Performance Tuning Tips for Apache SPARK Machine Learning workloads

ShreeHarsha GN

Senior Staff Software Engineer, IBM Power System Performance

Amir Sanjar

IBM OpenPower Solution Architect, OpenPOWER solutions and Development

Agenda

Spark Overview

Why OpenPower ?

OpenPower Design & Benefits

Spark on OpenPower

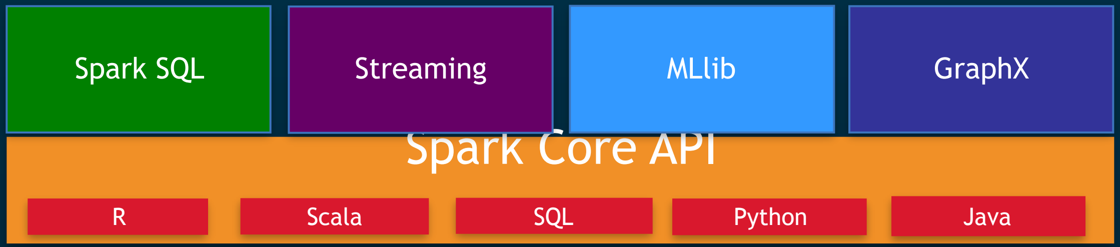
Performance Tuning Tips for Apache SPARK Machine Learning Workloads

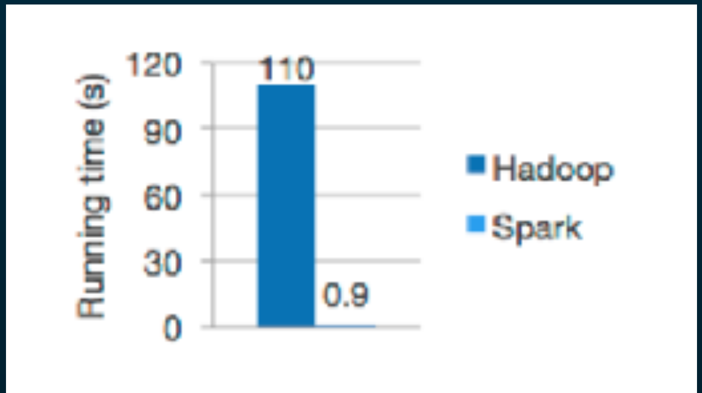
Demo

What is Apache Spark

- Unified Analytics Platform
  - Combine streaming, graph, machine learning and sql analytics on a single platform
  - Simplified, multi-language programming model
  - Interactive and Batch
- In-Memory Design
  - Pipelines multiple iterations on single copy of data in memory
  - Superior Performance
  - Natural Successor to MapReduce

