Data-intensive and Simulation-intensive Computing: Parallel or Perpendicular?

BIG DATA AND EXTREME COMPUTING APPLICATIONS BREAKOUT REPORT*
29-30 JANUARY 2015, BARCELONA

DAVID KEYES, KAUST
ROSA BADIA, BSC (CO-CHAIR)

BIG DATA AND EXTREME-SCALE COMPUTING

*Presented at SC'15 in the BOF "Big Data and Exascale Computing" and in the Tokyo Institute of Technology booth talk series on big data

Barcelona applications breakout participants, January 2015

- James Ahrens
- Jean-Claude André
- Rosa Badia
- Thierry Bidot
- Mathis Bode
- Fernando Cucchietti
- Anshu Dubey
- Françoise Genova
- David Keyes
- Jysoo Lee
- Takemasa Miyoshi
- Morris Riedel
- Alejandro Ribés
- Joel Saltz
- Alex Szalay
- Bill Tang
- Akira Ukawa
- Pier Luigi Vidale

Profile of the authors

- Nations represented
  - Asia: Japan, Korea
  - Americas: USA
  - EU: France, Germany, Iceland, Spain, UK
  - Middle East: Saudi Arabia
- Disciplines represented
  - PDE-based: combustion, climate/meteorology, fusion
  - Observation-based: astronomy, biomedical informatics
  - Tools-based: PDE frameworks, runtime systems

Comparing "Numerically Intensive" and "Data Intensive" High Performance Computing

- Both numerically intensive (NI) and data intensive (DI) approaches share the common challenge of gaining scientific insights, making prediction, and quantifying uncertainty
  - NI primarily through first principles models
  - DI primarily through statistical models
- Disclaimer: these labels are imperfect; the right labels are a "work in progress"

Application types (1/2)

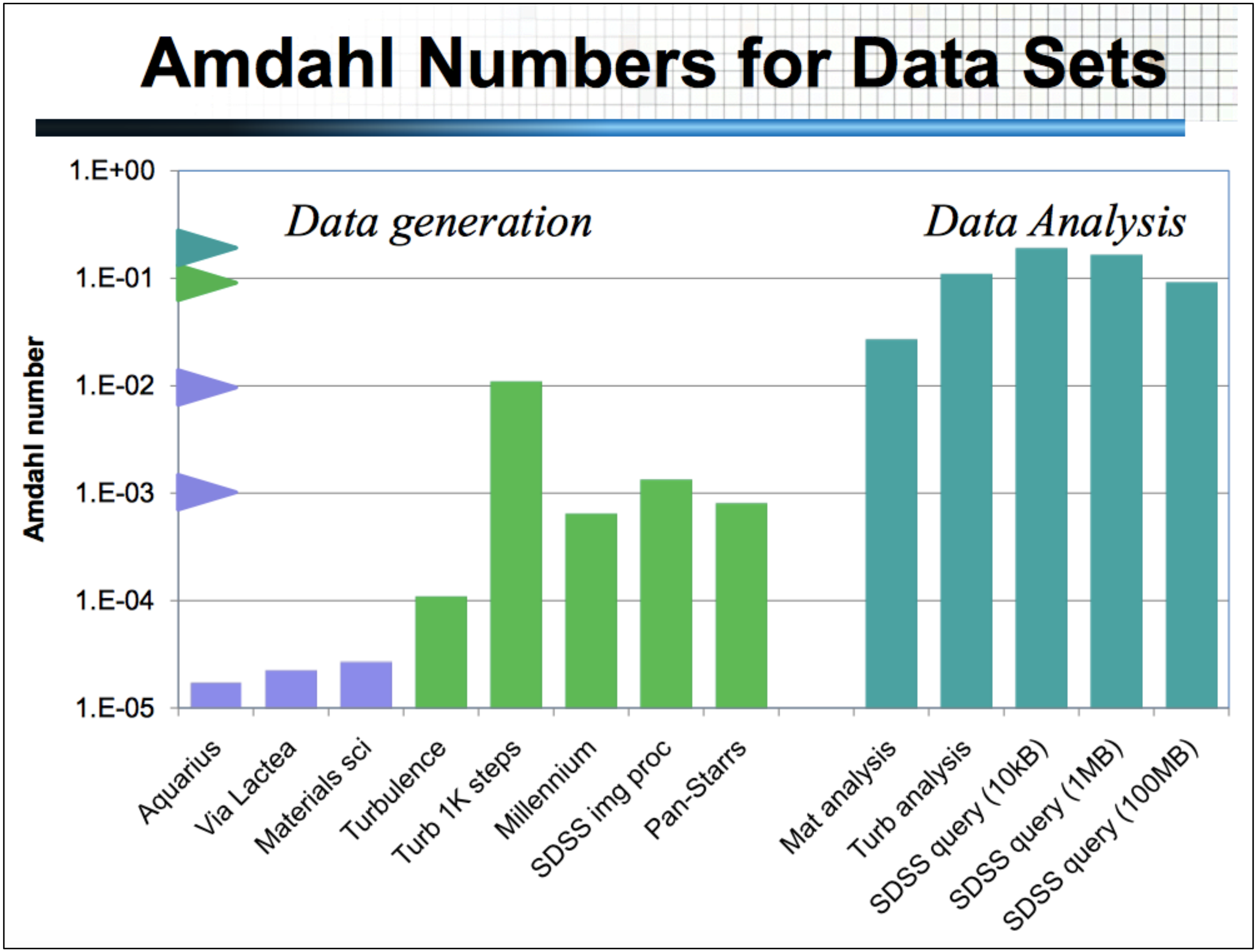
- Third paradigm
  - PDE-based models
  - Particle-based models
  - Linear algebra-based models (e.g., DFT)
  - Image processing
- Fourth paradigm
  - Archiving and retrieving from massive data sets
  - Clustering
  - Searching
  - Knowledge discovery

