Machine Learning at the Limit

John Canny*^

* Computer Science Division

University of California, Berkeley

^ Yahoo Research Labs

@GTC, March, 2015

My Other Job(s)

Yahoo [Chen, Pavlov, Canny, KDD 2009]*

Ebay [Chen, Canny, SIGIR 2011]**

Quantcast 2011-2013

Microsoft 2014

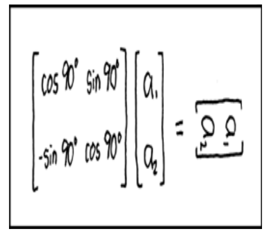
Yahoo 2015

* Best application paper prize
** Best paper honorable mention

Data Scientist’s Workflow

Sandbox

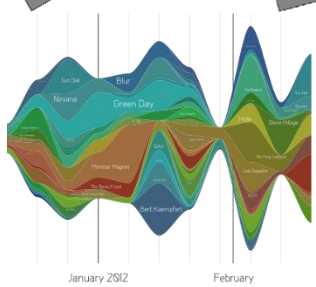
Digging Around in Data



Hypothesize
Model
Customize

Production

Large Scale Exploitation

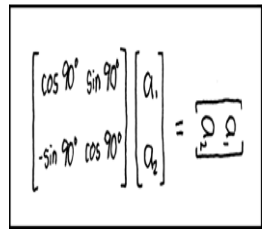


Evaluate
Interpret

Data Scientist's Workflow

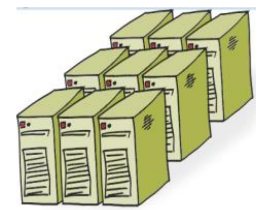
Sandbox

Digging Around
in Data

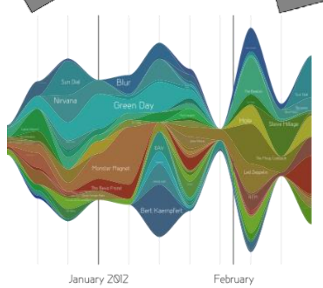


Hypothesize
Model
Customize

Production



Large Scale
Exploitation



Evaluate
Interpret

Why Build a New ML Toolkit?

- **Performance:** GPU performance pulling away from other platforms for *sparse* and dense data.
  Minibatch + SGD methods dominant on Big Data,…

- **Customizability:** Great value in customizing models (loss functions, constraints,…)

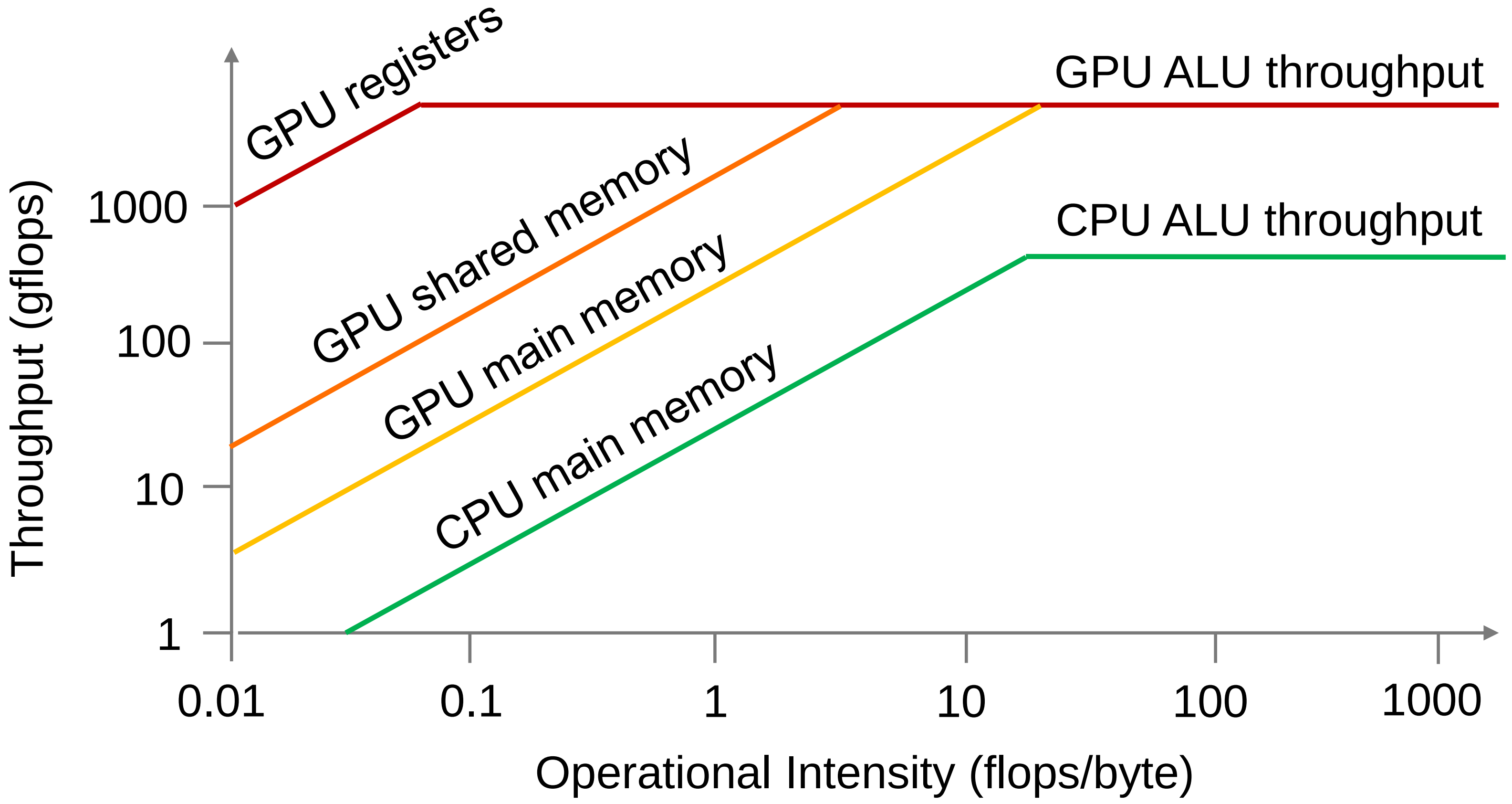
- **Explore/Deploy:** Explore fast, run the same code in prototype and production. Be able to run on clusters.