Fast and general engine for large-scale data processing

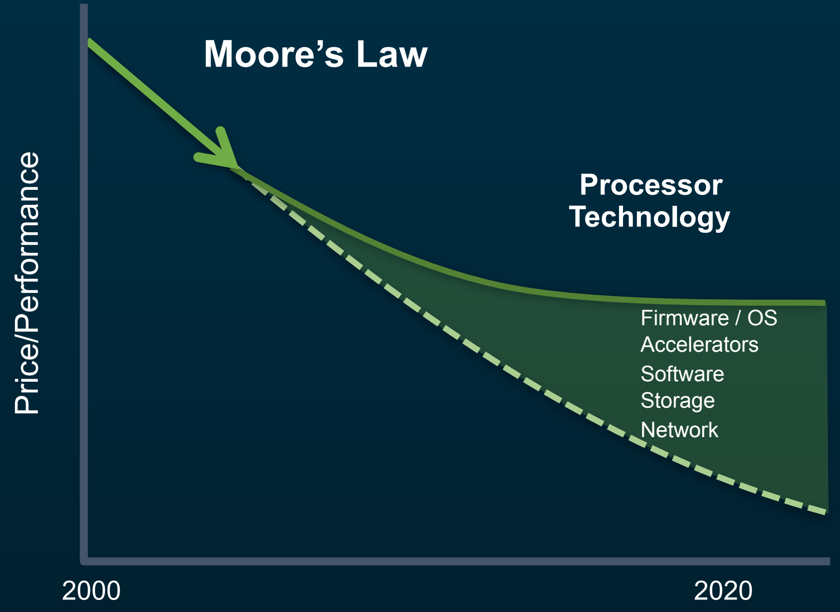




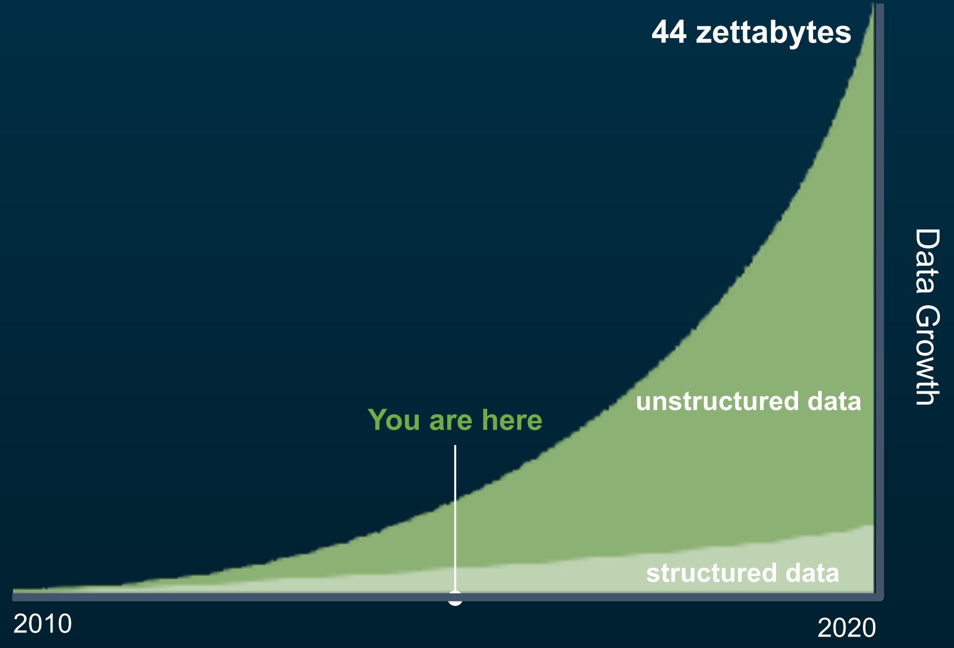
Logistic regression in Hadoop and Spark

Today's challenges demand innovation

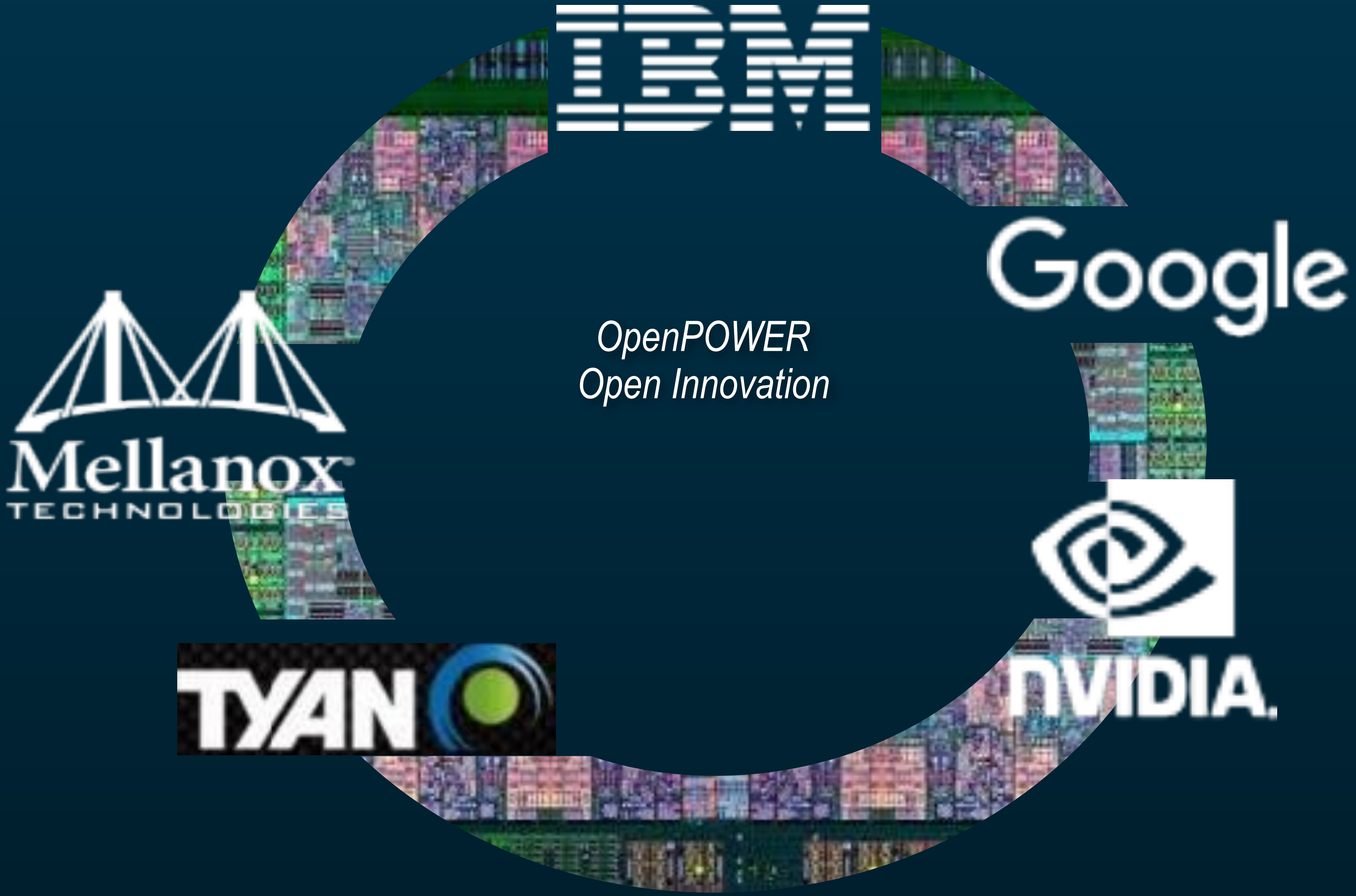
Full system and stack open innovation required



Data holds competitive value



Open Power Ecosystem



Spark on OpenPower

- **Streaming and SQL benefit from High Thread Density and Concurrency**
  - Processing multiple packets of a stream and different stages of a message stream pipeline
  - Processing multiple rows from a query

Spark on OpenPower

- **Machine Learning benefits from Large Caches and Memory Bandwidth**
  - Iterative Algorithms on the same data
  - Fewer core pipeline stalls and overall higher throughput

Spark on OpenPower

- **Graph also benefits from Large Caches, Memory Bandwidth and Higher Thread Strength**

  - Flexibility to go from 8 SMT threads per core to 4 or 2

  - Manage Balance between thread performance and throughput

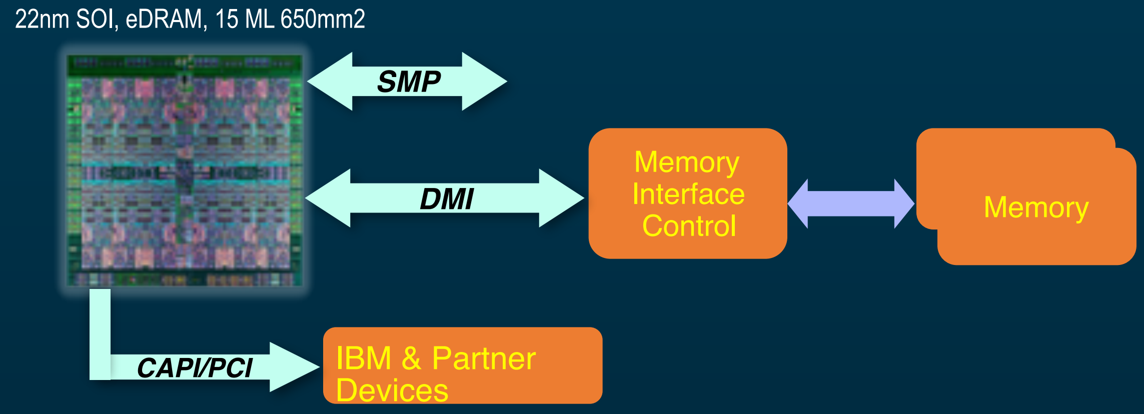
Spark on OpenPower

- **Headroom**
  - Balanced resource utilization, more efficient scale-out
  - Multi-tenant deployments

Machine workload deployment on Spark

- Bigtop
  - https://git-wip-us.apache.org/repos/asf?p=bigtop.git

POWER8 Processor - Design



Cores
- 12 cores / 8 threads per core
- TDP: 130W and 190W
- 64K data cache, 32K instruction cache

Accelerators
- Crypto & memory expansion
- Transactional Memory

Caches
- 512 KB SRAM L2 / core
- 96 MB eDRAM shared L3

Memory Subsystem
- Memory buffers with 128MB Cache
- ~70ns latency to memory

Bus Interfaces
- Durable Memory attach Interface (DMI)
- Integrated PCIe Gen3
- SMP Interconnect for up to 4 sockets

Coherent Accelerator Processor Interface (CAPI)

Virtual Addressing
- Accelerator can work with same memory addresses that the processors use
- Pointers de-referenced same as the host application
- Removes OS & device driver overhead

Hardware Managed Cache Coherence
- Enables the accelerator to participate in "Locks" as a normal thread
- Lowers Latency over IO communication model

6 Hardware Partners developing with CAPI

Over 20 CAPI Solutions
- All listed here http://ibm.biz/powercapi

Examples of Available CAPI Solutions
- IBM Data Engine for NoSQL
- DRC Graphfind analytics
- Erasure Code Acceleration for Hadoop

Newly Announced OpenPOWER systems and solutions:

http://openpowerfoundation.org/wp-content/uploads/2016/04/HardwareRevealFlyerFinal.pdf

