An Open Dynamic Big Data Driven Application System Toolkit: An Update

Craig C. Douglas[1,2,3]

Enrico Fabiano[1], Mookwon Seo[1], and Xiaoban Wu[1]

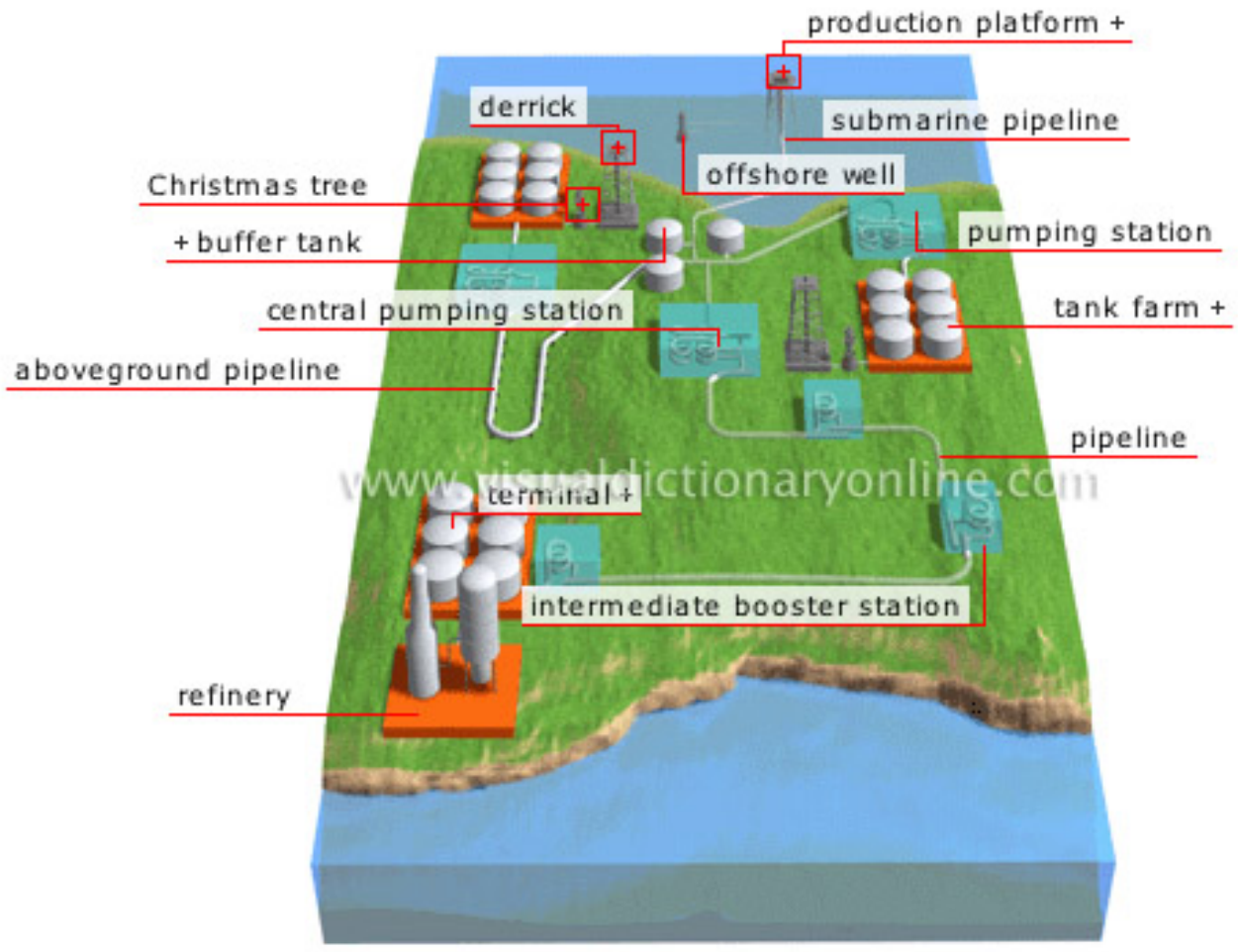
[1] University of Wyoming
[2] KAUST
[3] Kyoto University

This research is supported in part by the National Science Foundation, King Abdullah University of Science & Technology (KAUST), and Kyoto University.