Desiderata

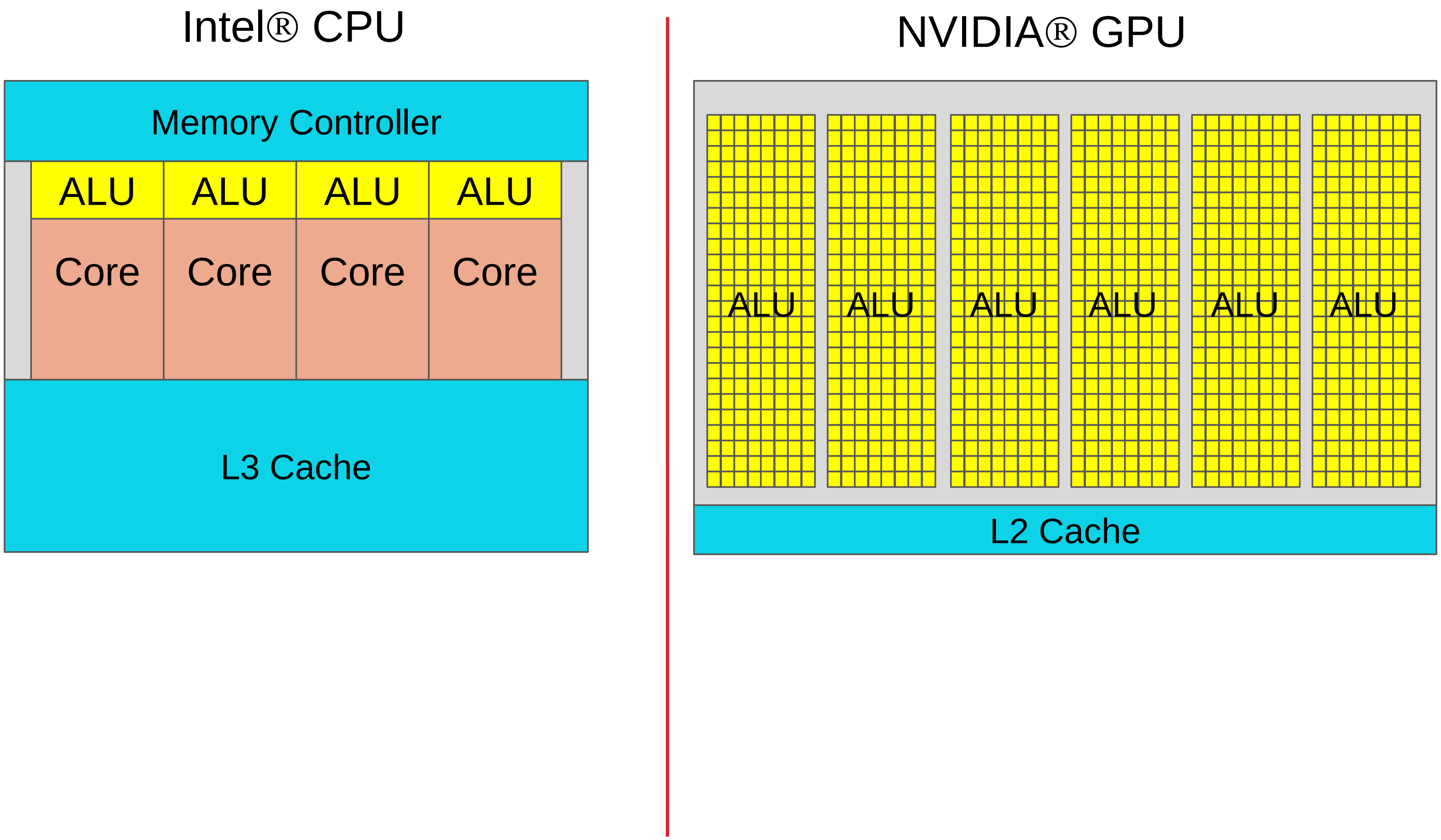
- **Performance:**
  - Roofline Design (single machine and cluster)
  - General Matrix Library with full CPU/GPU acceleration
- **Customizability:**
  - Modular Learner Architecture (reusable components)
  - Likelihood “Mixins”
- **Explore/Deploy:**
  - Interactive, Scriptable, Graphical
  - JVM based (Scala) w/ optimal cluster primitives