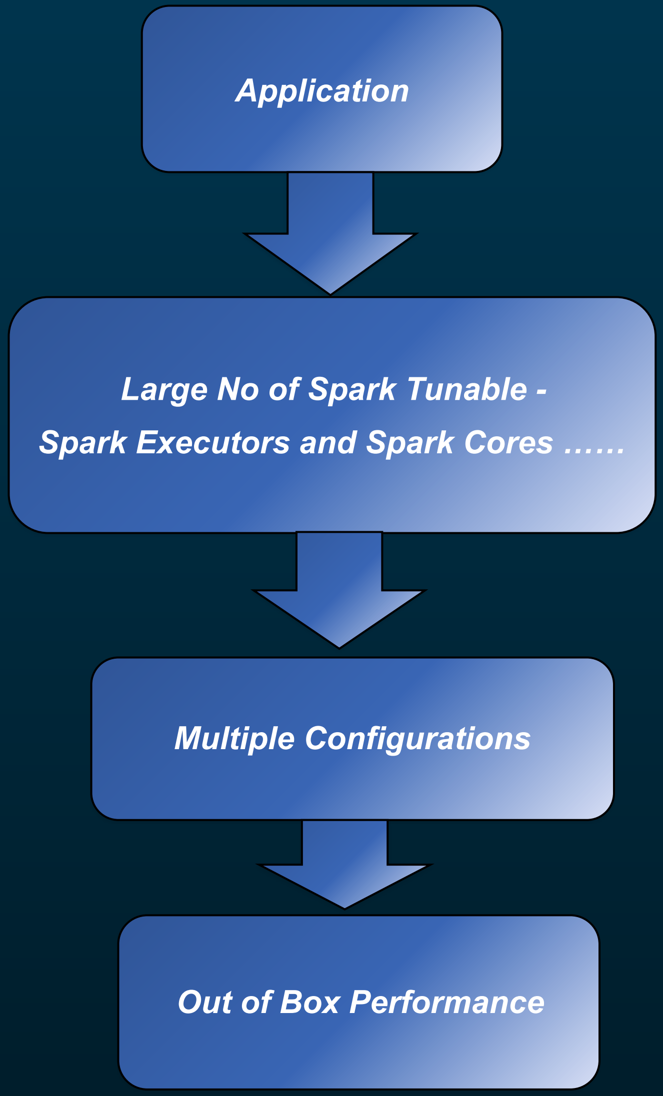
Performance Tuning Tips for SPARK Machine Learning Workloads

Methodology:

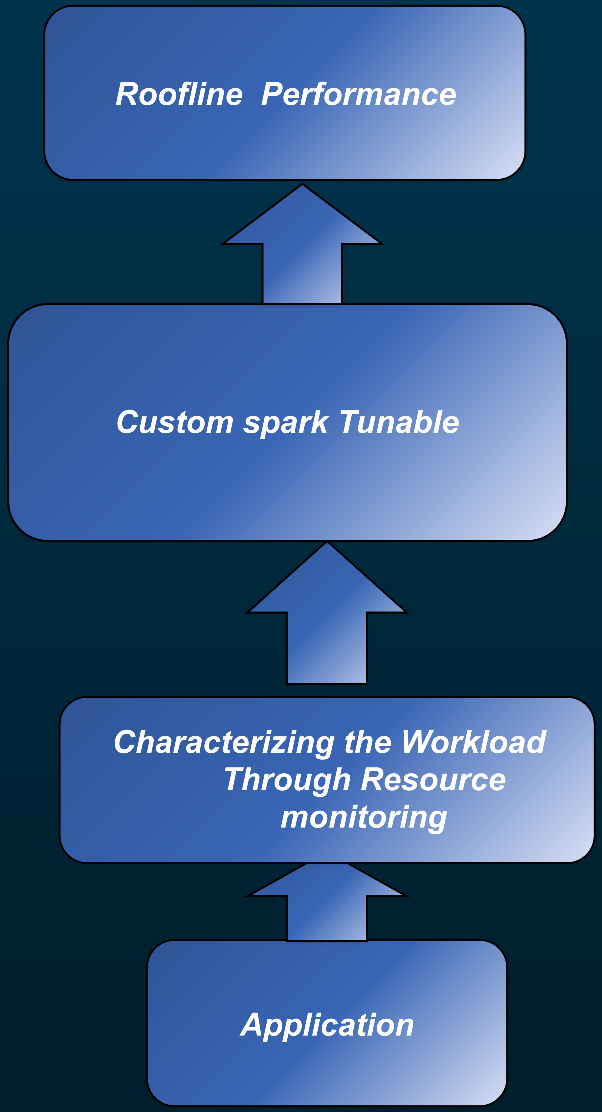
Alternating Least Squares Based Matrix Factorization application

Optimization Process:

Spark executor Instances
Spark executor cores
Spark executor memory
Spark shuffle location and manager
RDD persistence storage level



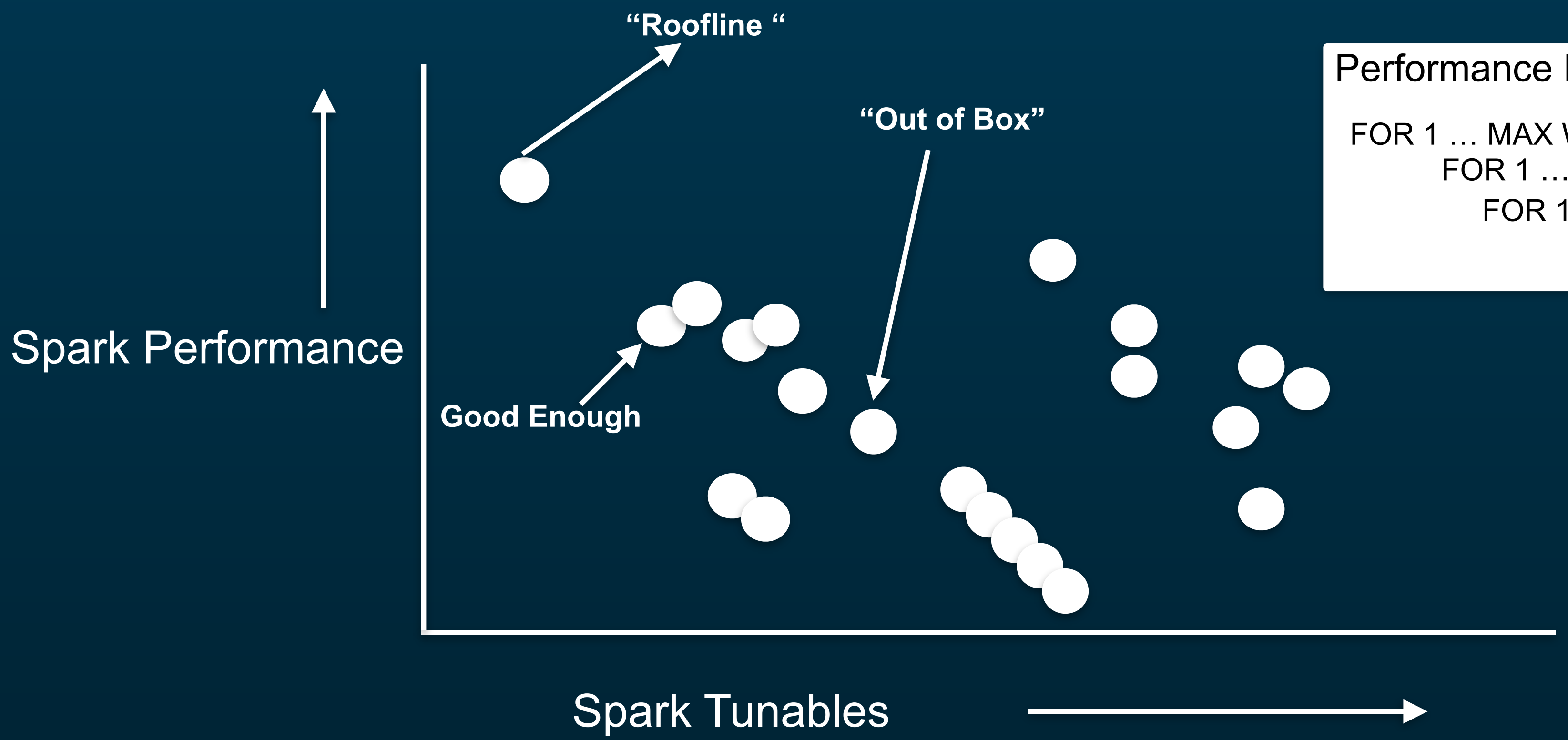
Bottom Up Approach



Top Down Approach

Courtesy Rajaram Krishnamurthy

Roofline SPARK Performance Model



Performance Navigation Automation Script

```
FOR 1 … MAX WORKERS
    FOR 1 …. MAX CPU PER NODE
        FOR 1 … MAX THREADS PER CPU
            FOR 1… MAX PARTITIONS
```