DBDDAS = DDDAS + BD

- A dynamic data-driven application system (DDDAS) is the integration of a simulation with dynamically assimilated data, multiscale modeling, computation, and a two way interaction between the model execution and the data acquisition methods.
  - If named more recently, DDDAS would probably be called *intelligent* (or *smart) data assimilation* and be known as iDA.
- Data assimilation began with satellite data in the 1960's and was truly transformative research.
- All DDDAS have been dealing with Big Data issues as long as there have been DDDAS.
- Combining the two paradigms not only makes sense, but is essential to creating new DBDDAS quickly and efficiently.

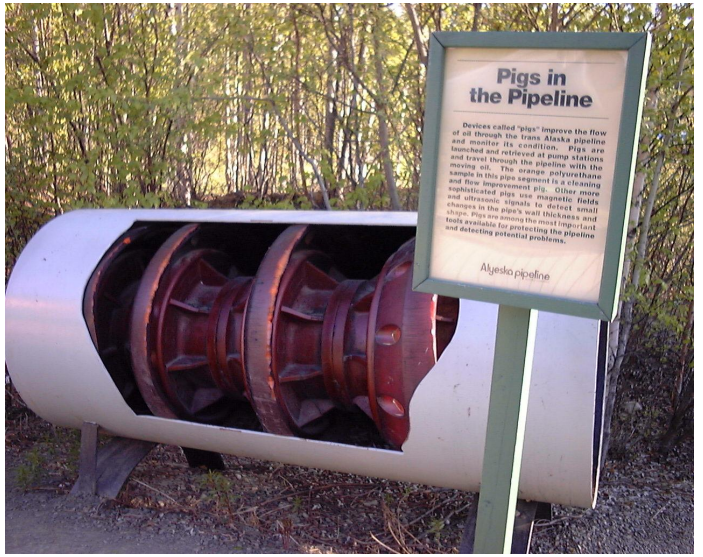
Energy Example: Oil/Gas Pipelines



Picture courtesy of Miriam Webster Dictionary

Pipeline Network Properties

- Pipe diameters range from 2 inches to 5 feet.
- Rarely straight and level.
- Contain
  - Possibly different grades of oil or gas simultaneously.
  - Pigs as separators.
  - Sensors (inside and outside)
- Not restricted to oil/gas (water, chemicals, etc.).



1970's Style DBDDAS

- Commercial product with international sales from 1974.
- Problem modeled mathematically based on time dependent, nonlinear coupled partial differential equations (two models).
- Sensors on all pipeline components (recall the cartoon).
- *Two way data/model control*.
- Distributed GRID computing with scattered phone booths:
  - 2 minicomputers, 4 array processors, a heat pump on top.
  - Sensors provided data (temperature, pressure, and velocity) dynamically based on need and anomalies and controlled by the environment and running model.
  - No central computing, just central and distributed control sites.
  - 2,000 pieces of telemetry/minute in complete KSA oil pipeline network (1978) using an ad-hoc phone line data network.

Current Pipeline DBDDAS

- Central cluster computing.
- Fiber optic TCP/IP with WiFi and Gigabit Ethernet near pipelines.
- Many smart sensors including new ones to measure pipe shape changes, internal pollutants, and external gas leakage.
- 3D math models of pipelines with topography.
- When 1978 system replaced in KSA in 1998, 100,000 times the telemetry/minute.
- In 2015, a tsunami of data.

Monitoring Site

- In 1970's:
  - Primitive center where *what if* scenarios were run to keep pipelines from breaking in parallel with regular monitoring.
  - Testing to verify math model was actually good to about 100 feet for predicting pipe events.
- Now:
  - Large scale visualization is used to monitor pipelines in a multiscale framework.
  - Individual high resolution monitors (1080p and 4K) used for *what if* scenarios.
  - Still trying to find anomalies in the data streams to avoid pipeline problems.