Roofline Design (Williams, Waterman, Patterson, 2009)

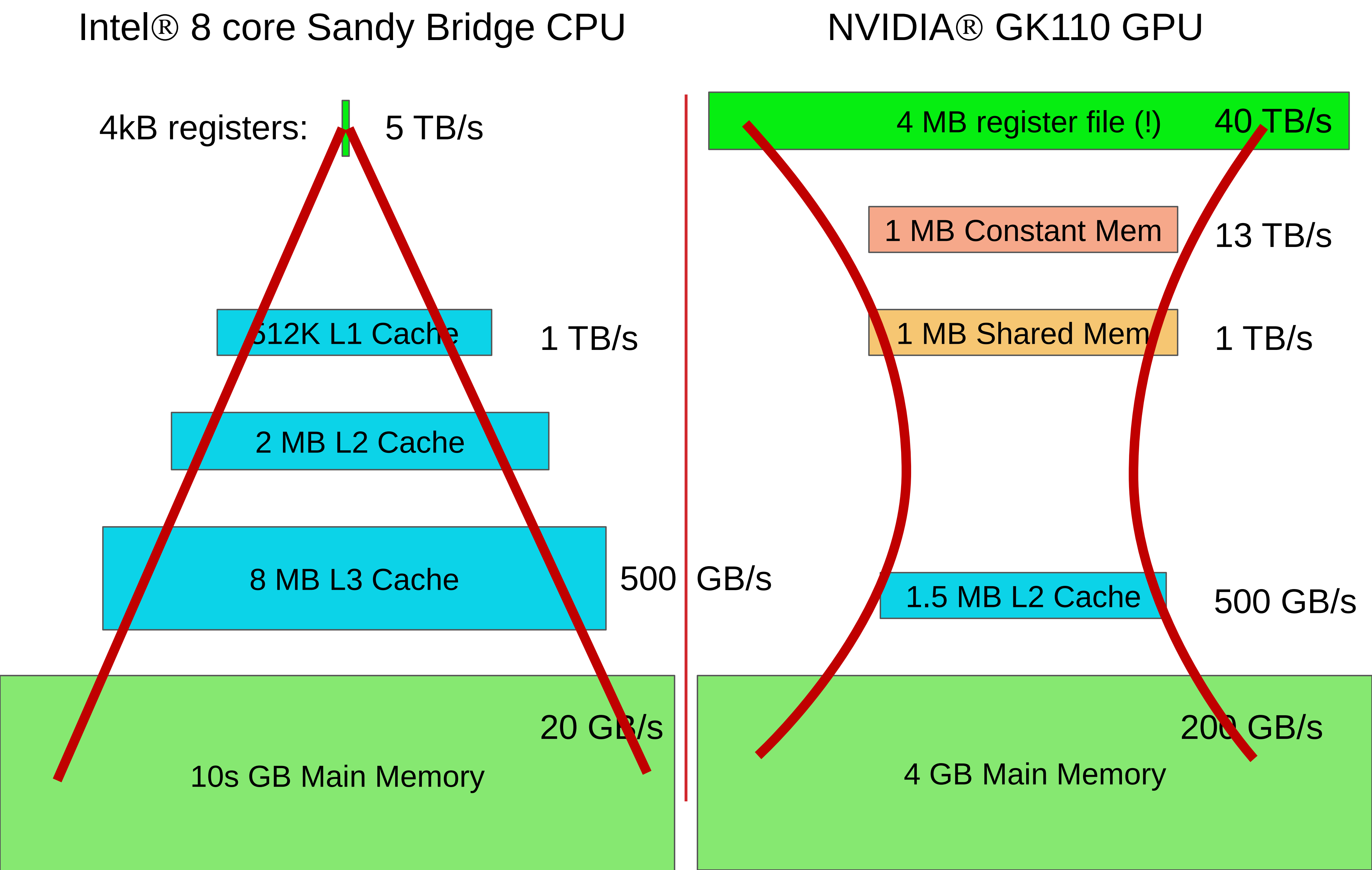
- Roofline design establishes fundamental performance limits for a computational kernel.



