FatMan vs. LittleBoy: Scaling up Linear Algebraic Operations in Scale-out Data Platforms

Luna Xu (Virginia Tech)

Seung-Hwan Lim (ORNL)

Ali R. Butt (Virginia Tech)

Sreenivas R. Sukumar (ORNL)

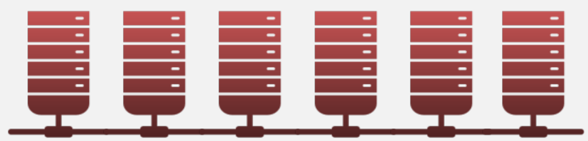
Ramakrishnan Kannan (ORNL)

HPC is used to enable scientific discovery

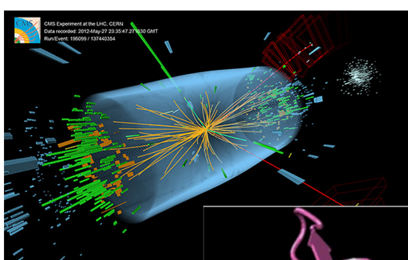
Scientific Simulation/Observation

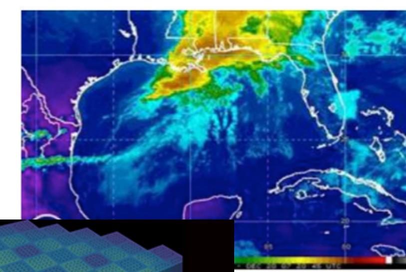
1100 1010 0101 | 1100 1010 0101 | 1100 1010 0101 | 1100 1010 0101 | 1100 1010 0101 | 1100 1010 0101 | 1100 1010 0101

