Algorithms and Libraries

Breakout Summary

Bridging HPC-BD Computing Environment Gaps

- HPC and BD have separate computing environment heritages.
  - Data: R, Python, Hadoop, MAHOUT, MLLIB, SPARK
  - HPC: Fortran, C, C++, BLAS, LAPACK, HSL, PETSc, Trilinos.
- Determine capabilities, requirements (application, system, user), opportunities and gaps for:
  - Leveraging HPC library capabilities in BD (e.g., scalable solvers).
  - Providing algorithms in native BD environments.
  - Providing HPC apps, libraries as appliances (containers aaS).

Refactoring & leveraging of HPC Capabilities for BD

- Sparse computations:
  - HPC: low, consistent degree graphs.
  - BD: highly variable degree, “power law” graphs.
  - Requires different partitioning, parallel strategies.
- Dense LA for some machine learning.
- High performance communication libraries (MPI).
  - Global collectives for machine learning (dense).
  - Point-to-point for graphs.

New Math & Algorithms

- Math & Algorithms for Intrinsically Discrete Data (Ie.g., light sources)
  - Model extraction.
  - Surrogate development.
  - Inverse problems.
  - In general: Converting observations to models.
  - Mature in HPC (e.g., Oil & Gas), but new areas: e.g., sensors.
- Factorizations, spectral algorithms, other NA for tensors.
- Algorithms based on random sampling.
  - Stochastic Gradient Descent algorithms from sampling.
    - Already being done, but reconsider from HPC perspective.
  - Better methods than gradient descent?
- Streaming algorithms, “online” algorithms.
- Complexity reduction: Decrease from n^2 to *n log n* or *n*.
  - Similar to multi-pole expansion, FMM.
- Low-rank representations: e.g. H-matrix approaches.
- General: Revisit BD problems with mindset of “HPC is in your toolbox.”

New Libraries

- HPC-BD libraries are needed.
  - Scalable. Not trivial for many reasons.
  - Support virtual resources (e.g. virtual clusters).
  - Agreed upon abstractions.
    - Graph, KV, pixel ?
    - File formats (HDF5, FITS): Reconcile common data/file formats with big data.
  - Usability, accessibility: “Bring to the BD community”
    - Address multiple situations from long tail to big science.
  - Conceptual software stack.
    - Low-level services to high-level knowledge.

Requirements for other breakouts

- A well defined infrastructure (virtual cluster concept):
  - Important for providing libraries.
  - It's a good model in general.
  - Must be high performance.
- High performance virtual network APIs.
  - Infiniband is fast, need virtual, fast API.
- Programming model & communication layers:
  - Bring together the best of HPC and BD.
  - Examples: MPI+Hadoop/Spark, Load balancing + Giraph/Pregel
- Support for workflow, data fusion.
  - E.g., Drawing from multiple data sources.