A Tale of Two Architectures



CPU vs GPU Memory Hierarchy



Natural Language Parsing (Canny, Hall, Klein, EMNLP 2013)

Natural language parsing with a state-of-the-art grammar (1100 symbols, 1.7 million rules, 0.1% dense)

End-to-End Throughput (4 GPUs):

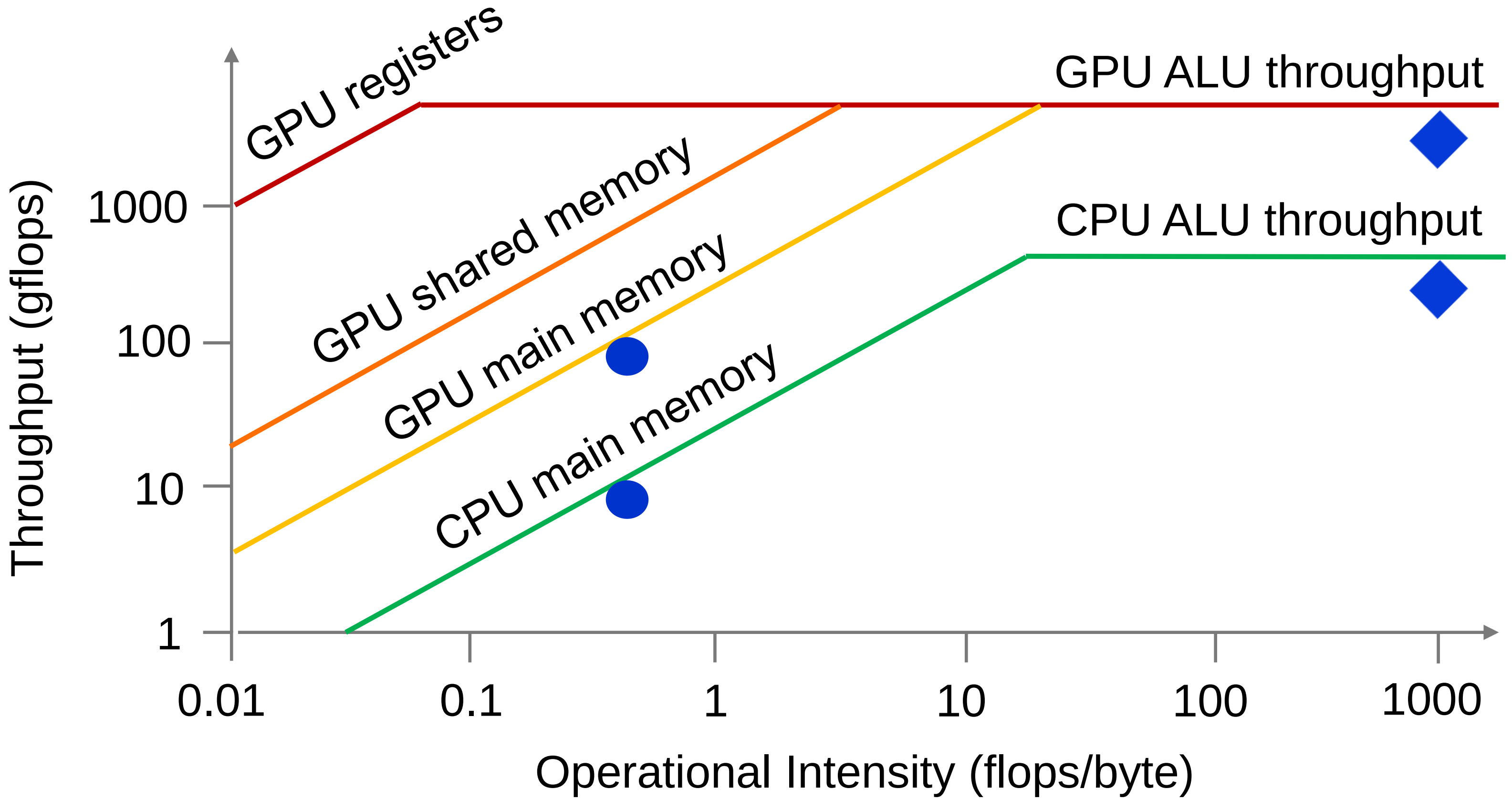
2-2.4 Teraflops (1-1.2 B rules/sec), **1000 sentences/sec**.