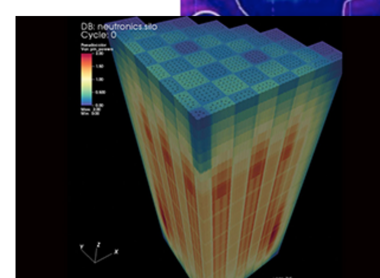
Data Analysis



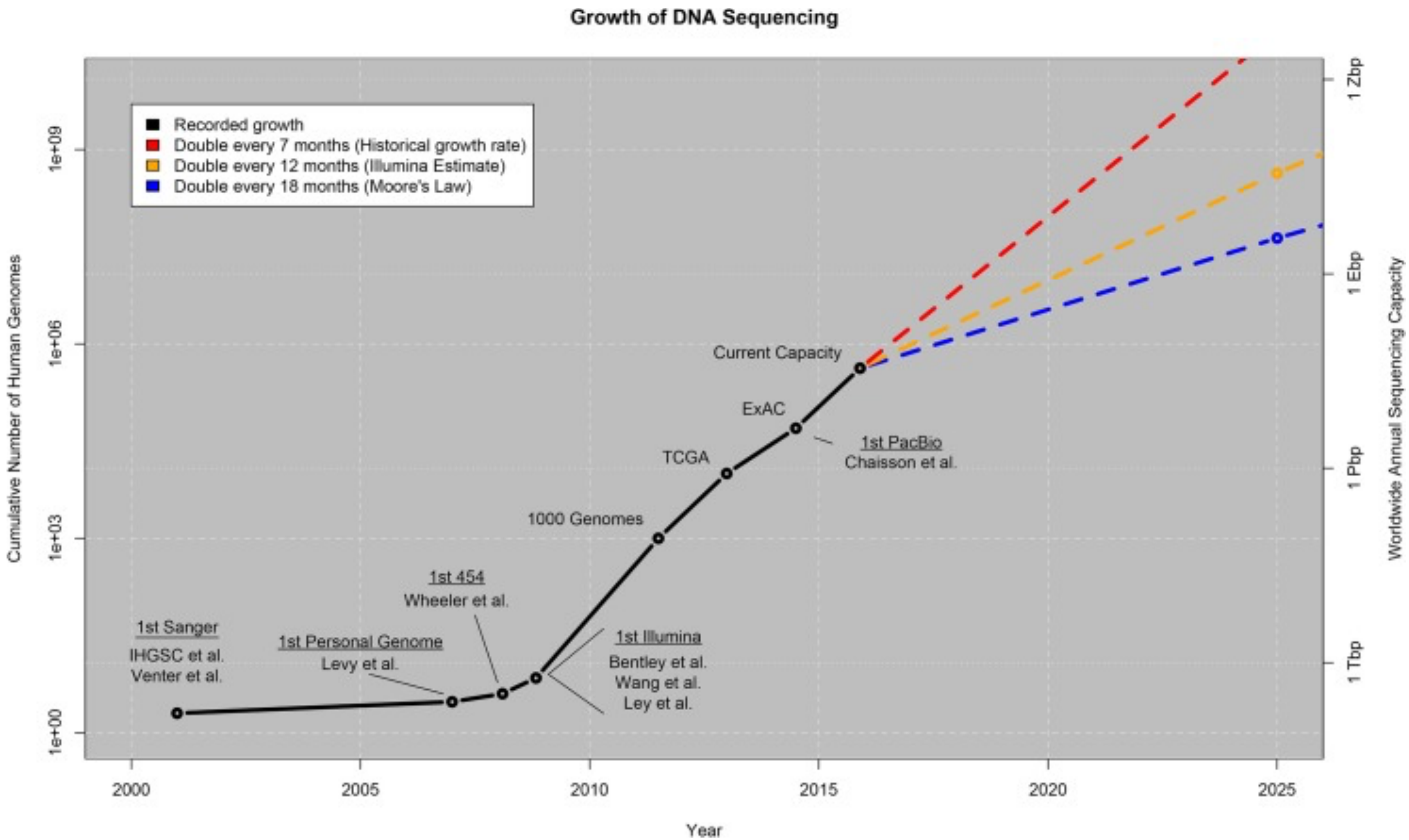
Scientific Discovery







Increasing role of data in HPC



Source: Stephens, Zachary D. et al. "Big Data: Astronomical or Genomical?" *PLoS Biology* 13.7 (2015) *PMC*. Web. 3 Nov. 2016.