"Roofline" **Performance Navigation** uses system resource workload characterization and analysis to look for fundamental inefficiencies

Courtesy Rajaram Krishnamurthy

WorkFlow

- Matrix Factorization from SPARKBENCH
  - https://github.com/SparkTC/spark-bench
- Training
- Validation
- Prediction

Courtesy Rajaram Krishnamurthy

Matrix Factorization with Alternating Least Squares

Data generation parameters	**Value**
Rows in data matrix	62000
Columns in data matrix	62000
Data set size	100 GB

Parameters used for data generation in MF application

Courtesy Rajaram Krishnamurthy

Matrix Factorization with Alternating Least Squares

Spark parameter	**Value for MF**
Master node	1
Worker nodes	6
Executors per Node	1
Executor cores	80 / 40 /24
Executor Memory	480 GB
Shuffle Location	HDDs
Input Storage	HDFS

Spark environment details for application evaluation

Courtesy Rajaram Krishnamurthy

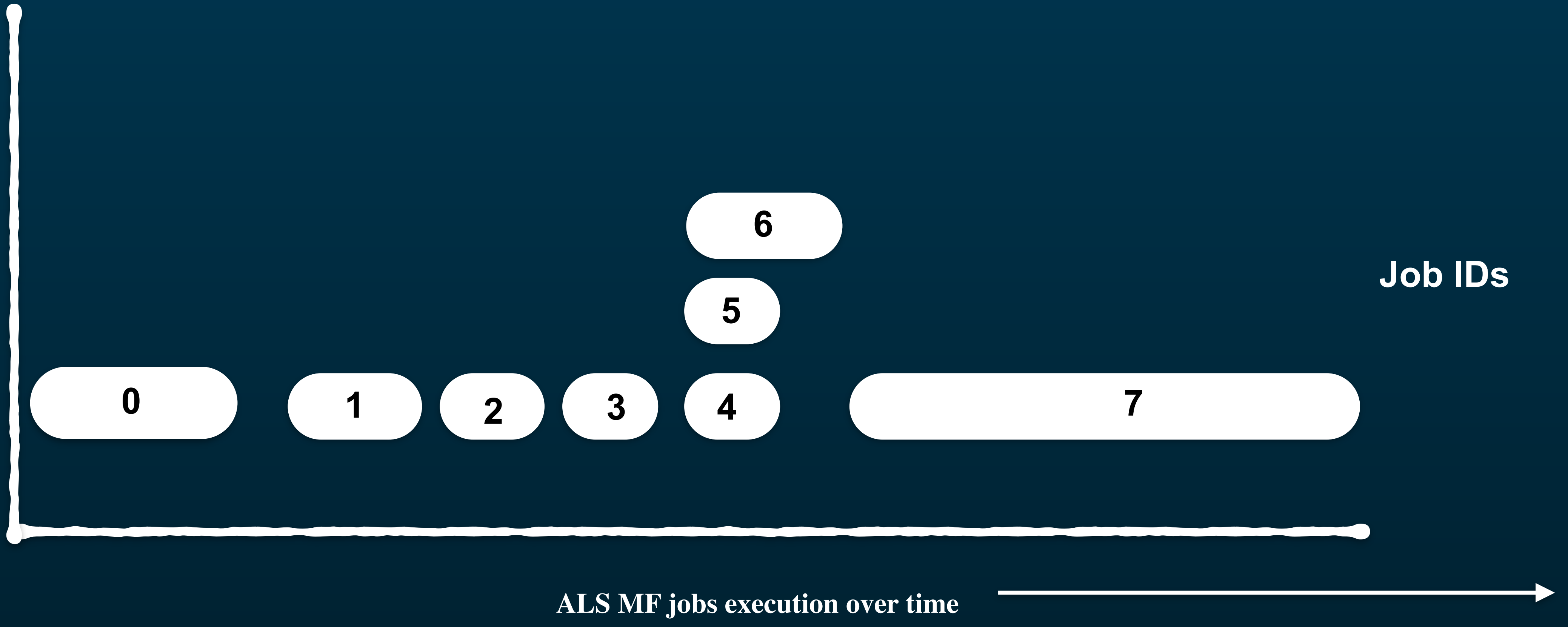
Matrix Factorization with Alternating Least Squares

Job	Function	Description / API called
7	Mean at MFApp.java	AbstractJavaRDDLike.map MatrixFactorizationModel.predict JavaDoubleRDD.mean
6	Aggregate at MFModel.scala	MatrixFactorizationModel.predict MatrixFactorizationModel.countApproxDistinctUserProduct
5	First at MFModel.scala	ml.recommendation.ALS.computeFactors
4	First at MFModel.scala	ml.recommendation.ALS.computeFactors
3	Count at ALS.scala	ALS.train and ALS.intialize
2	Count at ALS.scala	ALS.train
1	Count at ALS.scala	ALS.train
0	Count at ALS.scala	ALS.train

Description of jobs in MF application

Courtesy Rajaram Krishnamurthy

Matrix Factorization with Alternating Least Squares



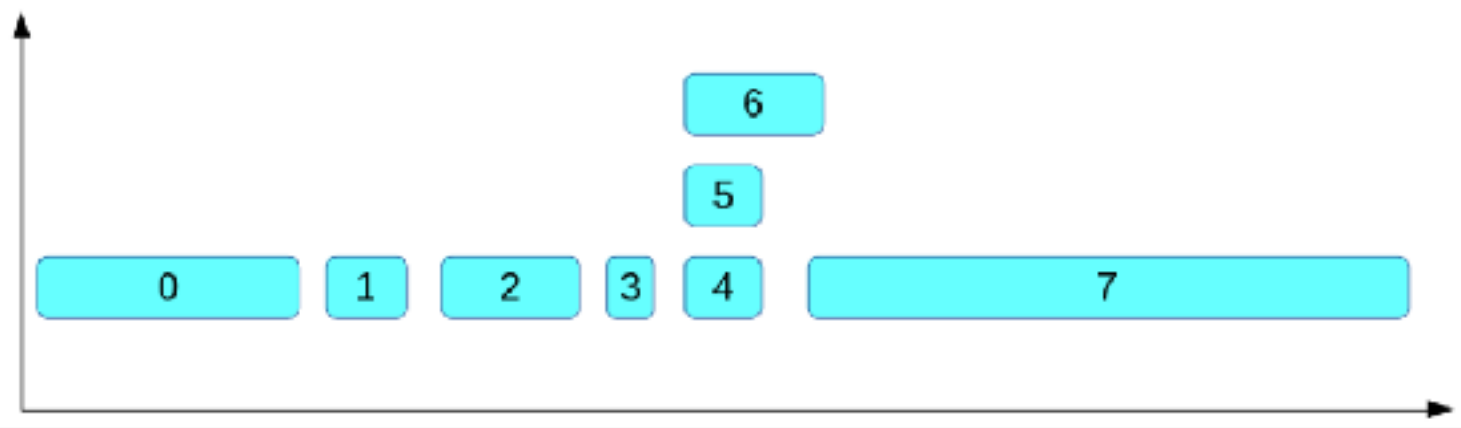
Courtesy Rajaram Krishnamurthy

Matrix Factorization with Alternating Least Squares

Data generation parameters	Value
Rows in data matrix	62000
Columns in data matrix	62000
Data set size	100 GB

