

BDEC Software Session Report

Leads: *Franck Cappello, Satoshi Matsuoka*

Scribe: *Kate Keahey*



Extreme Computing versus Big Data

- Big Data
 - It's not merely about the size!
 - New interaction patterns with focus on diverse types of data, sophisticated data models, communication and interactivity, community ecosystem baggage, modes of usage, etc.

We started out from different assumptions – then new requirements/opportunities came in – but those assumptions are now hardwired into the design of EC systems



EC vs BD: Application Requirements

Extreme Computing	Big Data
Static/predictable requirements for resources	Volatile/unpredictable requirements for resources
Non-interactive	On-demand/predictable/controlled response time, often interactive
Focus on performance	Focus on “productivity”
Data is private	Data is shared and managed for sharing (e.g., provenance), used collaboratively
Focus on domain-dependent methods	Include a wider range of methods including domain-independent methods e.g., statistical methods



EC vs BD: Models and Systems

Extreme Computing	Big Data
Scheduling optimizes provider concerns, e.g., utilization	Scheduling optimizes user concerns, e.g., response time and a range of other QoS constraints
Resource management focuses on compute	Resource management focuses on multiple resources including compute, memory, storage and network
Application is the focus of adaptation and optimization	Platform fit is automatic and the focus of optimization
Assume a static resource model optimizing to fixed assumptions	Assume a dynamic resource model: programming models that adapt to platform change
Fault tolerance is expensive	Redundancy-based fault tolerance

Stepping Stones: Towards EC/BD convergence

- Sharing the same resources
 - Resource management methods need to evolve so that BD and EC can share resources
 - Dynamic, QoS-aware, multi-aspect, leverage trends towards resource programmability, incentives
 - Knowing and communicating more about the application is key
- Programming models
 - Exploration of more flexible/malleable programming models
 - Resilience: models intolerant of failure versus models based on redundancy
- Security: from coarse-grained security (firewalls) for EC to fine-grained for BD



Domain Testbeds

- Experimental CS Testbeds
 - Chameleon, CloudLab, Grid'5000, FED4FiRE
- Resources exploring the BD management models
 - Comet, JetStream, Bridges, Wrangler
- Industry
 - Amazon, Google
- How about supercomputers?
 - Which ones are amenable for running convergent workload?



Game Changers

- Convergence towards “stepping stones”
- Challenges and demonstrations: software representing an entire system that can be used for BDEC
- Convergence
 - HPC features available in the cloud (HPC)
 - “Cloud” features available on HPC platforms (availability, predictable response time, etc.)
- End of Moore’s Law is a catalyst ;-)



Towards Convergence: Proposed Actions

- Grand Challenges and Success Stories
 - It has been noted that we need to disqualify Google ;-)
- “Throw money on the problem”
 - Research calls focusing on “stepping stones”
 - Mandate as part of procurement
- Workshops, forums, “melting pots” between communities
- Make available representative traces and workloads (chicken and egg problem?)
- Documents: create common vocabulary, architecture
- Solve incentive problems



Success of BDEC

- Documents and methodology plus key problems to drive community research
- Formation of a community
- Formulation of research programs
- Long-term: adoption of convergence technologies