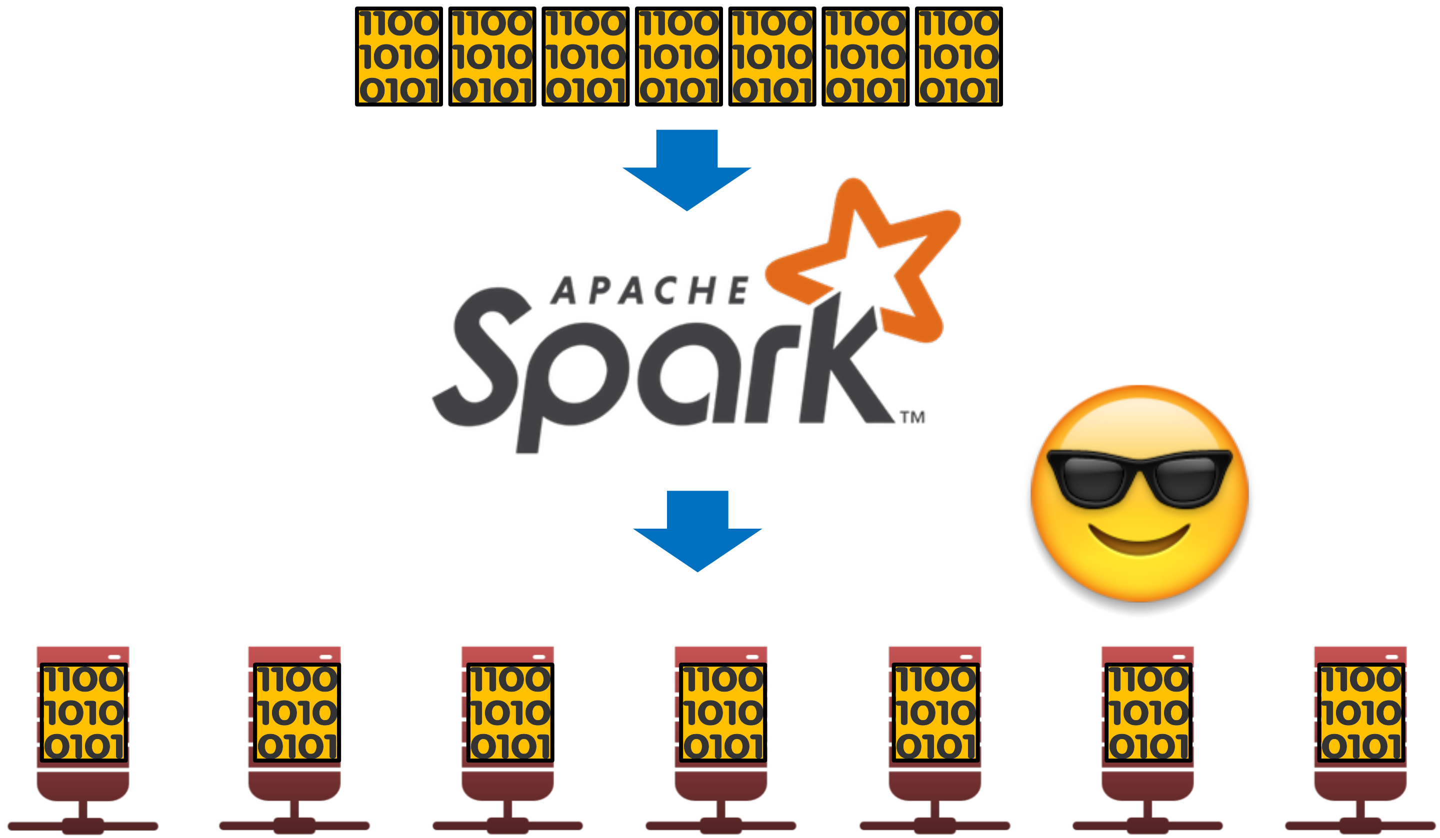
VirginiaTech
Invent the Future

Efficient big data processing → Faster HPC

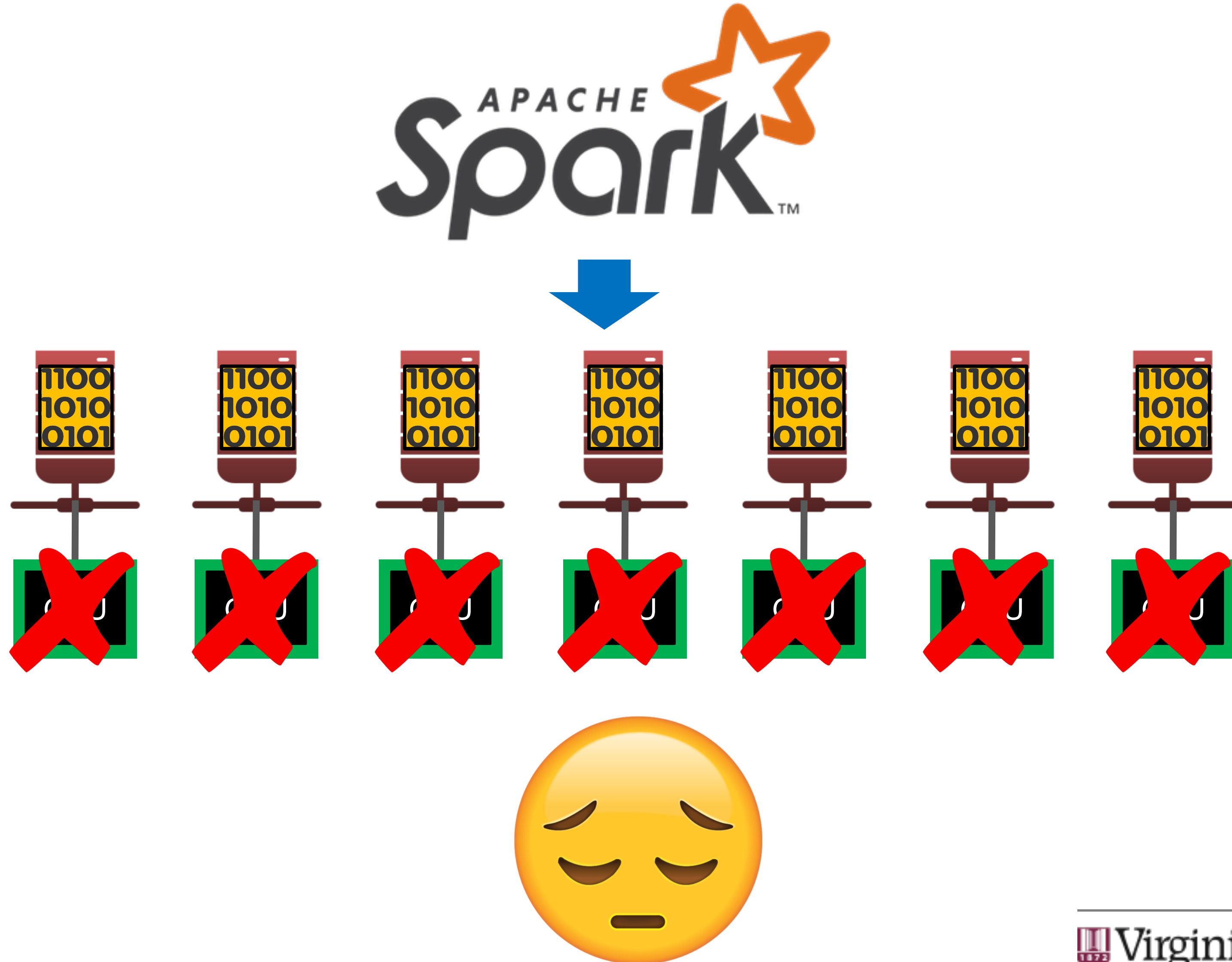
- Scale of data to be analyzed is growing exponentially

