BDEC Math, Algorithms and Libraries

Report from Barcelona Meeting

Michael A. Heroux, Sandia National Labs

Alison Kennedy, EPCC

Contributions from many others

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.

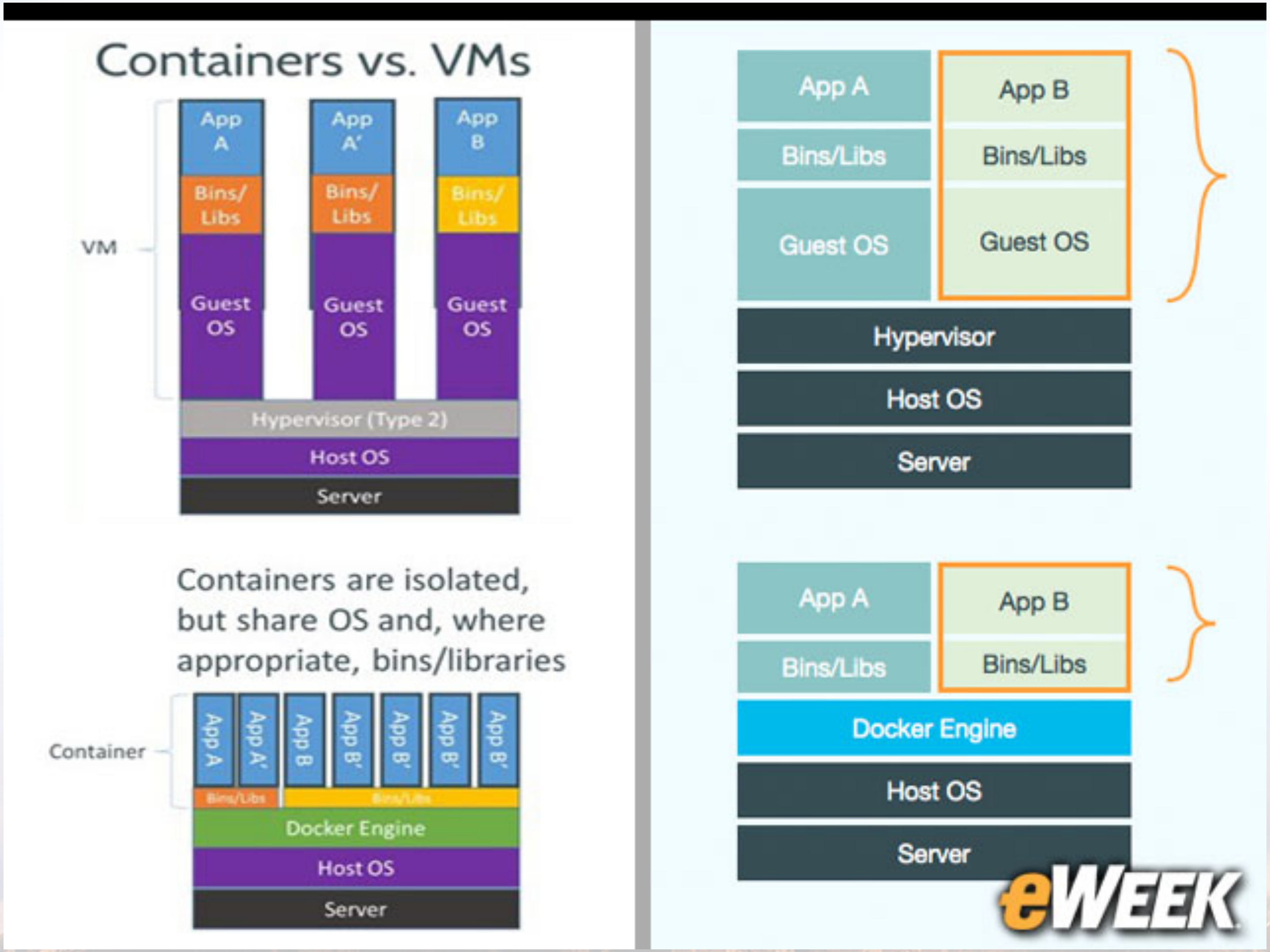
Some Barcelona-inspired work: Trilinos+Docker

- Talk with Jamie Kinney, AWS.
- Student with good tech skills.
- WebTrilinos +Docker.
- Peridigm + Trilinos + Docker.
- Long-term goal:
  - Reduce barrier to access complex software environment.
  - Lots to learn.