Spark parameter	Value for MF
Master node	1
Worker nodes	6
Executors per Node	1
Executor cores	80 / 40 /24
Executor Memory	480 GB
Shuffle Location	HDDs
Input Storage	HDFS

Job	Function	Description / API called
7	Mean at MFApp.java	AbstractJavaRDDLike.map MatrixFactorizationModel.predict JavaDoubleRDD.mean
6	Aggregate at MFModel.scala	MatrixFactorizationModel.predict MatrixFactorizationModel.countApproxDistinctUserProduct
5	First at MFModel.scala	ml.recommendation.ALS.computeFactors
4	First at MFModel.scala	ml.recommendation.ALS.computeFactors
3	Count at ALS.scala	ALS.train and ALS.intialize
2	Count at ALS.scala	ALS.train
1	Count at ALS.scala	ALS.train
0	Count at ALS.scala	ALS.train



Parameters used for data generation in MF application

Courtesy Rajaram Krishnamurthy

Analyzing SPARK Configuration Sweep

Various configurations tried in optimizing MF application on Spark

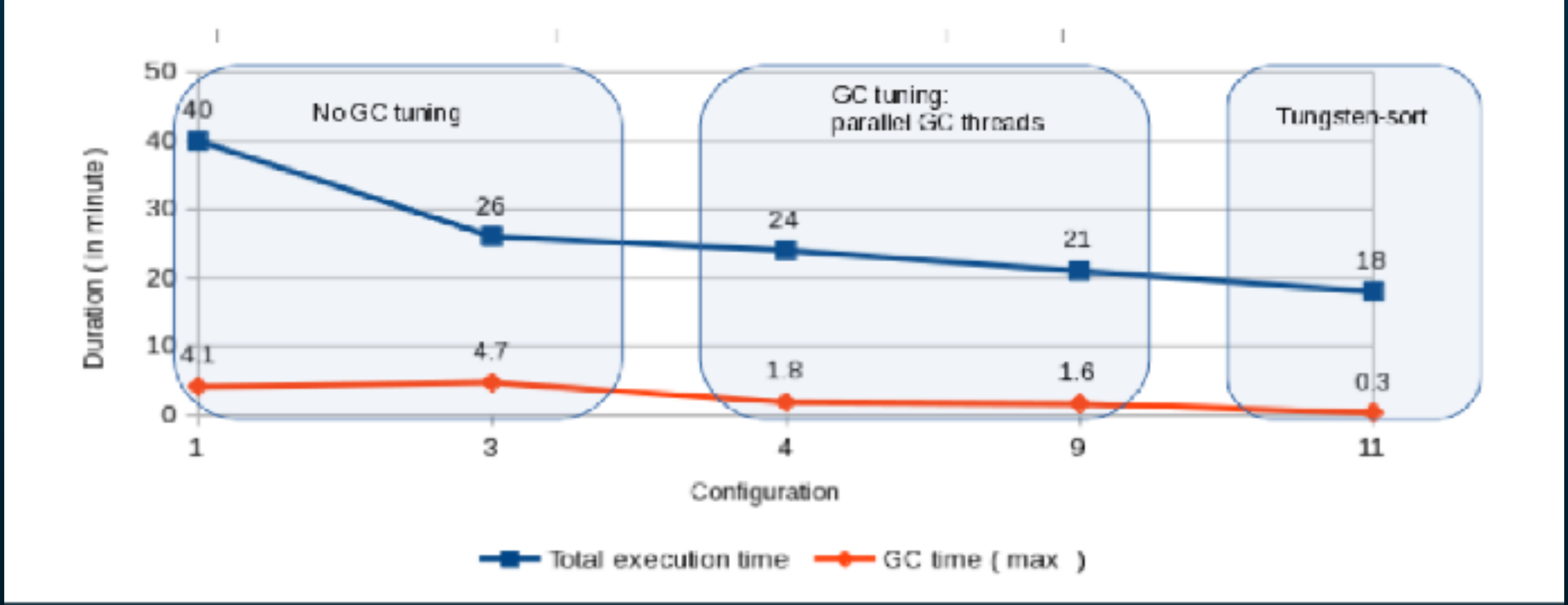
Configuration	1	2	3	4	5	6	7	8	9	10	11
Spark executor cores	80	80	40	40	40	40	40	40	24	24	24
GC options	Default	Default	Default	ParallelGCthreads=40	ParallelGCthreads=40	ParallelGCthreads=40	ParallelGCthreads=40	ParallelGCthreads=40	ParallelGCthreads=24	ParallelGCthreads=24	Default
RDD compression	TRUE	FALSE	FALSE	FALSE	TRUE	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE
Storage level	memory_and_disk	memory_only	memory_only	memory_only	memory_and_disk_ser	memory_only_ser	memory_only	memory_only	memory_and_disk_ser	memory_and_disk_ser	memory_and_disk_ser
Partition numbers	1000	1000	1000	1000	1000	1000	800	1200	1000	1000	1000
Shuffle Manager	Sort based	Sort based	Sort based	Sort based	Sort based	Sort based	Sort based	Sort based	Sort based	Tungsten-sort	Tungsten-sort
Run-time (minutes)	40	34	26	24	20	25	26	27	21	19	18

Courtesy Rajaram Krishnamurthy

GC and Memory Foot print

Configuration	Run time of last stage	GC time of last stage
1	12 min	4.4 min
4	4.4 min	1.8 min
9	3.5 min	1.6 min
11	**47s**	**16s**