- **Observation:** Data analysis in HPC is similar to data analysis in big data community
- Big data scale-out platforms can benefit data-based scientific discovery

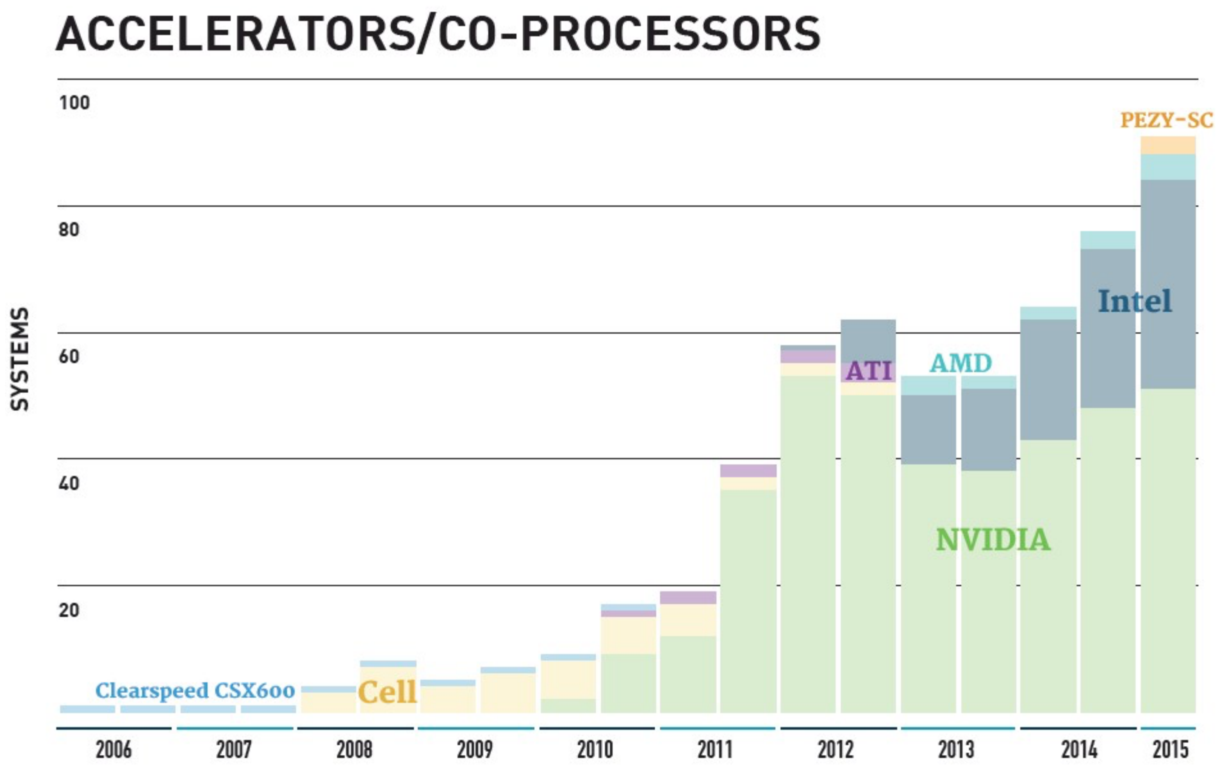
Scale-out data processing is becoming popular