Application types (2/2)

- Combinations of Third and Fourth paradigms
  - Fourth informs Third
    - Inverse problems
    - Data assimilation
    - Visualization and computational steering
  - Third informs Fourth
    - Design of experiments
  - Both paradigms in a virtuous loop

How to distinguish?

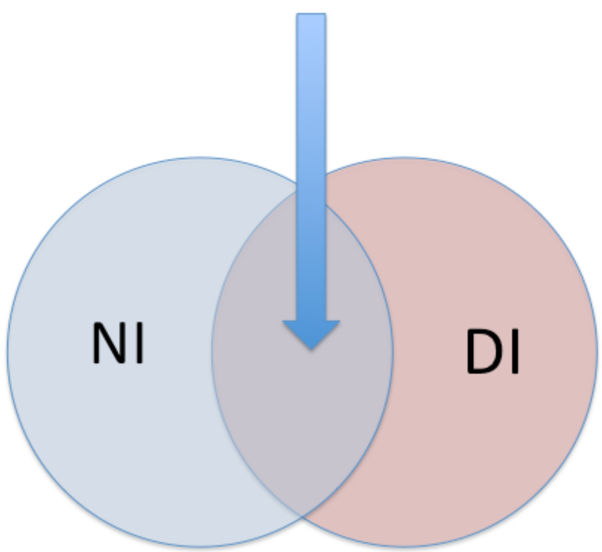
- "Amdahl number" (Bell, Gray & Szalay, 2006)
  - Ratio of bits by CPU cycle/s: O(10^-3) for NI and O(1) DI
  - See this SC'15 Sidney Fernbach Award talk by Alex Szalay
- Data access pattern is also a useful distinguishing feature (see below)



From A. Szalay, Johns Hopkins University)

Main differences and commonalities (1/3)

- There is a growing overlap between NI and DI in applications. Many NI applications produce Big Data and DI is a growing consumer of HPC capabilities.
- There is a growing opportunity for common approaches, common software frameworks, and even reuse of data structures



Significant overlap in the spatio-temporal data realm

Main differences and commonalities (2/3)

- NI has always been oriented towards causality (dynamics, hypothesis-driven), whereas DI has traditionally been oriented towards archiving data for future discovery
  - "The most important thing about Big Data is that you can go back to it." (Amazon philosophy)
- NI data, arising from continuous models on meshes or swarms, tends to be highly structured and highly correlated, which may not be true of DI data
- NI data tends to do lots of writes, whereas DI tends to read many times and write few times

Main differences and commonalities (3/3)

- I/O - Storage:
  - DI has a tendency to use machines with larger memory spaces and I/O infrastructure more oriented towards random access
  - NI usage is dominated by bulk streaming writes
  - Still, a growing degree of parallelism with asynchronous, distributed computation is a trend in applications from both sides
- In NI applications the main action occurs in *memory* and I/O tends to be *regular*, whereas in DI the I/O stream is much more *complex*

Distinctions from a LANL workshop

Axis	Sub-axis	Numerically Intensive	Data Intensive
Hardware	Nodes and Interconnect	High performance and power	Lower performance and power
	Storage	Separate, independent	Integrated
Software	Synchronization	Tightly coupled	Loosely coupled
	Reliability	Checkpoint restart	Replication
Workload	Number of Users	Single per node	Multiple per node
	Data	Dynamic, heterogeneous (unstructured grid)	Static, homogeneous (text, images)
	Algorithms	Global	Distributed
	User Interface	Complex Application	Simple Web
	Data Model	Files	Database
Workflow	Scheduling	Batch	Interactive
	Analysis	Offline post-processing	Online
	I/O	Bulk parallel writes	Streaming writes

