A Case for High Performance MVAPICH2 for Machine Learning on Extreme Scale (MaTEx)

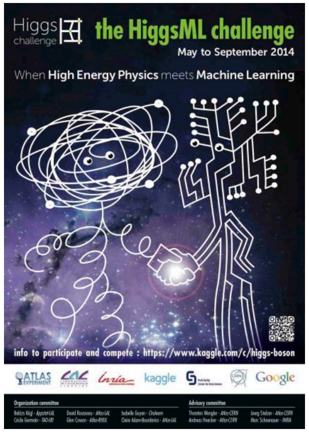
ABHINAV VISHNU

MUG' 15

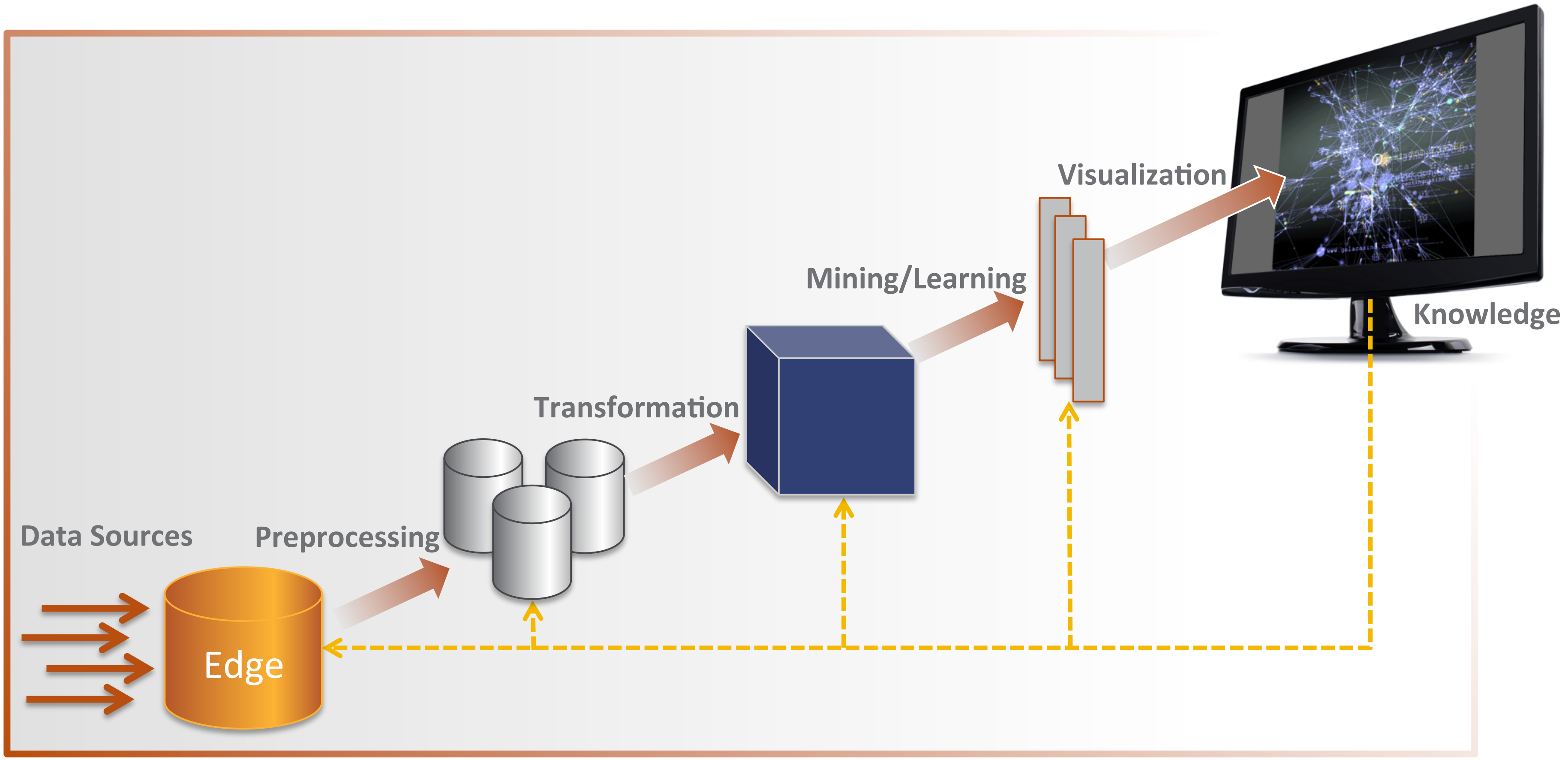
August 19-21, 2015

A Case for Machine Learning in Science and High Performance Computing

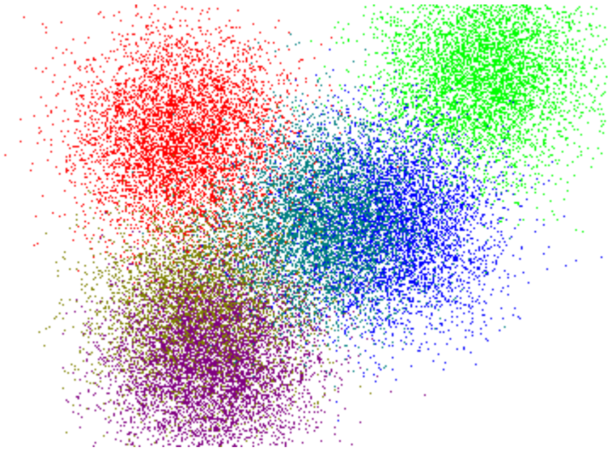
- Data Driven Science can lead to better discovery
  - Higgs ML challenge
- "Searching for Exotic Particles Using Deep Learning", Nature Communications, 2014
- "Machine Learning and its applications to Biology", PLOS, 2007
- Solve HPC problems
  - Performance Modeling
  - Resilience
  - Energy Modeling and Optimization