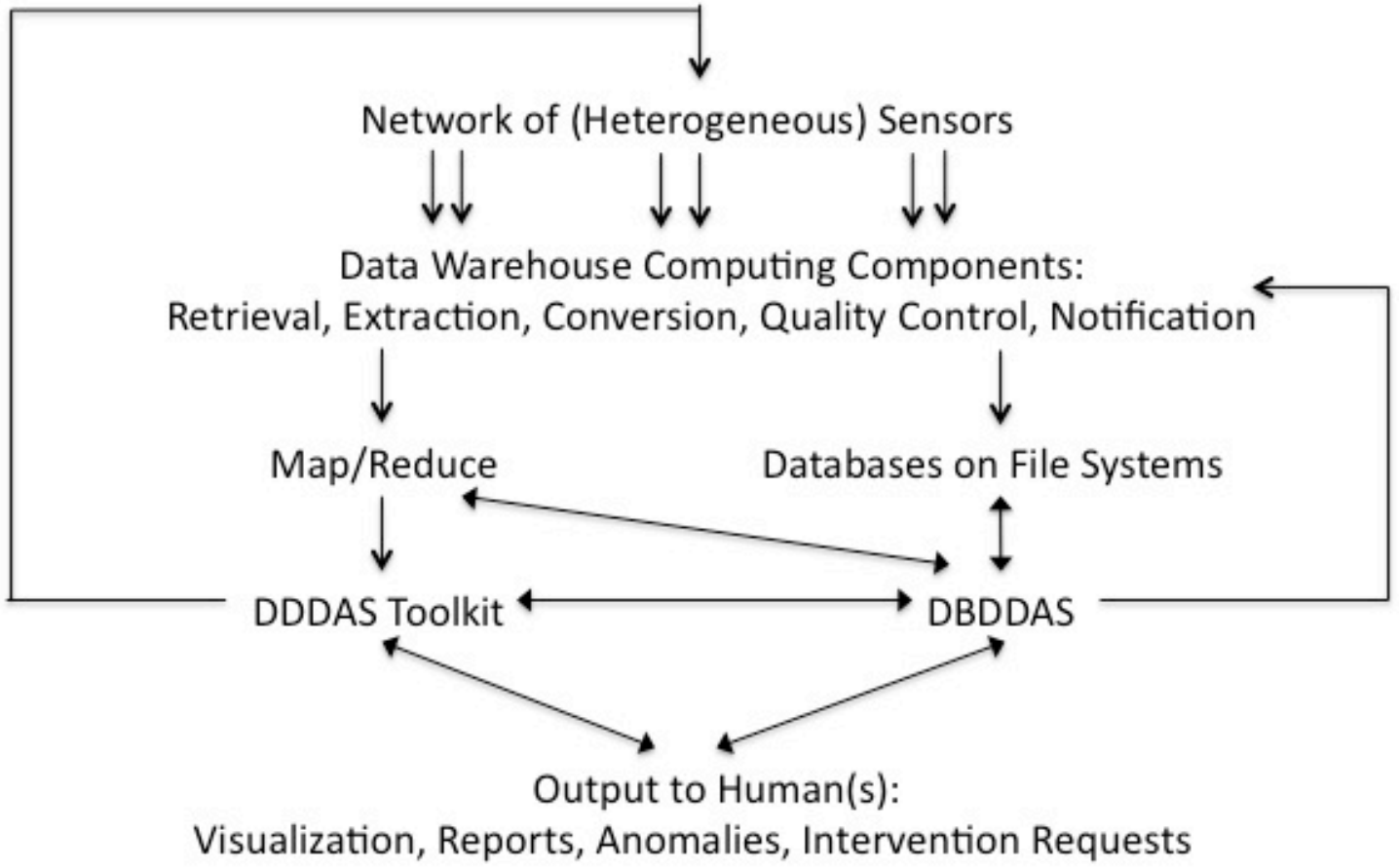
OpenDBDDAS Framework

- Much of the framework exists, but is not tied together.
- An open source sensor language standard that stays open.
- Cache aware, fast hash tables (not textbook variety).
- High performance disk I/O assuming there is already a (distributed) file system.
- SQL and NoSQL databases.
- Data warehouse with six point functionality
  - Retrieval
  - Extraction
  - Conversion
  - Quality control
  - Store
  - Notification
  - Intelligent sensing and processing (ISP) capable.