This is more than **10^5 speedup** for unpruned grammar evaluation (and it's the fastest constituency parser).

How: Compiled grammar into instructions, blocked groups of rules into a hierarchical 3D grid, fed many sentences in a queue, auto-tuned. **Max'ed** every resource on the device.

Roofline Design – Matrix kernels

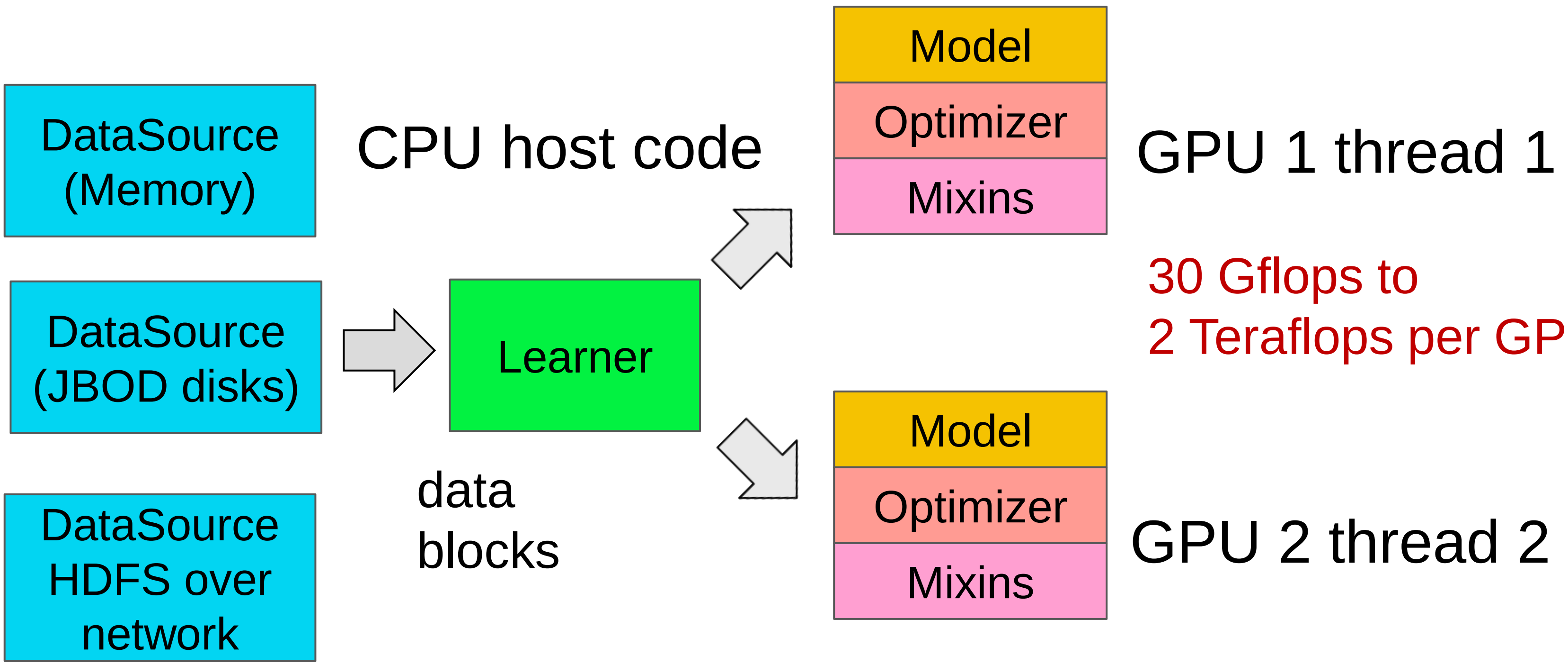
- Dense matrix multiply ◆
- Sparse matrix multiply ●