Run time and GC time of Stage 68 for different configurations



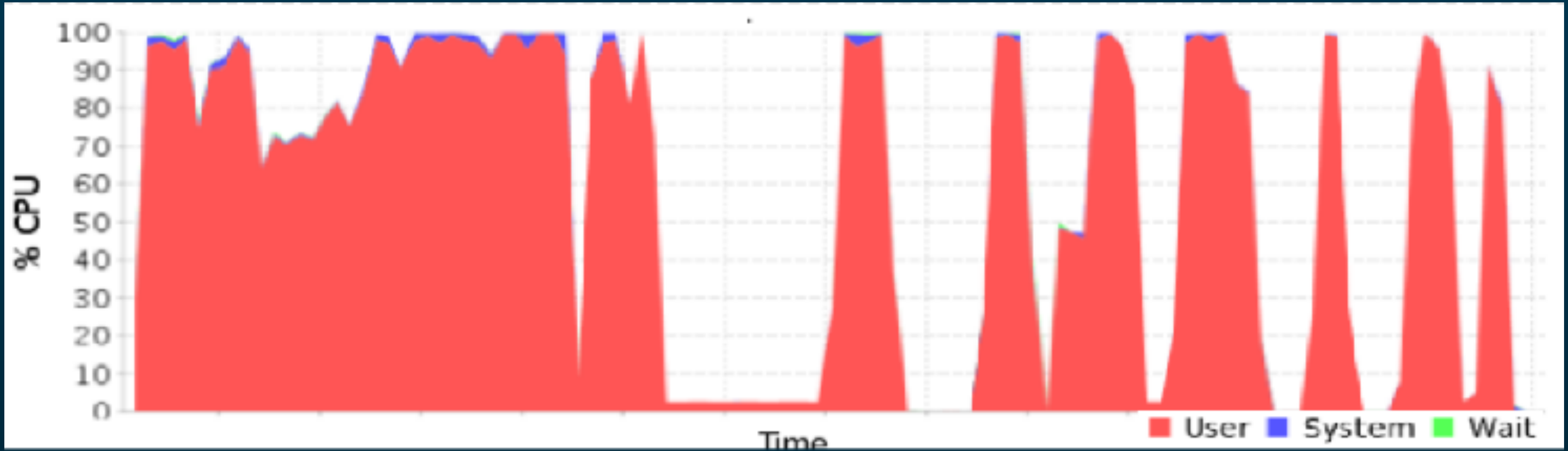
Courtesy Rajaram Krishnamurthy

Last Stage Analysis

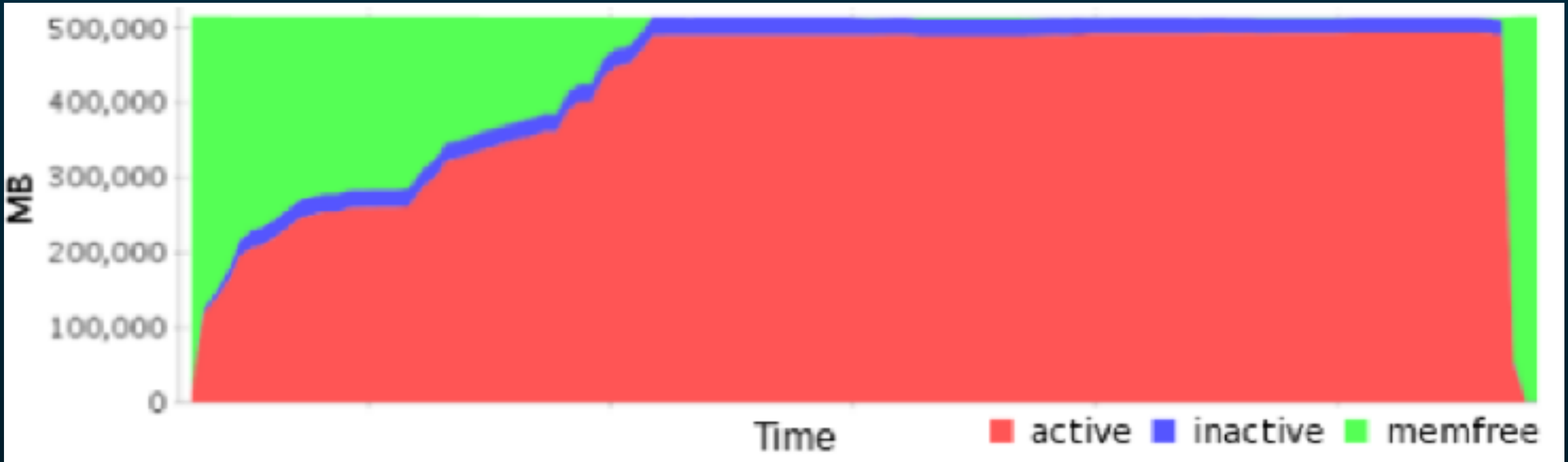
	Configuration	Duration	GC Time	Shuffle Details
#1	80 threads, Default GC, Memory+Disk	8.1 mins	1.2 mins	111 MB (Shuffle read), 1894MB (Shuffle Spill memory), 142 MB (Shuffle Spill Disk)
#5	40 threads, 40 GC, M+D Serialized	1.6 mins	17 secs	111 MB (Shuffle read)
#11	24 threads, M+D Serialized, Tungsten	38 secs	11 secs	111 MB (Shuffle read)

Courtesy Rajaram Krishnamurthy

Characterizing Configuration #1



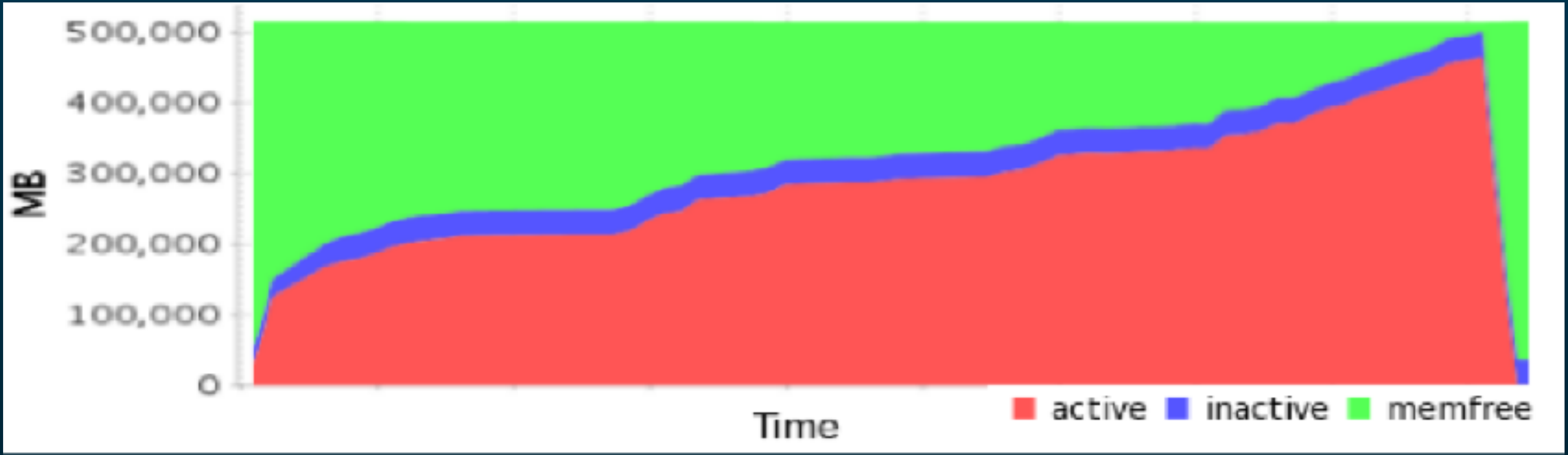
CPU utilization on a worker node (configuration 1)



Memory utilization on a worker node (configuration 1)

Courtesy Rajaram Krishnamurthy

Characterizing Configuration #1 and Configuration #11



Memory footprint of configuration 11

Courtesy Rajaram Krishnamurthy

Summary - How to Optimize Closer to Roofline Performance Faster?

- Classify workload into CPU, memory, IO or mixed (CPU, memory, IO) intensive

- Characterize "out-of-the-box" workload to understand CPU, Memory, IO and Network performance characteristics

- Floorplan cluster resources

- Tune "out-of-the-box" workload to navigate "Roofline" performance space in the above named dimensions

– If workload is memory/IO/Network bound then tune SPARK to increase operational intensity operations/byte as much as possible to make it CPU bound

- Divide search space into regions and perform exhaustive search

Courtesy Rajaram Krishnamurthy

Performance Wall

Accelerator Technology

Mellanox Interconnect

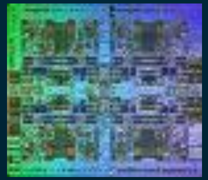
2015	2016	2017
Connect-IB FDR Infiniband PCIe Gen3	ConnectX-4 EDR Infiniband CAPI over PCIe Gen3	ConnectX-5 Next-Gen Infiniband Enhanced CAPI over PCIe Gen4