Typical Cmake Script (hopper)

```
cmake \
-D MPI_CXX_COMPILER="CC" \
-D MPI_C_COMPILER="cc" \
-D MPI_Fortran_COMPILER="ftn" \
-D Teuchos_ENABLE_STACKTRACE:BOOL=OFF \
-D Teuchos_ENABLE_LONG_LONG_INT:BOOL=ON \
-D Trilinos_ENABLE_Tpetra:BOOL=ON \
-D Tpetra_ENABLE_TESTS:BOOL=ON \
-D Tpetra_ENABLE_EXAMPLES:BOOL=ON \
-D Tpetra_ENABLE_EXPLICIT_INSTANTIATION:BOOL=ON \
-D Teuchos_ENABLE_EXPLICIT_INSTANTIATION:BOOL=ON \
-D TPL_ENABLE_MPI:BOOL=ON \
-D CMAKE_INSTALL_PREFIX:PATH="$HOME/opt/Trilinos/tpetraEval" \
-D BLAS_LIBRARY_DIRS="$LIBSCI_BASE_DIR/gnu/lib" \
-D BLAS_LIBRARY_NAMES="sci" \
-D LAPACK_LIBRARY_DIRS="$LIBSCI_BASE_DIR/gnu/lib" \
-D LAPACK_LIBRARY_NAMES="sci" \
-D CMAKE_CXX_FLAGS="-O3 -ffast-math -funroll-loops" \
\
..
```

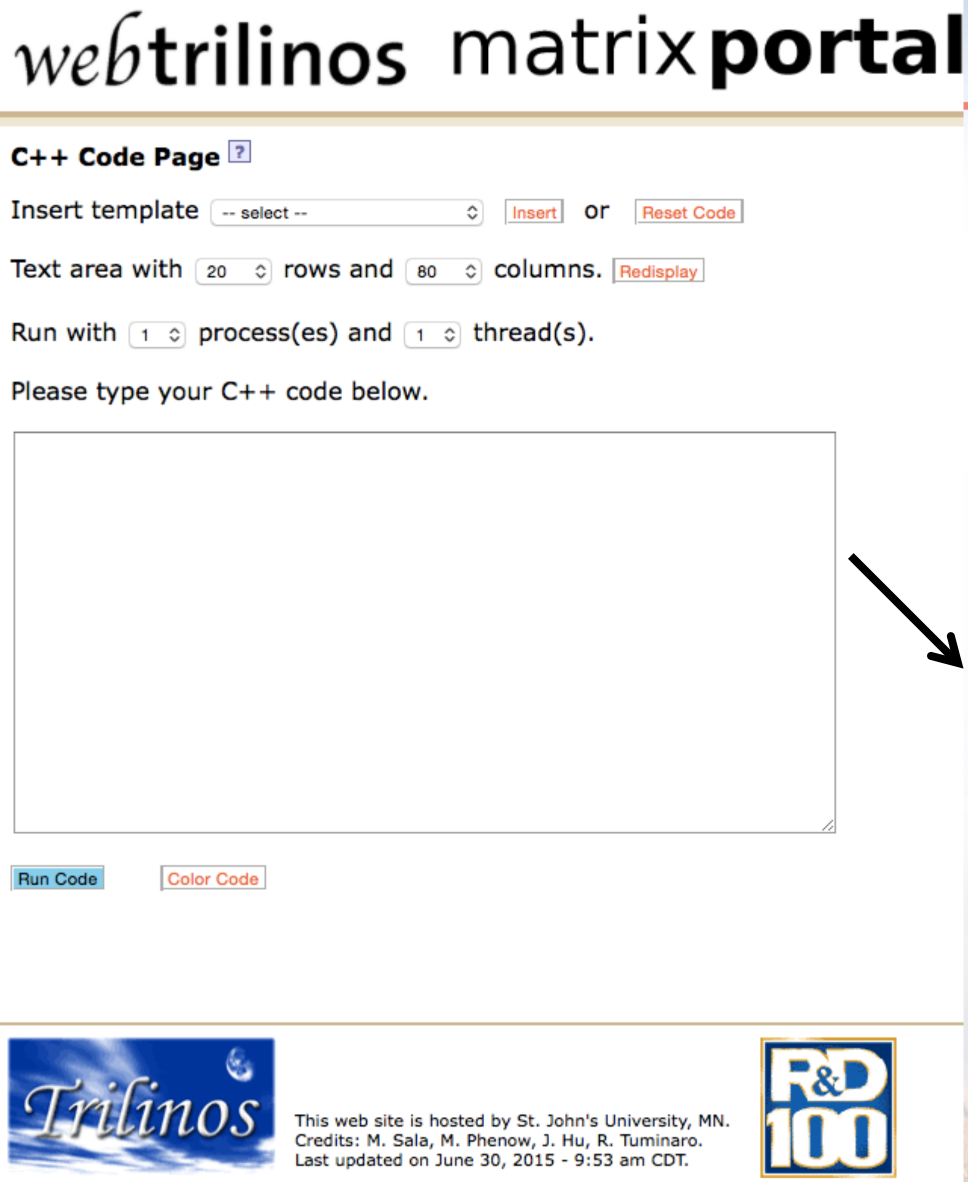
Docker (about which I know little)