A Rooflined Machine Learning Toolkit

BIDMACH

Zhao+Canny
SIAM DM 13, KDD 13, BIGLearn 13



30 Gflops to
2 Teraflops per GPU

Compressed disk streaming at
~ 0.1-2 GB/s ≅ 100 HDFS nodes

:
:

Matrix + Machine Learning Layers

BIDMAT BIDMACH

Written in the beautiful Scala language:

- Interpreter with JIT, scriptable.
- Open syntax +,-,*, °,•,⊗ etc, math looks like math.
- Java VM + Java codebase – runs on Hadoop, Yarn, Spark.
- Hardware acceleration in C/C++ native code (CPU/GPU).
- Easy parallelism: Actors, parallel collections.
- Memory management (sort of ☺).
- Pre-built for multiple Platforms (Windows, MacOS, Linux).

Experience similar to Matlab, R, SciPy

Benchmarks

Recent benchmarks on some representative tasks:

- **Text Classification** on Reuters news data (0.5 GB)
- **Click prediction** on the Kaggle Criteo dataset (12 GB)
- **Clustering** of handwritten digit images (MNIST) (25 GB)
- **Collaborative filtering** on the Netflix prize dataset (4 GB)
- **Topic modeling (LDA)** on a NY times collection (0.5 GB)
- **Random Forests** on a UCI Year Prediction dataset (0.2 GB)
- **Pagerank** on two social network graphs at 12GB and 48GB

Benchmarks

Systems (single node)

- BIDMach
- VW (Vowpal Wabbit) from Yahoo/Microsoft
- Scikit-Learn
- LibLinear

Cluster Systems

- Spark v1.1 and v1.2
- Graphlab (academic version)
- Yahoo’s LDA cluster

Benchmarks: Single-Machine Systems

RCV1: Text Classification, 103 topics (0.5GB).
Algorithms were tuned to achieve similar accuracy.

System	Algorithm	Dataset	Dim	Time (s)	Cost ($)	Energy (KJ)
BIDMach	Logistic Reg.	RCV1	103	14	0.002	3
Vowpal Wabbit	Logistic Reg.	RCV1	103	130	0.02	30
LibLinear	Logistic Reg.	RCV1	103	250	0.04	60
Scikit-Learn	Logistic Reg.	RCV1	103	576	0.08	120

Benchmarks: Cluster Systems

Spark-XX = System with XX *cores*
BIDMach ran on one node with GTX-680 GPU

System A/B	Algorithm	Dataset	Dim	Time (s)	Cost ($)	Energy (KJ)
Spark-72 BIDMach	Logistic Reg.	RCV1	1 103	30 14	0.07 0.002	120 3
Spark-64 BIDMach	RandomForest	YearPred	1	280 320	0.48 0.05	480 60
Spark-128 BIDMach	Logistic Reg.	Criteo	1	400 81	1.40 0.01	2500 16