Problem: Unable to leverage accelerators

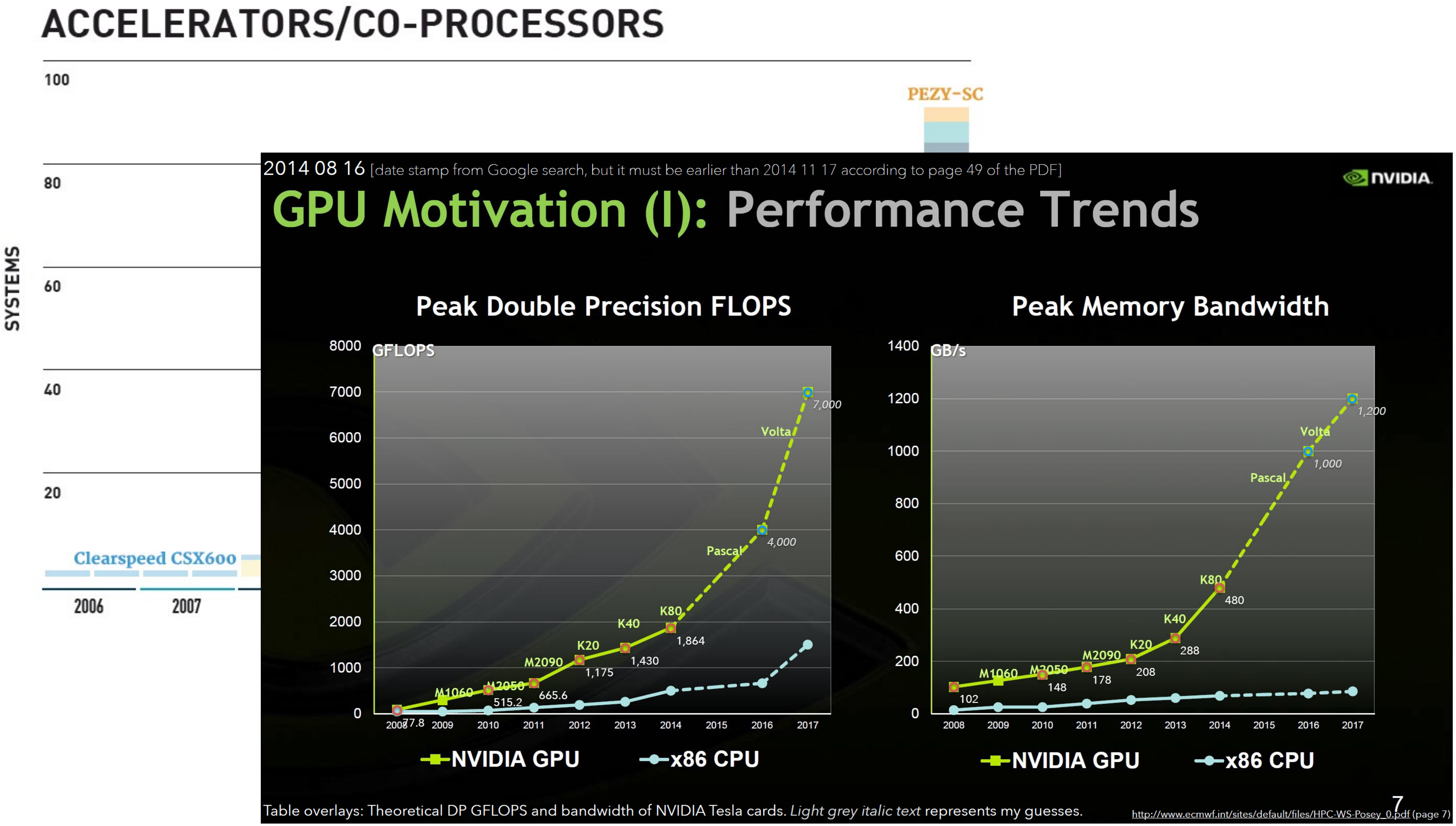


Per node performance does not scale



Sources: https://www.nextplatform.com/2015/07/13/top-500-supercomputer-list-reflects-shifting-state-of-global-hpc-trends/; Nvidia

Per node performance does not scale



Sources: https://www.nextplatform.com/2015/07/13/top-500-supercomputer-list-reflects-shifting-state-of-global-hpc-trends/; Nvidia