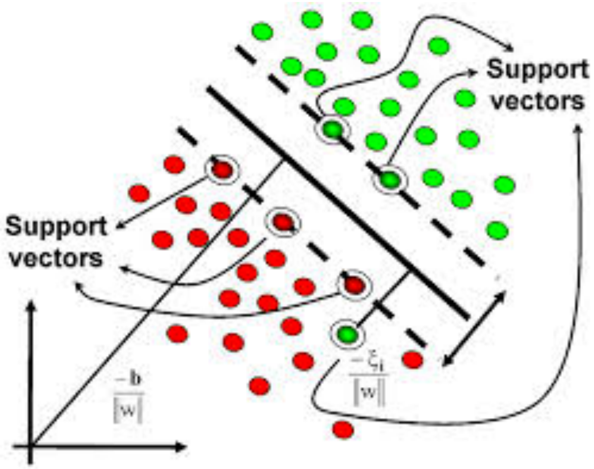
Elements of Data Driven Science



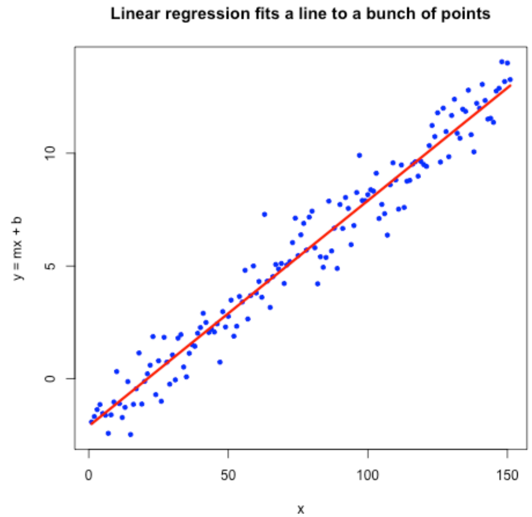
Typical Machine Learning Tasks



Clustering- Unsupervised Learning

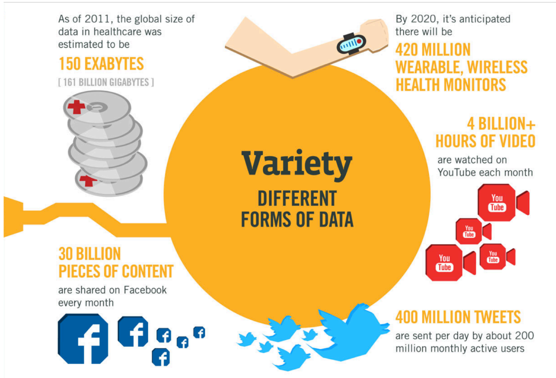


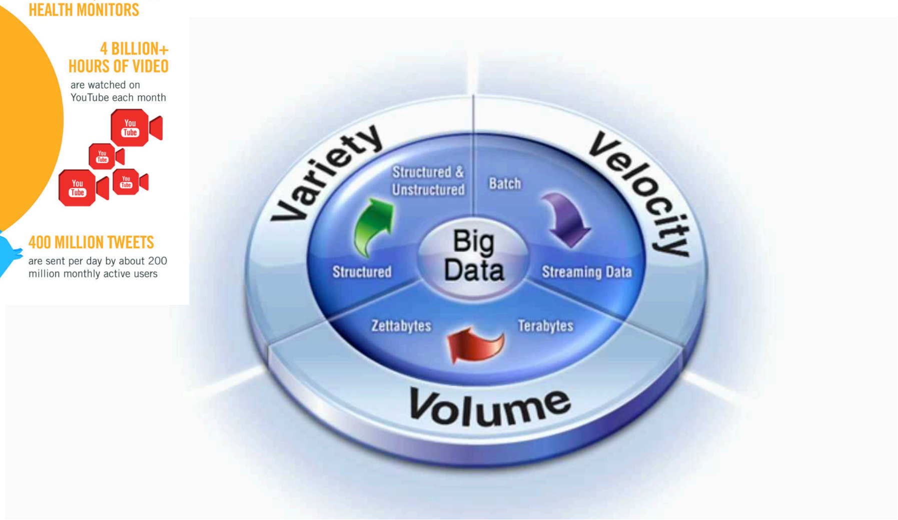
Classification – Supervised Learning



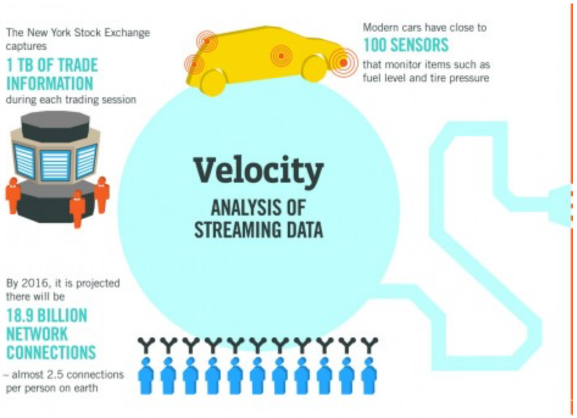
Regression – Supervised Learning

The V's in Data Driven Science

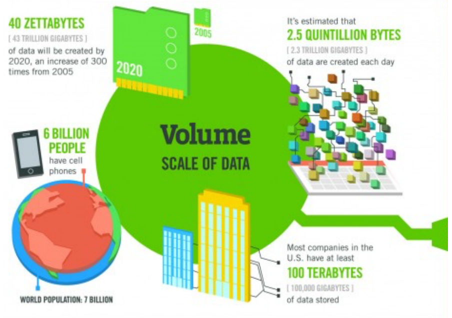




CERN produces 20 GB/s,
ARM produces 1 TB/day
HPC Performance Counters



CERN > 100 PB overall
ARM > 1 PB overall
HPC Performance Counters



Limitations of Existing ML Libraries

Spark MLlib

mahout

Non-native compute and communication primitives

Generally, Limited to single node Systems

Berkeley

NVIDIA

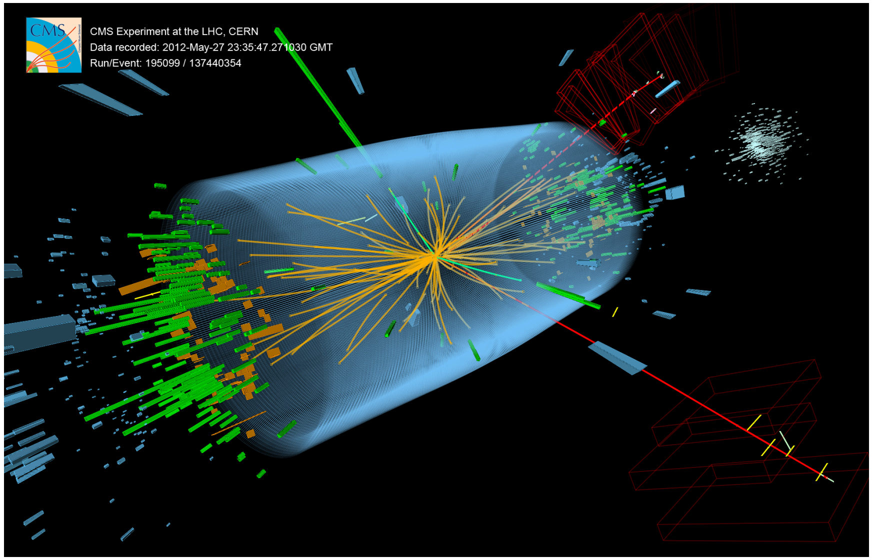
torch

What are the benefits of designing native compute ML algorithms?
What is the role of interconnects (and MPI) in Machine Learning?