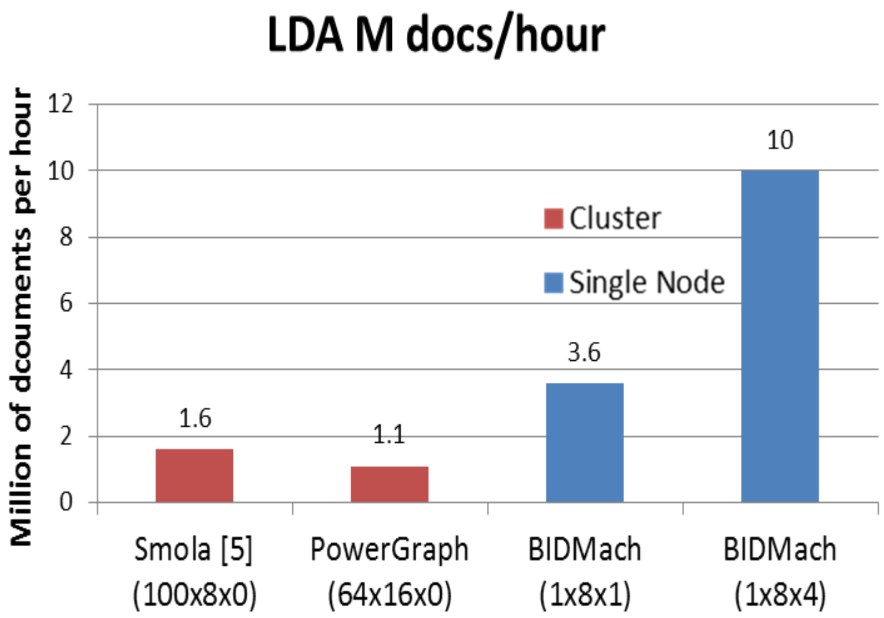
Benchmarks: Cluster Systems

Spark-XX or GraphLab-XX = System with XX *cores*
Yahoo-1000 had 1000 *nodes*

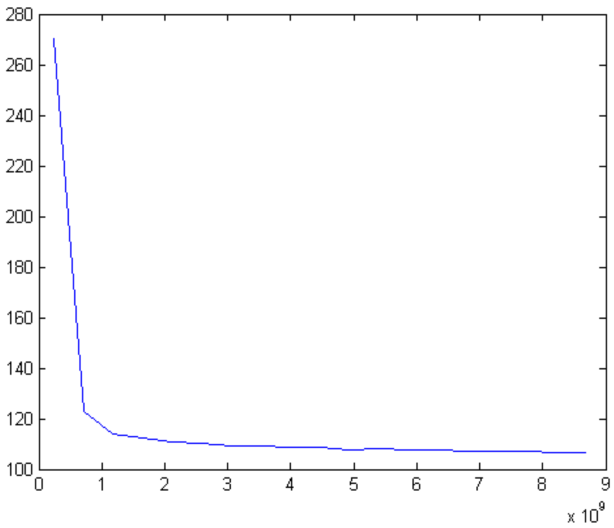
System A/B	Algorithm	Dataset	Dim	Time (s)	Cost ($)	Energy (KJ)
Spark-384 BIDMach	K-Means	MNIST	4096	1100 735	9.00 0.12	22k 140
GraphLab-576 BIDMach	Matrix Factorization	Netflix	100	376 90	16 0.015	10k 20
Yahoo-1000 BIDMach	LDA (Gibbs)	NYtimes	1024	220k 300k	40k 60	4E10 6E7

BIDMach at Scale

Latent Dirichlet Allocation



Convergence on 1TB data



BIDMach outperforms cluster systems on this problem, and has run up to 10 TB on one node.

Benchmark Summary

- BIDMach on a PC with NVIDIA GPU is at least **10x faster** than other single-machine systems for comparable accuracy.
- For Random Forests or single-class regression, BIDMach on a GPU node is **comparable with 8-16 worker clusters**.
- For multi-class regression, factor models, clustering etc., GPU-assisted BIDMach is **comparable to 100-1000-worker clusters**. Larger problems correlate with larger values in this range.

In the Wild (Examples from Industry)

- **Multilabel regression** problem (summer intern project):
  - Existing tool (single-machine) took ~ 1 week to build a model.
  - BIDMach on a GPU node takes 1 hour (120x speedup)
  - Iteration and feature engineering gave +15% accuracy.

- **Auction simulation** problem (cluster job):
  - Existing tool simulates auction variations on log data.
  - On NVIDIA 3.0 devices (64 registers/thread) we achieve a 70x speedup over a reference implementation in Scala
  - On NVIDIA 3.5 devices (256 registers/thread) we can move auction state entirely into register storage and gain a 400x speedup.

In the Wild (Examples from Industry)

- **Classification** (cluster job):
  - Cluster job (logistic regression) took 8 hours.
  - BIDMach version takes < 1 hour on a single node.

- **SVMs for image classification** (single machine)
  - Large multi-label classification took 1 week with LibSVM.
  - BIDMach version (SGD-based SVM) took 90 seconds.

Performance Revisited

- BIDMach had a **10x-1000x cost advantage** over the other systems. The ratio was higher for larger-scale problems.
- Energy savings were similar to the cost savings, at **10x-1000x**.

But why??