VirginiaTech Invent the Future

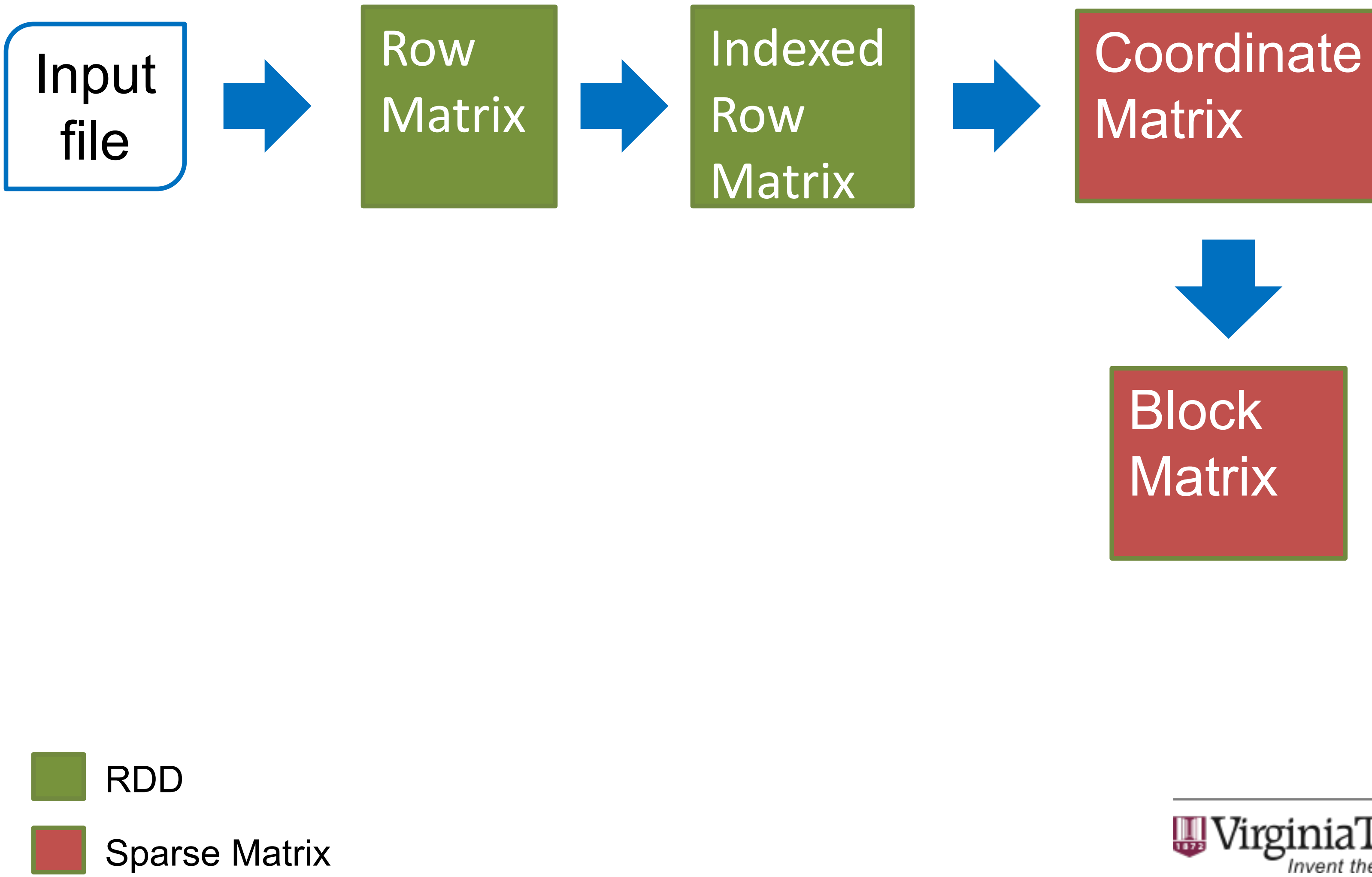
Our contribution: Scaling up linear algebra in scale-out data platforms

- Introduce the support for distributed dense matrix manipulations in Spark for scale-out matrix operations
- Adopt scale-up hardware acceleration for BLAS-3 operations of distributed matrices
- Design a flexible controller to decide when and whether to use hardware accelerators based on the density of matrices