MaTEx – Machine Learning Toolkit on Extreme Scale

- MaTEx is a library of ML and DM algorithms using MPI + X
  - Supervised: SVM, KNN
  - Unsupervised: Spectral, Kmeans
  - Frequent Pattern Mining (FPM): FP-Growth
- Specifically geared for supercomputers
  - Readily applicable to cloud systems
  - Leverage MPI portability
- Supports parallel incremental Support Vector Machine algorithm
  - Use case: Streaming data in several science and other domains
- 0.1 released earlier
  - 0.2-alpha currently released
- Libsvm-openmp is planned to be included
  - Enhances the highly popular libsvm to use multi-core systems
- URL: http://hpc.pnl.gov/matex

MaTEx SVM on Higgs Boson Dataset



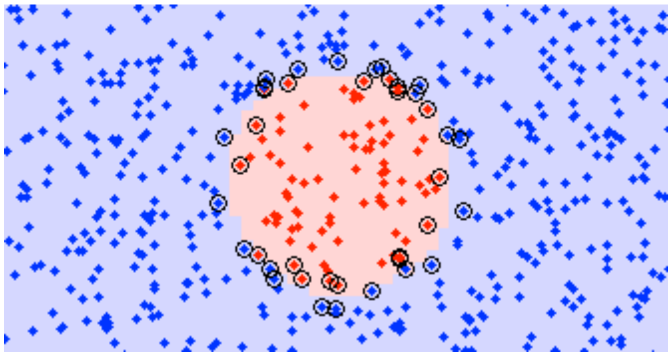
- Objective
  - To learn the model on the UCI Higgs Boson dataset
  - Classify Signal from Noise
  - 11 M Samples
  - MaTEx Support Vector Machines
    - Sequential Minimal Optimization (SMO) algorithm
    - Most widely used algorithm