Containers vs. VMs

Containers are isolated, but share OS and, where appropriate, bins/libraries

WebTrilinos



webtrilinos matrixportal

C++ Code Page

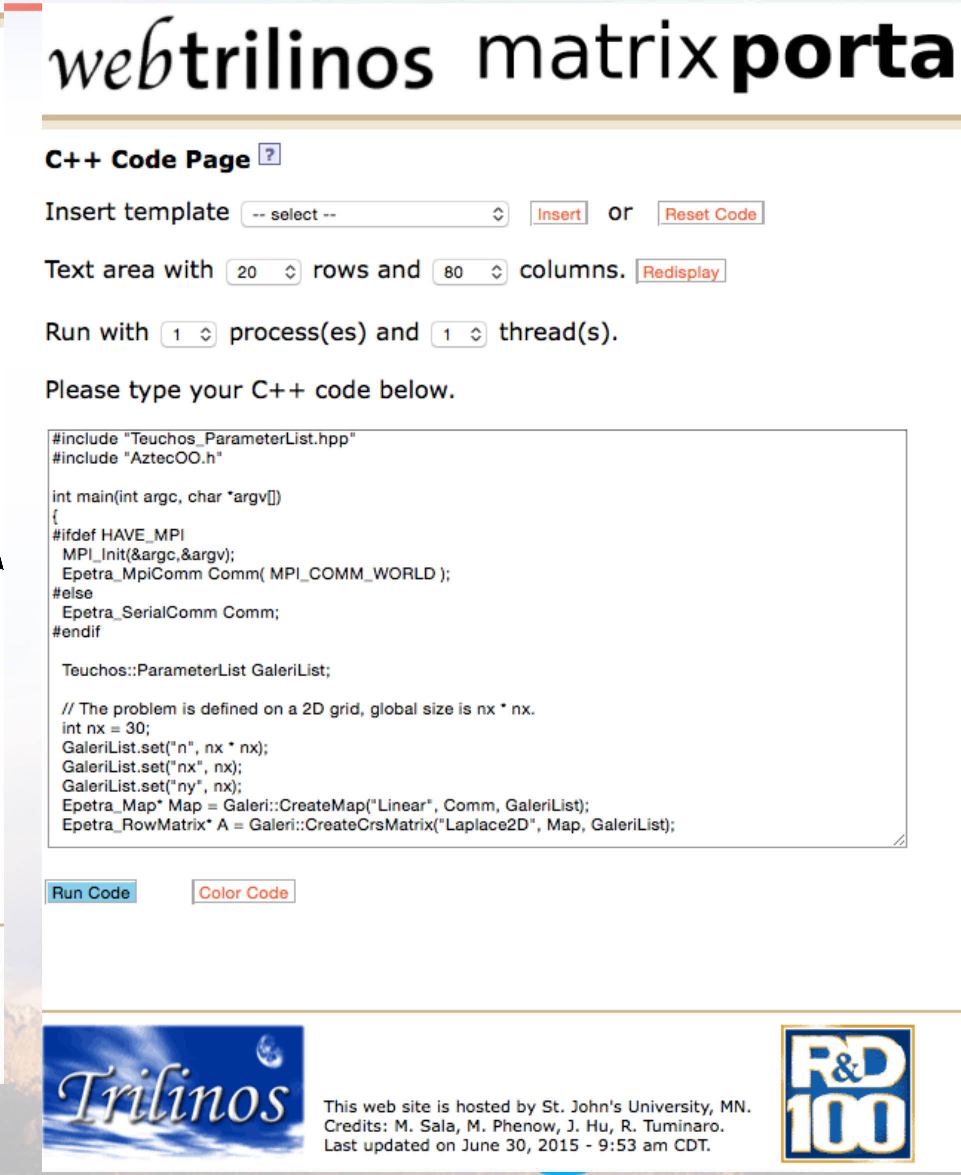
Insert template -- select --

Text area with 20 rows and 80 columns.

Run with 1 process(es) and 1 thread(s).

Please type your C++ code below.

This web site is hosted by St. John's University, MN.
Credits: M. Sala, M. Phenow, J. Hu, R. Tuminaro.
Last updated on June 30, 2015 - 9:53 am CDT.



webtrilinos matrixportal

C++ Code Page

Insert template -- select --

Text area with 20 rows and 80 columns.

Run with 1 process(es) and 1 thread(s).

Please type your C++ code below.

```
#include "Teuchos_ParameterList.hpp"
#include "AztecOO.h"

int main(int argc, char *argv[])
{
#ifdef HAVE_MPI
  MPI_Init(&argc,&argv);
  Epetra_MpiComm Comm( MPI_COMM_WORLD );
#else
  Epetra_SerialComm Comm;
#endif

  Teuchos::ParameterList GaleriList;

  // The problem is defined on a 2D grid, global size is nx * nx.
  int nx = 30;
  GaleriList.set("n", nx * nx);
  GaleriList.set("nx", nx);
  GaleriList.set("ny", nx);
  Epetra_Map* Map = Galeri::CreateMap("Linear", Comm, GaleriList);
  Epetra_RowMatrix* A = Galeri::CreateCrsMatrix("Laplace2D", Map, GaleriList);
```

This web site is hosted by St. John's University, MN.
Credits: M. Sala, M. Phenow, J. Hu, R. Tuminaro.
Last updated on June 30, 2015 - 9:53 am CDT.

Containerized Libraries

- BD seems require container approaches.
- HPC can provide, and can benefit.
- Eliminates complicated build process.