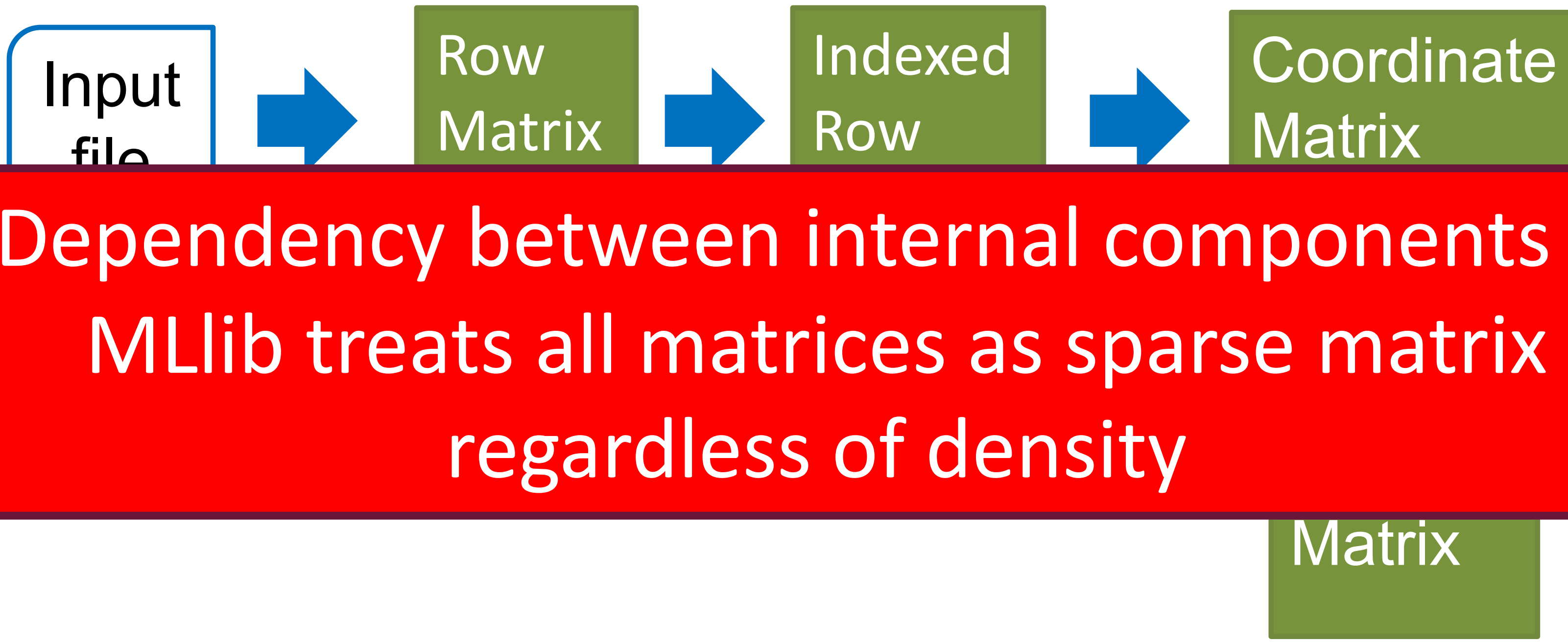
Agenda

- Introduction
- Background
- Design
- Evaluation
- Conclusion

Distributed matrix support in Spark



Distributed matrix support in Spark

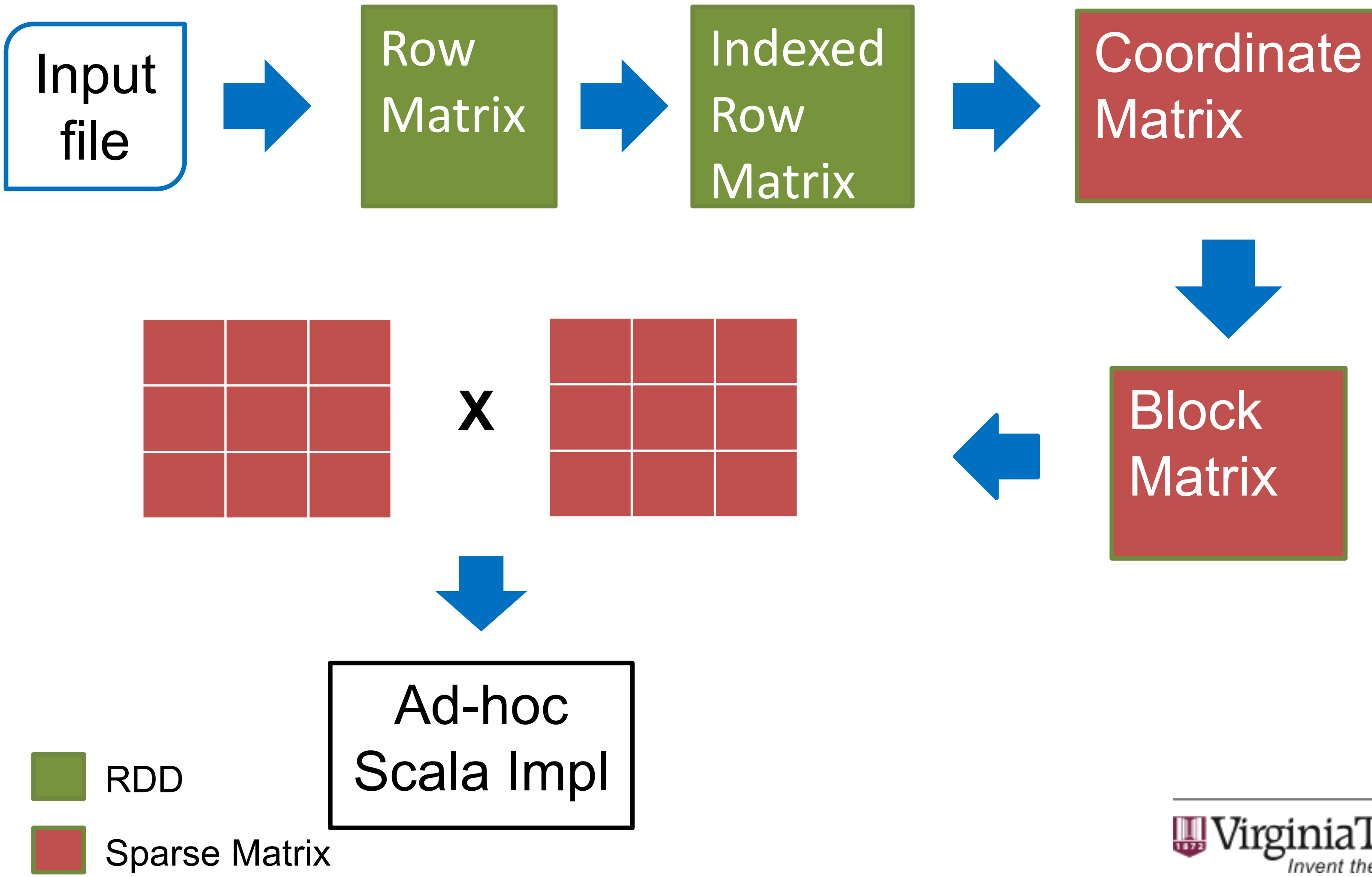


Dependency between internal components in MLlib treats all matrices as sparse matrix regardless of density