- We only expect about 10x from GPU acceleration?
- See our Parallel Forall post:

http://devblogs.nvidia.com/parallelforall/bidmach-machine-learning-limit-gpus/

BIDMach ML Algorithms

1. Regression (logistic, linear)
2. Support Vector Machines
3. k-Means Clustering
4. Topic Modeling - Latent Dirichlet Allocation
5. Collaborative Filtering
6. NMF – Non-Negative Matrix Factorization
7. Factorization Machines
8. Random Forests
9. Multi-layer neural networks
10. IPTW (Causal Estimation)
11. ICA

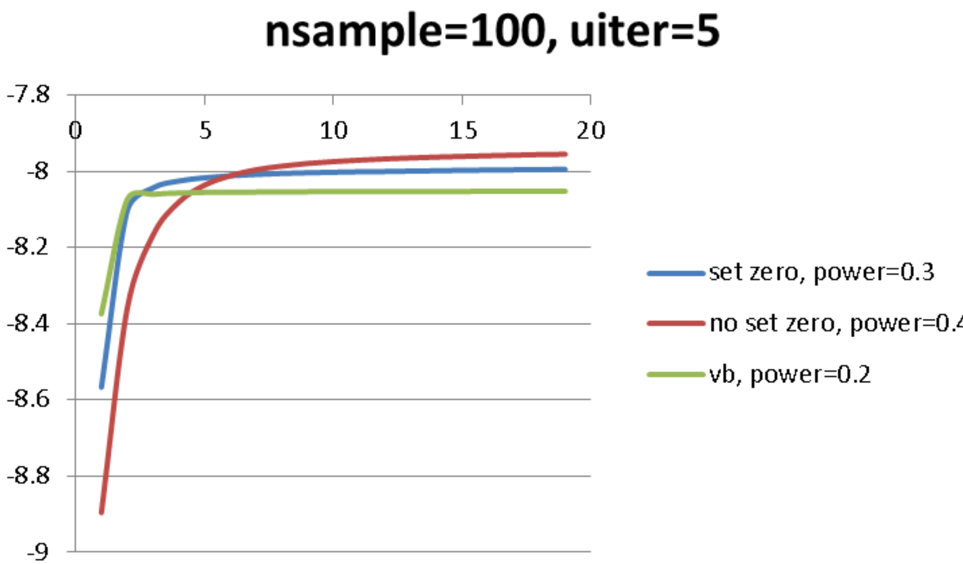
= Likely the fastest implementation available

Research: SAME Gibbs Sampling

- SAME sampling accelerates standard Gibbs samplers with discrete+continuous data.
- Our first instantiation gave a **100x** speedup for a very widely-studied problem (Latent Dirichlet Allocation), and was **more accurate** than any other LDA method we tested:

- SAME sampling is a general approach that should be **competitive with custom symbolic methods**.

- Arxiv paper on BIDMach website.

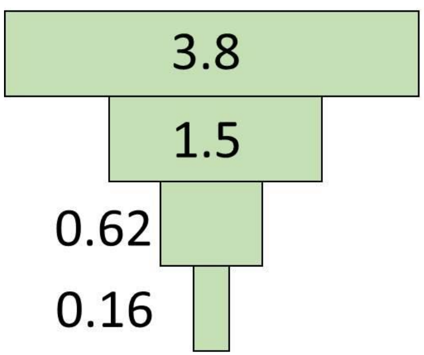


Research: Rooflined cluster computing

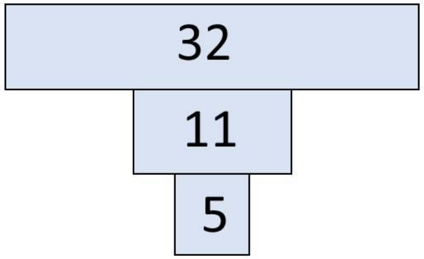
Kylix (ICPP 2014)

- Near optimal model aggregation for sparse problems.
- Communication volume across layers has a characteristic Kylix shape:

Twitter (8x4x2)



Yahoo (16x4)



Software (version 1.0 just released)

Code: github.com/BIDData/BIDMach

Wiki: http://bid2.berkeley.edu/bid-data-project/overview/

BSD open source libs and dependencies, papers

In this release:

- Random Forests, ICA
- Double-precision GPU matrices
- Ipython/IScala Notebook
- Simple DNNs

Wrapper for Berkeley's Caffe coming soon…

Thanks

Sponsors:

Microsoft Azure

amazon web services™

NVIDIA

Collaborators:

YAHOO!®

MANY TREES INC