  - The dozens of 3rd party libraries can be pre-built, installed.
- Full-featured installation, custom installations.
  - Example: Trilinos has 57 packages.
  - Most people use “aspects” of Trilinos.
- Portable debugging environments.
- Many other opportunities.
- Lots to be learned…

Peridigm usage via Docker

- http://johntfoster.github.io/posts/peridigm-without-building-via-Docker.html

  - docker pull johntfoster/trilinos
  - docker pull johntfoster/peridigm
  - docker run --name peridigm0 -d -v `pwd`:/output johntfoster/peridigm \
            Peridigm fragmenting_cylinder.peridigm
  - Etc…

One more Barcelona-inspired item: Trilinos Anasazi/RBGEN package

- SVD seems to be one important big data tool.
- Turnkey driver for large-scale truncated SVD:
  - Via eigensolve of A^TA.
  - Using the Anasazi package in Trilinos
    - Developed by R. Lehoucq (ARPACK), H. Thornquist
- Permits abstract definition of *A.*
- Permits sparse definition of *A.*
- Very storage-friendly.
- Goal: Containerized version of RBGEN.

Opportunities & Challenges

- FOAs for NA for BD.
  - Appears to be rich exploration space for expansion to tensors.
- New math models for discrete data sources.
- Leverage of HPC sparse technologies in BD.
- Containerization technologies from BD for BD & HPC.
  - Turnkey of common computations, e.g., SVD.
- A lot of domain knowledge building.
  - BD and HPC conversations.
  - Translation of terminology.
  - Domain model understanding.
- Challenge: Danica Patrick Culture of HPC.