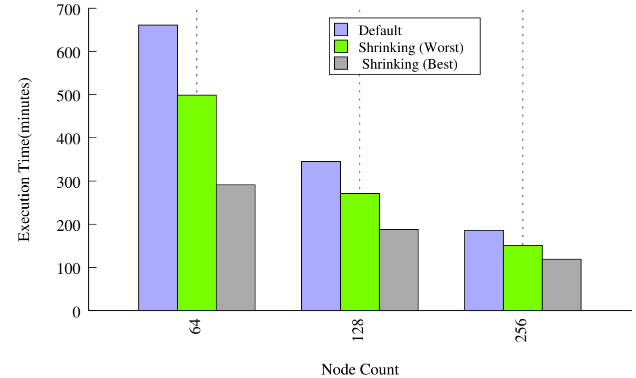
- We used **6144** cores for 2 days on PIC Constance using MVAPICH2
- We could only complete two experiments
- The objective is to improve the model from the published paper
  - We needed more experiments
- Larger scale (and time) access is critical!

Argonne Director's Discretionary Award

- Applied for ALCF Director's discretionary time award
  - **Scalable Machine Learning on Blue Gene/Q supercomputer**
- Granted 2M core hours!
- Initial results are encouraging
- Expecting to write INCITE for next year
  - Requested extension for the rest of the year

Can we improve model time without reducing accuracy?



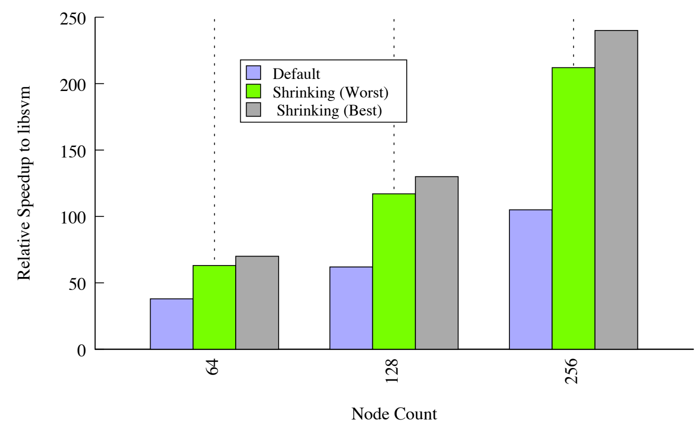


Observation1: A small fraction of Samples define the boundary

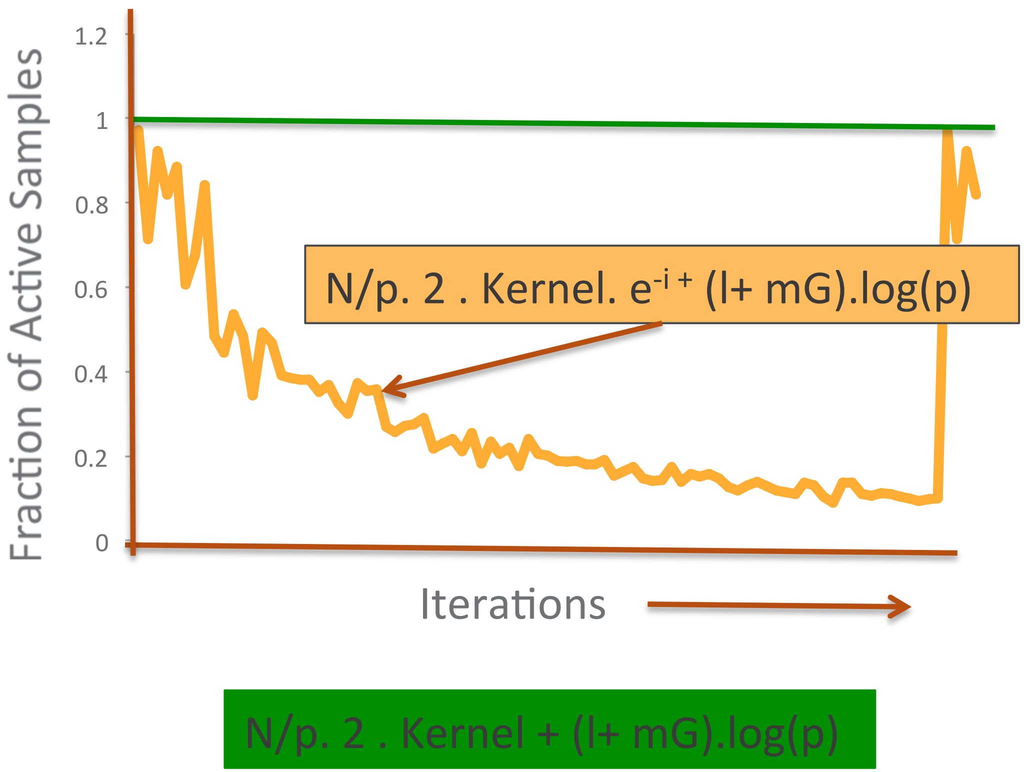
Observation2: No parallel algorithm Considers adaptive elimination (shrinking)