NVIDIA GPUs

2015	2016	2017
Kepler PCIe Gen3	Pascal NVLink	Volta Enhanced NVLink

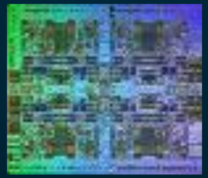
IBM CPUs

POWER8

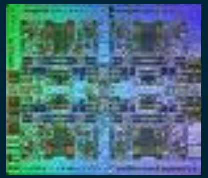


OpenPower
CAPI Interface

POWER8 with NVLink



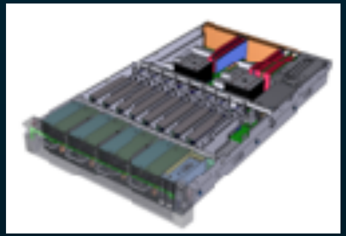
POWER9



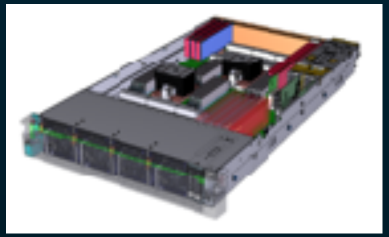
Enhanced
CAPI & NVLink

2015

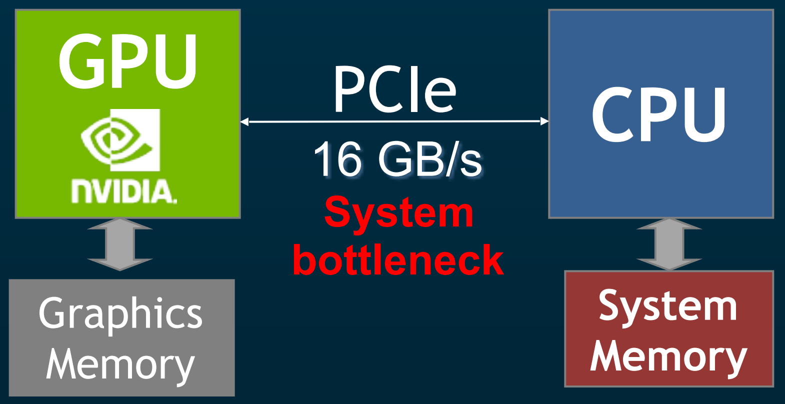
2016

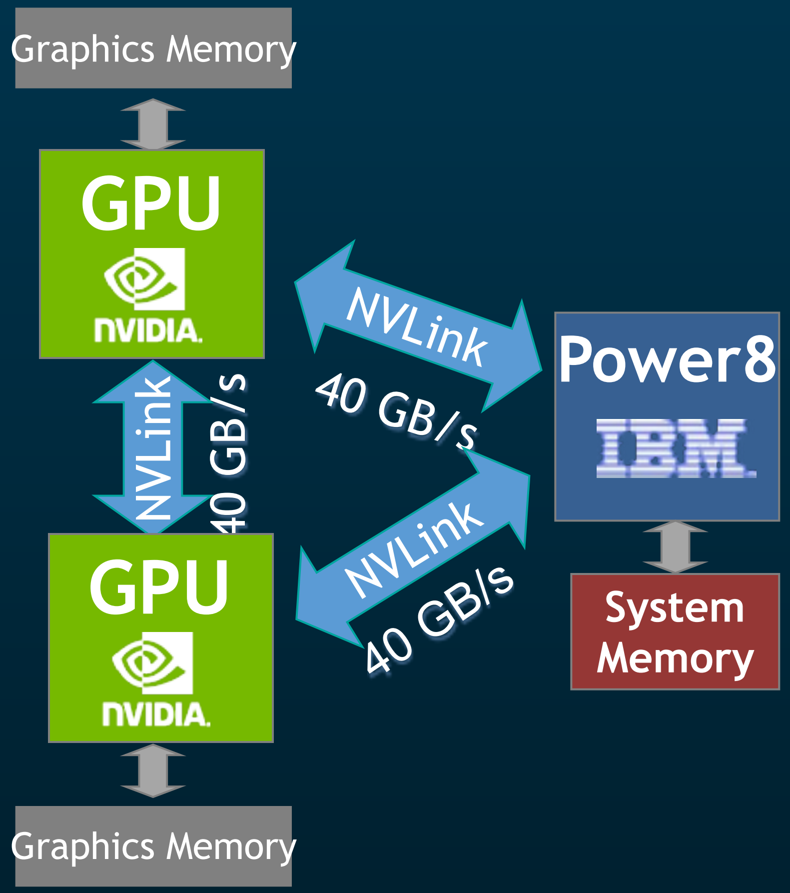


2017



OpenPOWER Technology: 2.5x Faster CPU-GPU Connection via NVLink

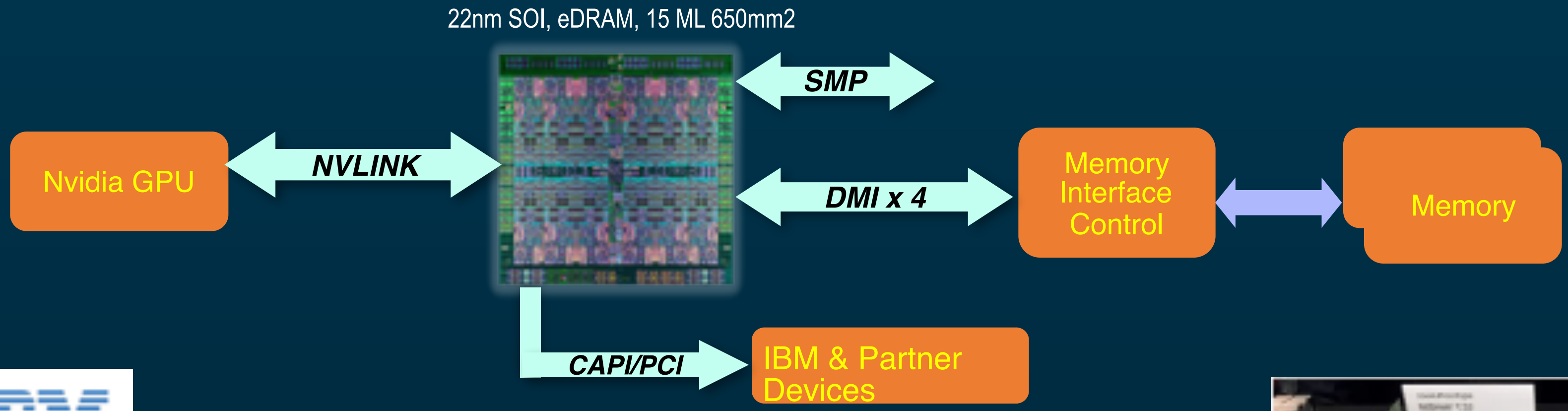




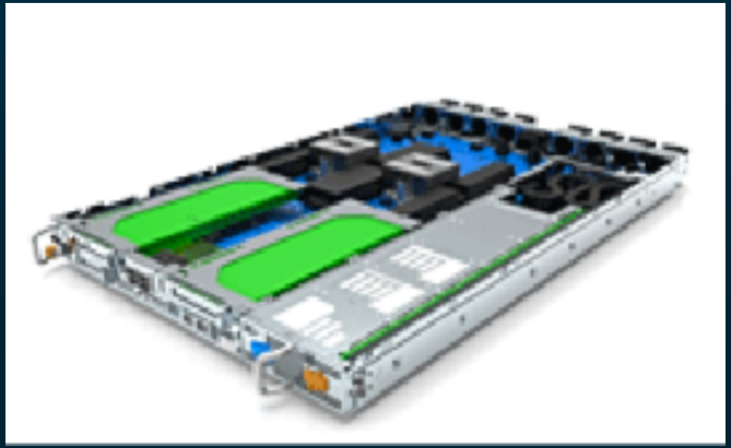
GPUs Bottlenecked by PCIe Bandwidth From CPU-System Memory

