<td>Semi-structured</td>
</tr>
</tbody>
</table>

c/o BDEC break-out work product, following J. Ahrens, LANL
Extra motivations for convergence

- Vendors wish to unify their offerings
  - traditionally 3\textsuperscript{rd} paradigm-serving vendors are now market-dominated by the 4\textsuperscript{th}

- Under all hardware scenarios, data movement is much more expensive than computation
  - simulation and analytics should be done \textit{in situ}, with each other on in-memory (in-cache?) data
  - exchange in the form of exchange of files between 3\textsuperscript{rd} and 4\textsuperscript{th} 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
- Predict categories: Classification & Clustering

Linear
Nonlinear, Max likelihood
Supervised labeled data: Classification

Bayesian
Decision tree

Neural networks & Deep learning
K-means

Unsupervised unlabeled data: Clustering

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
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
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.
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”

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

Seismic Modeling and Inversion Using Half Precision

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

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.
Bonus convergence benefit: Data center economy

Reduce the space burden of I/O

Uncompressed

Compressed (10^{-4})

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

<table>
<thead>
<tr>
<th>Simulation</th>
<th>Observation</th>
</tr>
</thead>
<tbody>
<tr>
<td>PDEs, SVDs, Molecular dynamics, Lattice Boltzmann, Cellular Automata, agents, etc.</td>
<td>Microscopy, telescopy, satellites, ground penetrating radar, light sources, etc.</td>
</tr>
<tr>
<td>Assimilation</td>
<td>Analytics</td>
</tr>
<tr>
<td>Ensemble Kalman filters</td>
<td>Data base queries</td>
</tr>
<tr>
<td>Optimization</td>
<td>Image or sonic segmentation</td>
</tr>
<tr>
<td>Design, Control, Identification</td>
<td>Visualization</td>
</tr>
<tr>
<td>Uncertainty Quantification</td>
<td>Regression</td>
</tr>
<tr>
<td>Reduced-order Modeling</td>
<td>Learning</td>
</tr>
<tr>
<td>Digital Twins (to complete system definition)</td>
<td>Classification (supervised)</td>
</tr>
<tr>
<td></td>
<td>Clustering (unsupervised)</td>
</tr>
</tbody>
</table>
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 the edge.
- Need lightweight machine learning at the edge to downsize the data.

**SKA (dishes pictured)**

1 TB/s, 31 EB/yr, red to 3 EB/yr

**CERN (ATLAS pictured)**

25 GB/s, 780 PB/yr

**SKA will produce annually about 6 global human DNA’s worth of data**
### The computing continuum

<table>
<thead>
<tr>
<th>IoT/Edge</th>
<th>Fog</th>
<th>HPC/Cloud/Instrument</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Size</strong></td>
<td><strong>Nano</strong></td>
<td><strong>Micro</strong></td>
</tr>
<tr>
<td>Example</td>
<td>Adafruit Trinket</td>
<td>Particle.io Boron</td>
</tr>
<tr>
<td>Memory</td>
<td>0.5K</td>
<td>256K</td>
</tr>
<tr>
<td>Network</td>
<td>BLE</td>
<td>WiFi/LTE</td>
</tr>
<tr>
<td>Cost</td>
<td>$5</td>
<td>$30</td>
</tr>
</tbody>
</table>

**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

~20 parameters:
- Composition
- Gas flow rates
- Temperature
- Nozzle geometry
- ...

Process control/feedback active learning

HPC or Cloud

Develop machine learning surrogate model(s)

Collect data

Thermo-chemical Models

Characterize product, e.g. particle size distributions

Bayesian Neural Network

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**

- **exascale.org/iesp**
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 😊
Summary convergence prediction

• No need to force a “shotgun” marriage of “convergence” between 3\textsuperscript{rd} and 4\textsuperscript{th} paradigms
  - a love-based marriage is inevitable in the near future

• Driver will be opportunity for both 3\textsuperscript{rd} and 4\textsuperscript{th} 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

applications drive

Math & CS enable

Math

cs

Applications
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

(science models)

dates are symbolic)

numerical algorithms

computer architecture

scientific software engineering

1686

1947

1976

1992
The second baton pass

Energy
austere

Bulk
synchronous
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
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
Models from physics
Or processed observations?
Better together!

Closing haiku
Covariances
In the billions require
ExaGeoStat

Closing haiku
Closing haiku

Vast sea of numbers
Can you be described by few
As bones define flesh?
Curse of dimension,
Can you be mitigated
By low rank’s blessing?

Closing haiku
Closing haiku

Exascale summits
are brought closer within reach
with insights from math

print c/o Toshi Yoshida
Thank you!

 شكرا

david.keyes@kaust.edu.sa