Observation3: Reducing accuracy at cost Of time improvement is not attractive (as Opposed to CA-SVM (IPDPS'15 – You et al.))

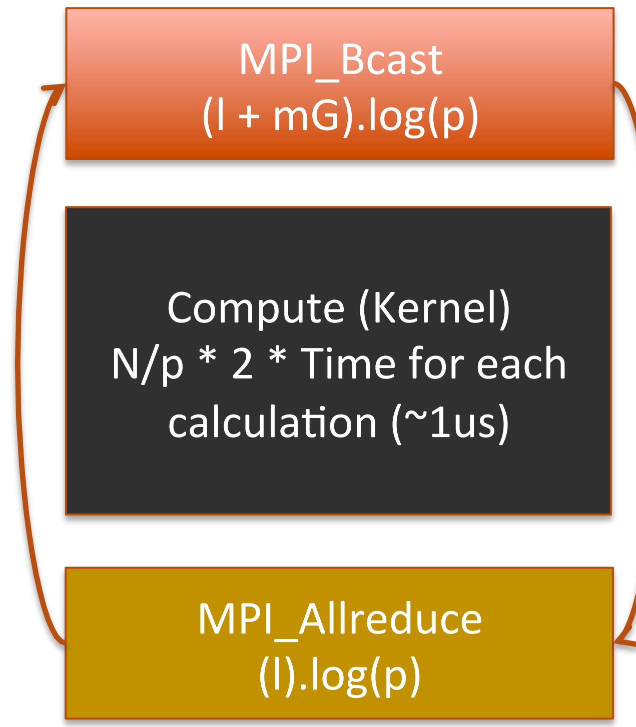
Enhance libsvm (default SVM library) To use OpenMP for baseline comparison --- planned to be released with MaTEx



Performance Modeling of MaTEx SVM Algorithm



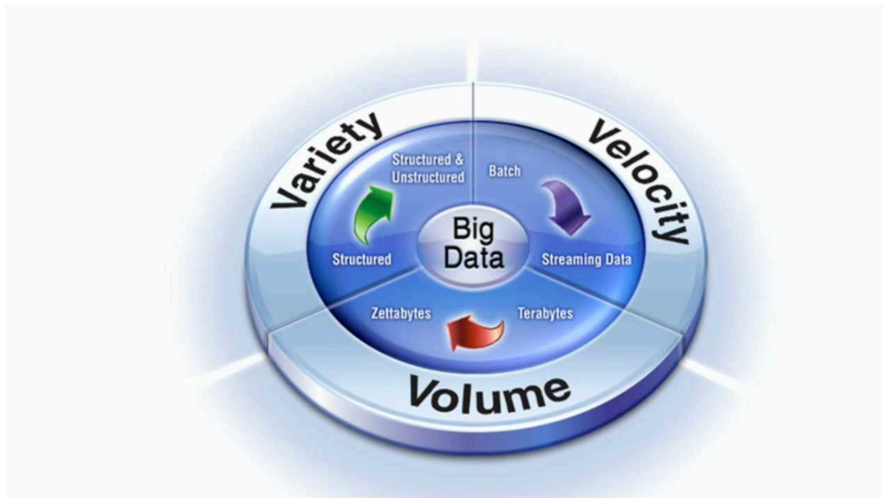
Till Convergence:

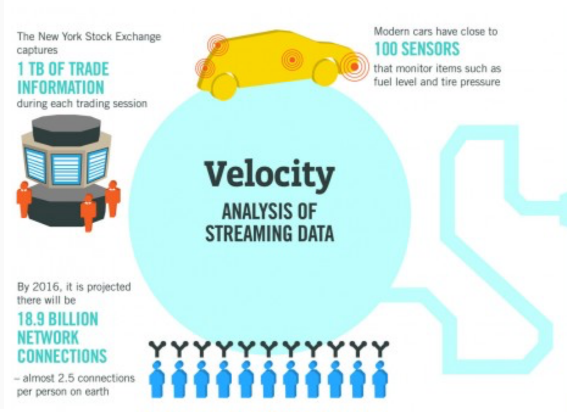


Shrinking algorithm converts a compute intensive SVM to communication Intensive ---- contrary to common belief!

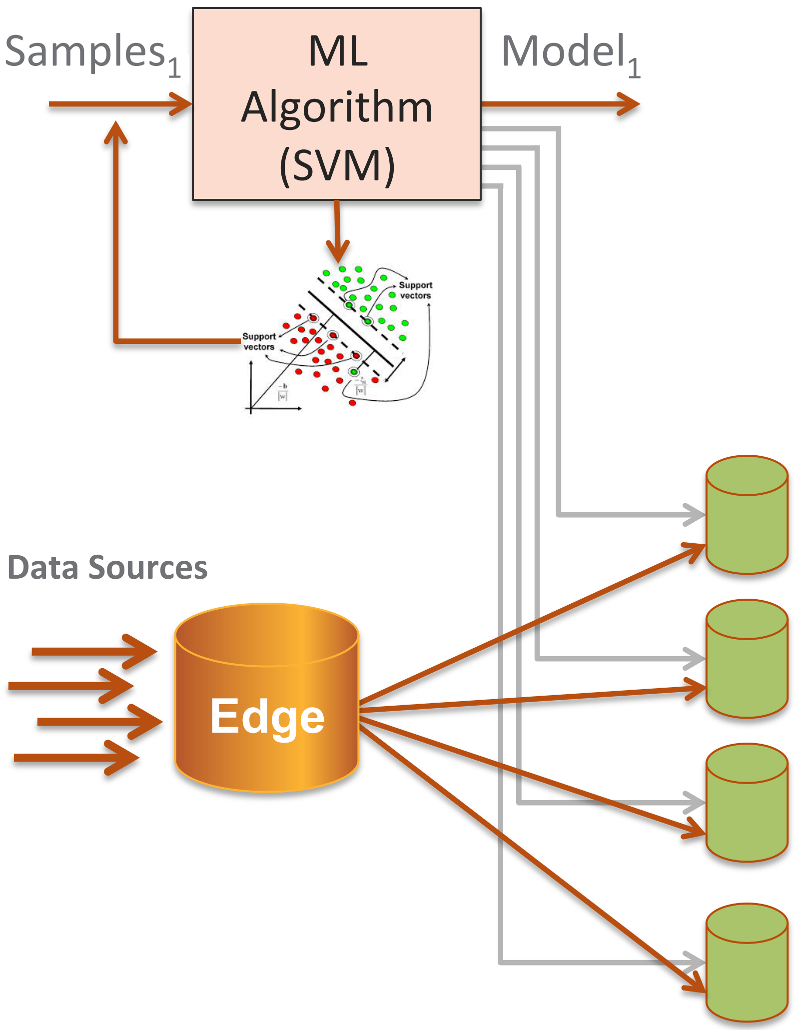
Addressing the Data Velocity

CERN produces 20 GB/s,
ARM produces 1 TB/day
HPC Performance Counters





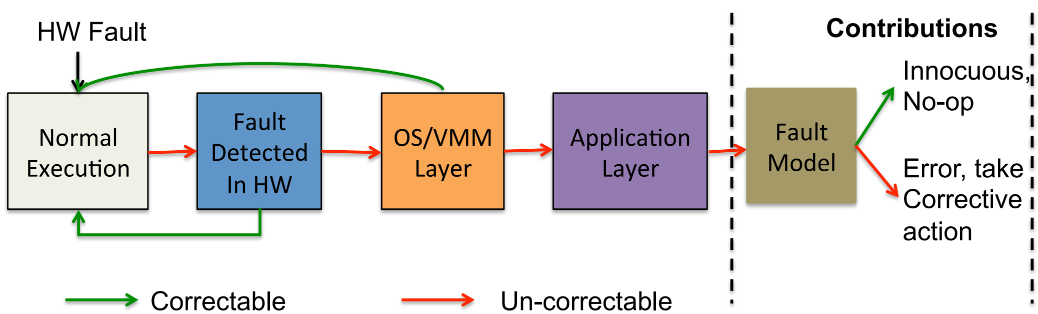
Parallel Incremental SFE Algorithms

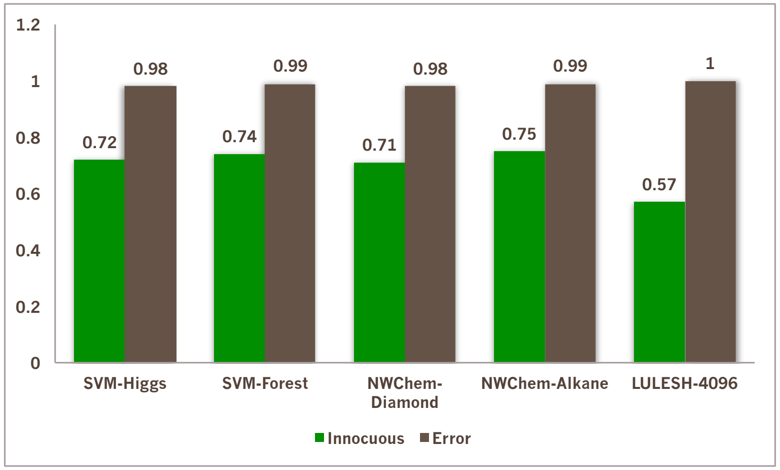


- Parallel incremental algorithms are critical
  - Address data volume and velocity
- Incremental algorithms
  - Create a new model using output of previous iteration + data of current window
- We use scikit (python ML)
  - Popular ML software
  - Currently non-incremental and sequential
- We have built a parallel incremental SVM on scikit

Solving HPC Problems using Machine Learning

- Fault tolerance is critical for Extreme Scale systems
- Systems suffer from several fault types in memory hierarchy
- However, fault != error
- Application fault modeling to minimize recovery





On-going and Future Work

- Extreme Scale Deep Learning
  - Communication avoiding algorithms for DNN
  - Deep Belief Networks (Unsupervised)
  - Convolutional Neural Networks (Images)
- Fault Tolerant ML algorithms – evaluation using MVAPICH2
  - Support Vector Machines (O(1) data movement for check-pointing)
  - Frequent Pattern Mining (O(1) space and O(n/p.log(p)) recovery
- SVM and Deep Learning for Accelerators
  - APUs, GPUs and MICs
- Applications to HPC
  - HPC is a “big data” producer
    - Can modeling with machine learning provide new insights?

Thanks!

Machine Learning Toolkit for Extreme Scale (MaTEx)
http://hpc.pnl.gov/matex

Contact: abhinav.vishnu@pnnl.gov