NVLink Enables Fast Unified Memory Access between CPU & GPU Memories

POWER8 with NVLink



- NVLink High Speed CPU <-> GPU Interconnect
- 160+ GigaBytes per second bi-directional
- 5-12x faster than PCIe Gen3 x16
- Nvlink Accelerator Lab - accellab@us.ibm.com



Zaius

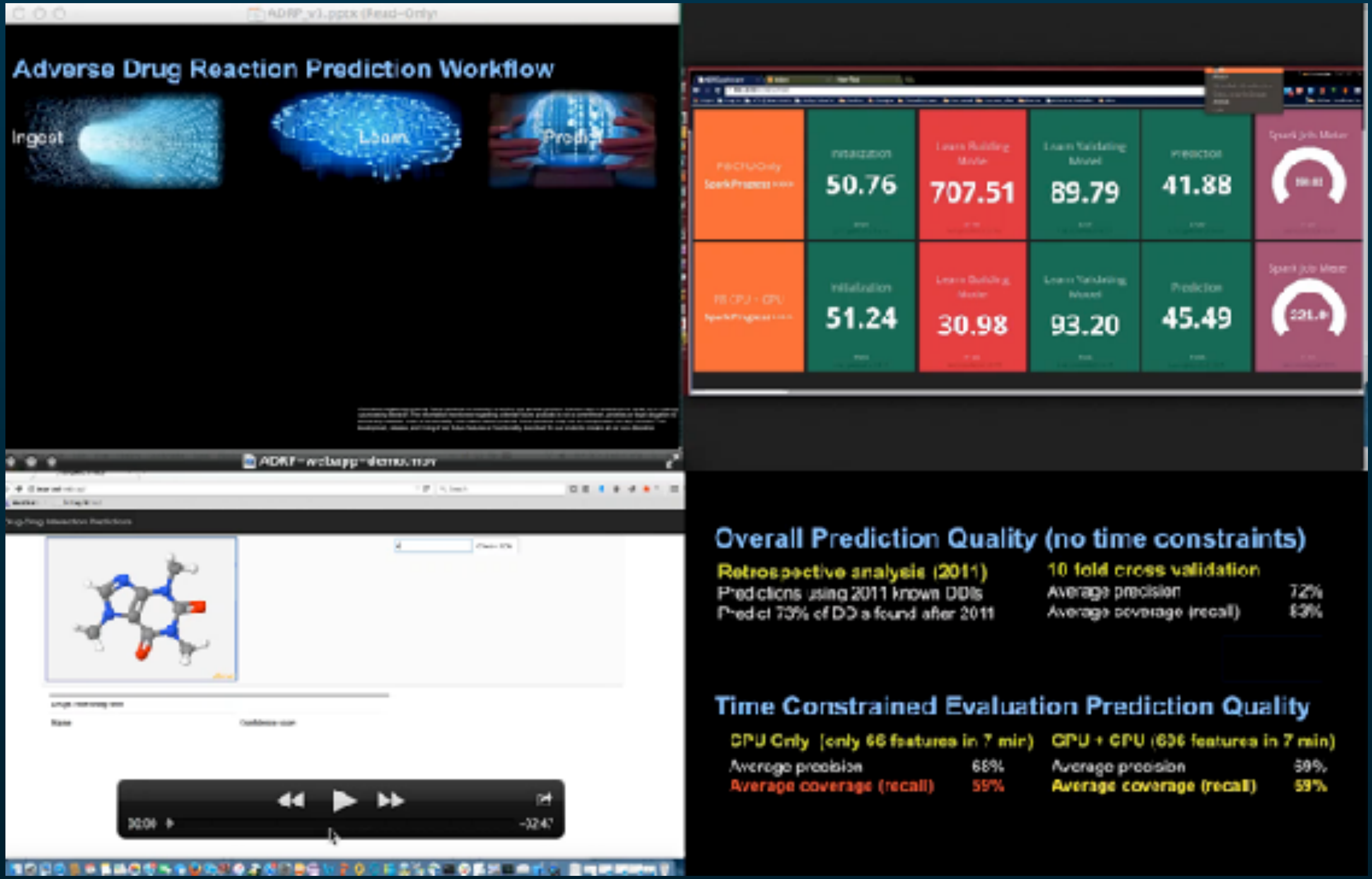
Google and Rackspace P9 server

Zoom Inventec

NVLink POWER Systems

Demo

GPU performance Demo



Acknowledgements

- India Team
  - Shreeharsha GN/India/IBM
  - Anjil R Chinnapatlolla/India/IBM

- Power OPEN Source and Solutions Development
  - Amir Sanjar /Austin/IBM

- Toronto Team
  - Gang L Liu/Toronto/IBM
  - Charlie Wang/Toronto/IBM
  - Zi Yin/Toronto/IBM@IBMCA

- Austin and Poughkeepsie Team
  - Rajaram B Krishnamurthy/Poughkeepsie/IBM
  - Mahalaxmi Lakshminarayanan/Austin/IBM
  - Yves Serge Joseph/Austin/IBM

- Data and Analytics Performance Lab

- POWER Systems Performance Team

Q & A

Notices and Disclaimers

Information in these presentations (including information relating to products that have not yet been announced by IBM) has been reviewed for accuracy as of the date of initial publication and could include unintentional technical or typographical errors. IBM shall have no responsibility to update this information. THIS DOCUMENT IS DISTRIBUTED "AS IS" WITHOUT ANY WARRANTY, EITHER EXPRESS OR IMPLIED. IN NO EVENT SHALL IBM BE LIABLE FOR ANY DAMAGE ARISING FROM THE USE OF THIS INFORMATION, INCLUDING BUT NOT LIMITED TO, LOSS OF DATA, BUSINESS INTERRUPTION, LOSS OF PROFIT OR LOSS OF OPPORTUNITY. IBM products and services are warranted according to the terms and conditions of the agreements under which they are provided.

IBM products are manufactured from new parts or new and used parts. In some cases, a product may not be new and may have been previously installed. Regardless, our warranty terms apply."

Any statements regarding IBM's future direction, intent or product plans are subject to change or withdrawal without notice.

Performance data contained herein was generally obtained in a controlled, isolated environments. Customer examples are presented as illustrations of how those customers have used IBM products and the results they may have achieved. Actual performance, cost, savings or other results in other operating environments may vary.

References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which IBM operates or does business.

Workshops, sessions and associated materials may have been prepared by independent session speakers, and do not necessarily reflect the views of IBM. All materials and discussions are provided for informational purposes only, and are neither intended to, nor shall constitute legal or other guidance or advice to any individual participant or their specific situation.

It is the customer's responsibility to insure its own compliance with legal requirements and to obtain advice of competent legal counsel as

to the identification and interpretation of any relevant laws and regulatory requirements that may affect the customer's business and any

actions the customer may need to take to comply with such laws. IBM does not provide legal advice or represent or warrant that its services

or products will ensure that the customer is in compliance with any law

Notices and Disclaimers Con't.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products in connection with this publication and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products. IBM does not warrant the quality of any third-party products, or the ability of any such third-party products to interoperate with IBM's products. IBM EXPRESSLY DISCLAIMS ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

The provision of the information contained h erein is not intended to, and does not, grant any right or license under any IBM patents, copyrights, trademarks or other intellectual property right.