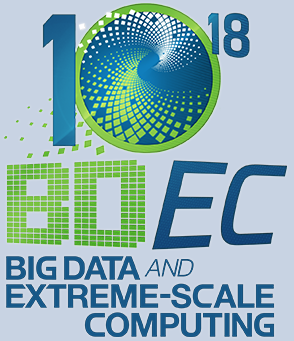
Big Data and Extreme Computing Workshop
Architecture and Operation



Architecture and Operations Working Group Participants

Bill Kramer, Ewa Deelman, Francois Bodin, Piyush Mehrotra, MarieChristine Sawley, Giovanni Erbacci, Yutaka Ishikawa, Toshio Endo, JeanFrancois Lavignon, Pete Beckman, Osman Unsal, Jamie Kinney, Bronis de Supinski, Masaaki Kondo, Marek Michalewicz, Malcolm Muggeridge





Current State

In many ways, BD and EC have co-existed in the same facilities if not the same systems for many decades

Current Petascale EC resources have many examples of processing BD applications well

- K machine is #1 of the Graph500 list
- LSST pipeline has been shown to run well and efficiently on Blue Waters
- Examples of BD frameworks on EC file systems
    - E.g - *Benchmarking and Performance Studies of Mapreduce, Hadoop Framework on Blue Waters Supercomputer*, Manisha Gajbe, Kalyana Chadalavada, Gregory Bauer, William, Kramer, ABDA'15 International Conference on Advances in Big Data Analytics, July 27-30, 2015, Las Vegas, USA
- More papers and posters this week.
- Yet there are still many gaps for convergences
    - It is not easy to be converged

Differences and commonalities between the EC and BD

Differences

BD uses more commercial software data like Oracle

BD **uses** VMs and style **analysis**

BD used to immediately assigned resources

BD has Issues of data transport across national boundaries

BD uses of shared nodes/**virtual** machines

BD is "Highthroughput"

BD has aggregate I/O more important that individual IOPS

BD typically focuses on Importance of timetosolution rather than raw performance

BD wants Longer term data storage

BD data sometimes/often unstructured

BD often uses Java

BD often communicates between tasks with files, EC with messages

BD is more Integer vs floating point performance

BD has streaming data or data streaming over time

Many EC methods require tightly coupled communication

BD is more read than write, EC is more write than read

Commonalities

EC uses commercial software in industry but not in research

Interactivity is needed for data intensive applications

Can use the basic underlying hardware

EC used to high utilization

Some EC is "capacity"

HEP applications use high throughput computing resources and clouds for their computing

EC has many steps from start to insight

EC and BD use many files

Output of EC has big data problems

Future Convergent Applications

Looking towards applications such as LSST and SKA will be using both HPC and HTC computing, issues of realtime data analysis

Smaller apps like UK 10,000 genome project: pulling data from different databases

CyberGIS also has issues of data integration

There was not much use of Hadoop in EC

Climate modeling, a workflow has TB of write and read data, so Hadoop is useful

Are there common needs/problems/interfaces could serve as the basis (or as stepping stones) along a path to (some reasonable level of) infrastructure and application convergence?

Need to support streaming data and databases that change over time, not to re-compute the previous computations

Architectural Choices

Research commonly beneficial for EC and BD

High capacity, high bandwidth

Processors making use of 3D memory (string matching)

High speed solid state storage (HDD is a bottleneck)

Interconnects speed order magnitude (inter-processors, inter-clusters) differences

Possibly novel data representations

High level abstractions for computation and data

Research to promote convergence

High performance object file systems

Speed in/off chips

Avoiding data movements using active storage

Software defined provisioning and management of resources

Co-existing VM in traditional HPC systems / software and application stack control in HPC

Ease of application validation in different environments

Methods for co-location of computation and data

Benchmark application models, traces

Automated, easier way to express optimal/efficient use of deep memory hierarchies, on- the fly data processing

Efficient graph libraries

Common features of infrastructure and application convergence

Data movement dominates energy cost, human cost in moving data

Relative cost of memory is increasing (vs compute), deeper memory hierarchies, relative ability to store data and pull it from storage is decreasing