OpenDBDDAS Framework

- DDDAS toolkit that allows
  - Uncertainty quantification and statistical analysis.
  - Numerical solvers for standard (nonlinear time dependent coupled) PDEs.
  - Optimization solvers.
  - Data assimilation using multiscale interpolation methods.
  - Item identification, tracking, and steering.
  - Anomaly tracking and verification.
  - Human notification methods.
- An open source MapReduce suitable for medical records
  - Must be HIPPA compliant.
  - Must scale well for very large databases.
  - Must have individual/group access capabilities (ala Linux files) for records.
  - Must not have complexity O(disk access) on a DFS.
    - Should use OpenMP and MPI.
    - Should use (disk) cache aware hashing methods.
  - *Will be useful well beyond medical records*.
- *Usable by non-computer scientists*

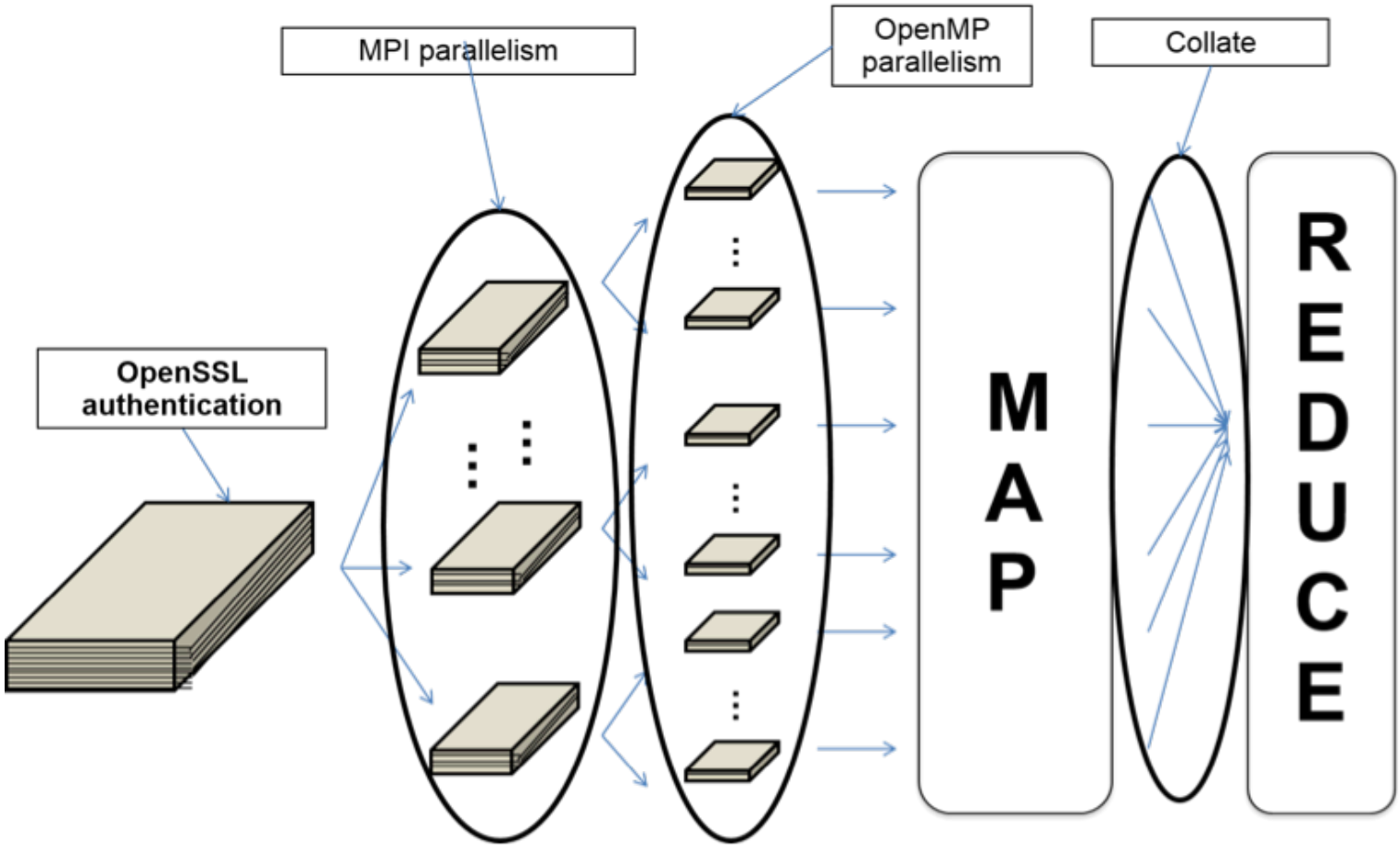
Typical Use of OpenDBDDAS Framework



Why a Secure MapReduce?