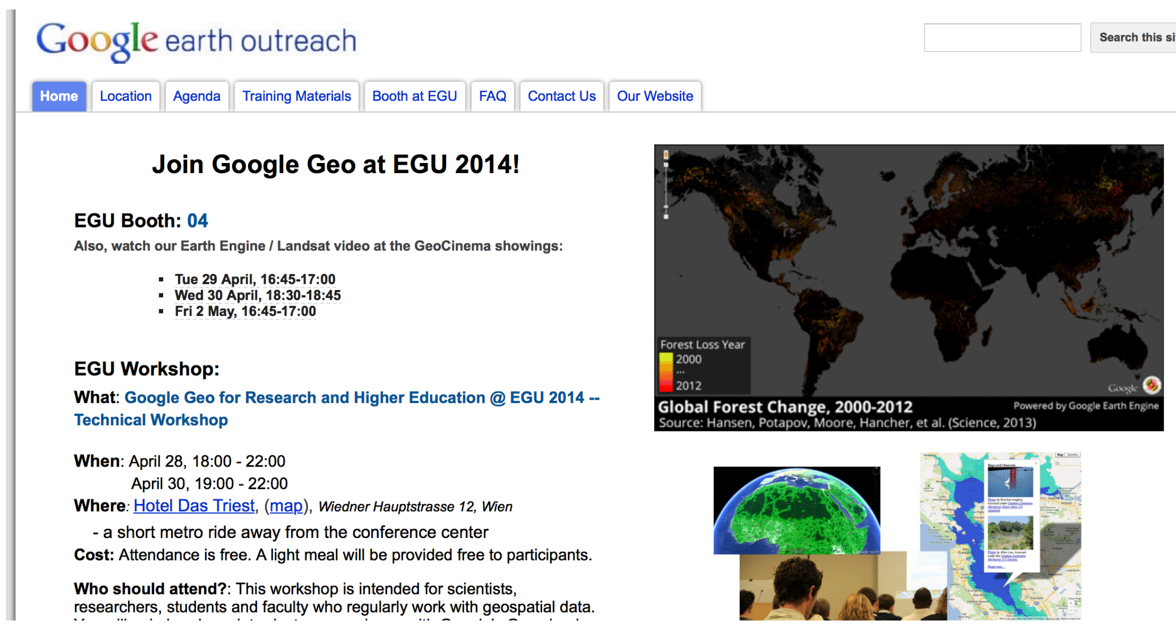
c/o J. Ahrens, 2012

Common needs/problems/interfaces that could serve for application convergence

- To some degree, convergence has been dictated by the *hardware* trends and what vendors make available
  - Less convergence in *software*
- Under all hardware scenarios, data movement is becoming relatively increasingly expensive and analytics should be computed *in situ,* or as close as possible to the data source
  - This may dictate changes in the store vs. recompute spectrum

Testbeds for convergence

- Joint initiative between academia and industry to investigate applicability of high end "Google-type" platforms to scientific, medical, and engineering problems
  - e.g., Google Earth collaboration with European Geoscience Users
  - https://sites.google.com/a/earthoutreach.org/google-egu-2014/



Game changers (1/2)

- Technologies with potential of enhancing *massive data processing* while in transit and therefore enable many of the computations needed to extract knowledge in *both domains*
  - embedded processing components
  - 3D memory technology
  - out-of-core memory algorithms exploiting burst buffers

Game changers (2/2)

- No clear answers
  - Systematic assessment needed to determine what are the *actual requirements* of users/scientists
  - Need to investigate models that *get away from classical batch processing* and take more into account how the information is rendered accessible to decision makers

Highest priorities for convergence (1/4)

- Conferences and workshops that foster a unification of the research communities
  - Now highly fragmented, as evidenced by several presentations at the workshop
- Friendlier allocation processes for major facilities for data intensive workflows
  - Currently adapted to the needs of a typical simulation user (e.g., batch, temporary use of storage)
  - Allow the persistent storage of relevant data bases
  - See new provisioning policy for Cori (NERSC-8), as described by Kathy Yelick in the SC'15 Kennedy Award talk

Highest priorities for convergence (2/4)

- Open data repositories
- Accessible data “grand challenges” that require the unification of the techniques
  - Does the data analytics community fully utilize optimal algorithms from simulation-intensive workloads?
    - Sparse grids
    - Hierarchical matrices
  - Does the simulation community address challenges in data analytics related to its enabling technology successes?
    - Non-negative matrix factorizations

Highest priorities for convergence (3/4)

- Algorithmic research to achieve scaling of parallel techniques on the Big Data front
- New hierarchical representations of data
  - Instead of storing all data at finest resolution store representative sets at finest resolution, along with recursively space and time coarsened sets, and statistics.
- New data structures
  - e.g., based on space filling curves which may be efficient in accessing data at adaptively increasing scales
  - Coarser scales useful for transmission and low-resolution visualization
  - Finer scales useful for analysis once the phenomena of interest are identified.

Highest priorities for convergence (4/4)

- Requirements on the SW/HW environment
  - Portability
    - And performance portability
  - Usability
  - *In situ* use of the data
    - On the fly extraction
  - Means of tracking provenance through data transformations

Conclusions

- We should not force a "shotgun" marriage of "convergence"
  - When a love-based marriage is inevitable in the near future
- The ultimate test of our efforts will be whether the convergence of NI+DI applications meets *mission-critical needs in scientific discovery and engineering design*

Thanks!