Heterogeneous execution environments

Workflows are more complex in the future for both EC and BD applications

- should we share the same facilities providing HPC, HTC, data storage resources, or are we looking at more serviceoriented architectures? (physical convergence or integration?)
  Common API that are energyaware for data access and computer resources benefit both BD and EC

Need better ways of transferring data other than FedEx

General need for data reduction: new algorithms

Datacentric view:

- looking at various filesystems,
- what new filesystems we would develop?
- the I/O infrastructure needs to converge
- are there commonalities between big data and HPC computing: visualization, processing of simulation data just like you would process instrumental data?
- consideration of long term data storage for both HPC and BD
- flexible resiliency and consistency models (depending on user's and application's needs)

Need a way to specify the mix of resources that you need and the system would allocate them

Need for a malleable scheduler

Need for virtualization for both BD and HPC, this is where scheduler come in as well

Need to figure out network topologies of HPC interconnect that support BD access patterns

Are there interdomain testbeds that combine BDA and HPC workflows

Possible Drivers: Climate modeling, cosmological modeling requires a number of different resources, Smart Cities

Testbed characteristics:

- Systems with mix nodes and cluster types that can be used in a coordinated manner
- Scheduling capabilities to access and schedule different types of nodes
- Interactive use of resources from desktop, batch jobs
- Potentially different network topologies for BD and EC
- Methods that allow to tradeoff storing data vs recomputing data, methods to calculate the tradeoff
- Monitoring tools and Performance, energy, etc models for applications
- With BD, there is a loose requirements about nodes being up and how does that affect the communication fabrics, so testbeds would support repairs on the nodes vs communication fabric (different modes of repairing)
- largescale tightly coupled resources for users, (if a node failed within a resource, the whole resource will be taken offline)
- a hierarchical system view, with some subsystems would be tightly coupled and some would be loosely coupled (different system composition), scheduling of those
  example: Amazon, how to use spot resources, benchmarking of different instances
- Scheduling of resources extended to the network and i/o fabric, storage the kinds of resources are changing, worries about QoS
- We need to allow endusers to describe their needs in a more comprehensive way
- When faults occur, need to go up the software stack back to the workload management system/application

Technology or new research game changer?

Research that benefit both BD and EC, without need for convergence

- Processors that make use of 3D memory
- highcapacity, highbandwidth, cheap memory (what users want to see is a flat memory)
- highspeed interconnect (LAN and wide area)
- highspeed storage (bottleneck now is disk)
- highperformance file systems
- novel representations of data (floating point)

Research to promote convergence:

- Resoruce management SW for all resources
- speeds in/off the chip
- dataaware algorithms
- minimizing data movement, active storage
- better use of SDN, virtualization
- ability of software defined provisioning and management of resources
- coexisting of VMs in traditional HPC systems/ Software and application stack control in EC
- ease of validating applications in different environments
- methods for colocation of computation and data
- automated, optimal and easy use of deep memory hierarchies, going beyond the cluster level
- onthefly data processing (percipient storage)
- highlevel abstractions for computation and data
- benchmarks, application models, execution traces
- efficient graph processing libraries

Executive summary

General:

- We need to identify different levels of convergence and evaluate their benefits.
- We need to conduct a cost benefit analysis to determine where and how convergence would benefit the user community and how to best prioritize the activities in a way that reflects the needs of the user community and the priorities of the funding organizations.
- Address transnational policies to encourage collaborations and flexible, efficient resources allocations. Exploit new synergies between countries and organizations.

Architecture:

- More robust and dynamic methods to move the data where they are needed
- Ensure I/O and storage technology research and productization targeting need of convergent system receive sufficient focus and funding

Operations:

- We have a clear need for convergence of resource allocation and management mechanisms and services (that accommodate both styles of applications).
- Set up a repository of reference components and workflow systems useful for HPC and BD applications.
- Encourage resource providers to adopt a usercentric model that includes support for convergent BD/HPC applications.