- In U.S., there is a federal law, HIPPA (or Health Insurance Personal Privacy Act), a 900+ page law.
  - Most hospitals in U.S. use Hadoop as a front end to their NoSQLs.
  - Hadoop violates HIPPA.
  - Hospitals put on notice in June, 2013 and fined since January 2014.
  - Very easy to see in action in small towns with one hospital used by town's doctors.
- HIPPA has straightforward requirements:
  - New users granted access via third party e.g., (sysadmin). We use a hidden *user file*.
  - All data hidden and expunged when use completed.
  - All data encrypted while on an electronic storage device.
  - Third party can resurrect all data should disaster occur (a failsafe system).
  - All encryption keys are known by the third party.

UWMR MapReduce Workflow

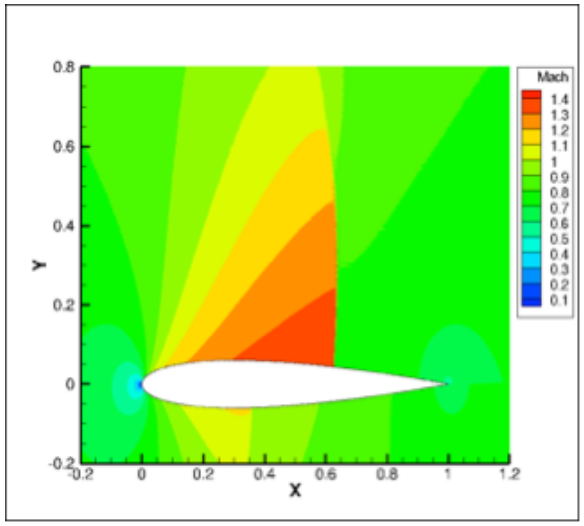


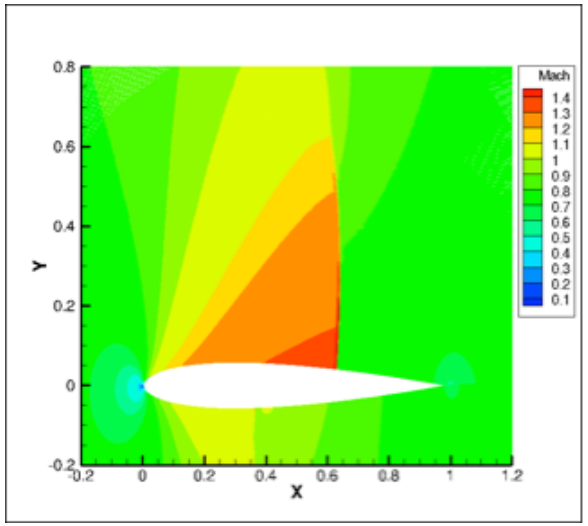
UWMR MapReduce Properties

- High performance:
  - MPI between nodes.
  - OpenMP inside a node.
  - Does not run at O(DFS) read/write speeds (avoids disk I/O).
  - Distributes files in use between processes.
- Secure:
  - User can only access authorized files or parts of files.
  - MapReduce data transmission secure.
  - Uses a NSA approved encryption package.
  - One user cannot access another user's data.

Simple Tests

- Contour plotting for Mach number contours for the transonic flow around the NACA 0012 airfoil.
  - Choose number of contours and estimates of max/min of velocities.
  - Compared with Tecplot.





Simple Tests

- Traditional Hadoop word identification
  - UWMR much faster than Hadoop due to not using disk I/O for message passing
  - UWMR faster than MR-MPI (Sandia) for a small number of processes
  - Currently does not scale as well as MR-MPI.
    - Next version will.
- Further improvements clearly indicated:
  - Someone who understands parallel communication and system software is rewriting UWMR.
  - A much faster hashing scheme (ala Google and Yahoo!) will speed UWMR up.

Final Comments

- An open comprehensive toolkit would help the DDDAS community compete with several CyberPhysical Systems libraries (e.g., iOS, Android, and Windows).
  - *Or* admit that CPS has won over the DDDAS, DISD, DA concepts.
- A Sourceforge project *group* is needed to provide a wider definition of what should be included in the toolkit and to see that it works well.