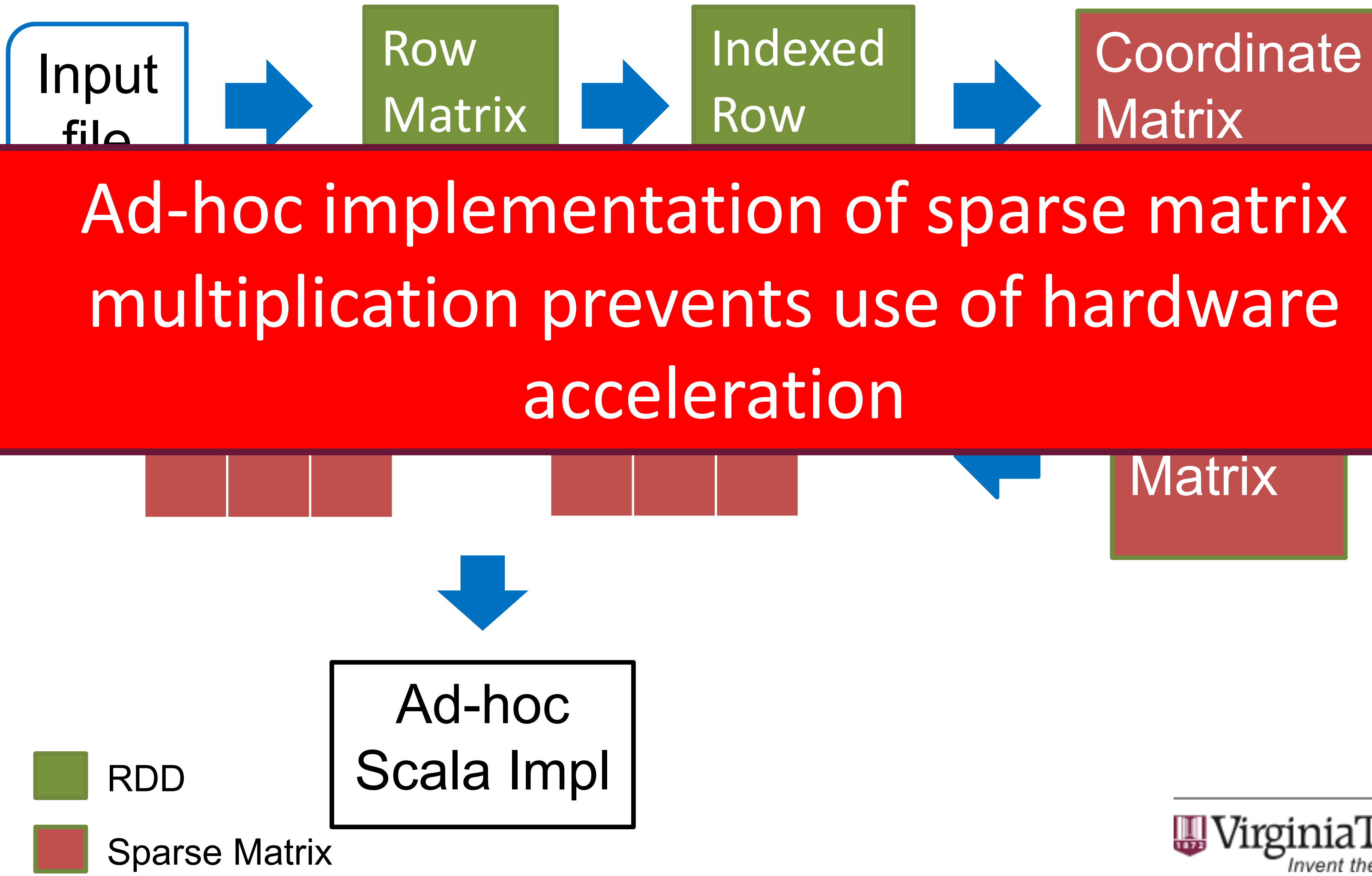
RDD

Sparse Matrix

Distributed matrix support in Spark



Distributed matrix support in Spark



Ad-hoc implementation of sparse matrix multiplication prevents use of hardware acceleration

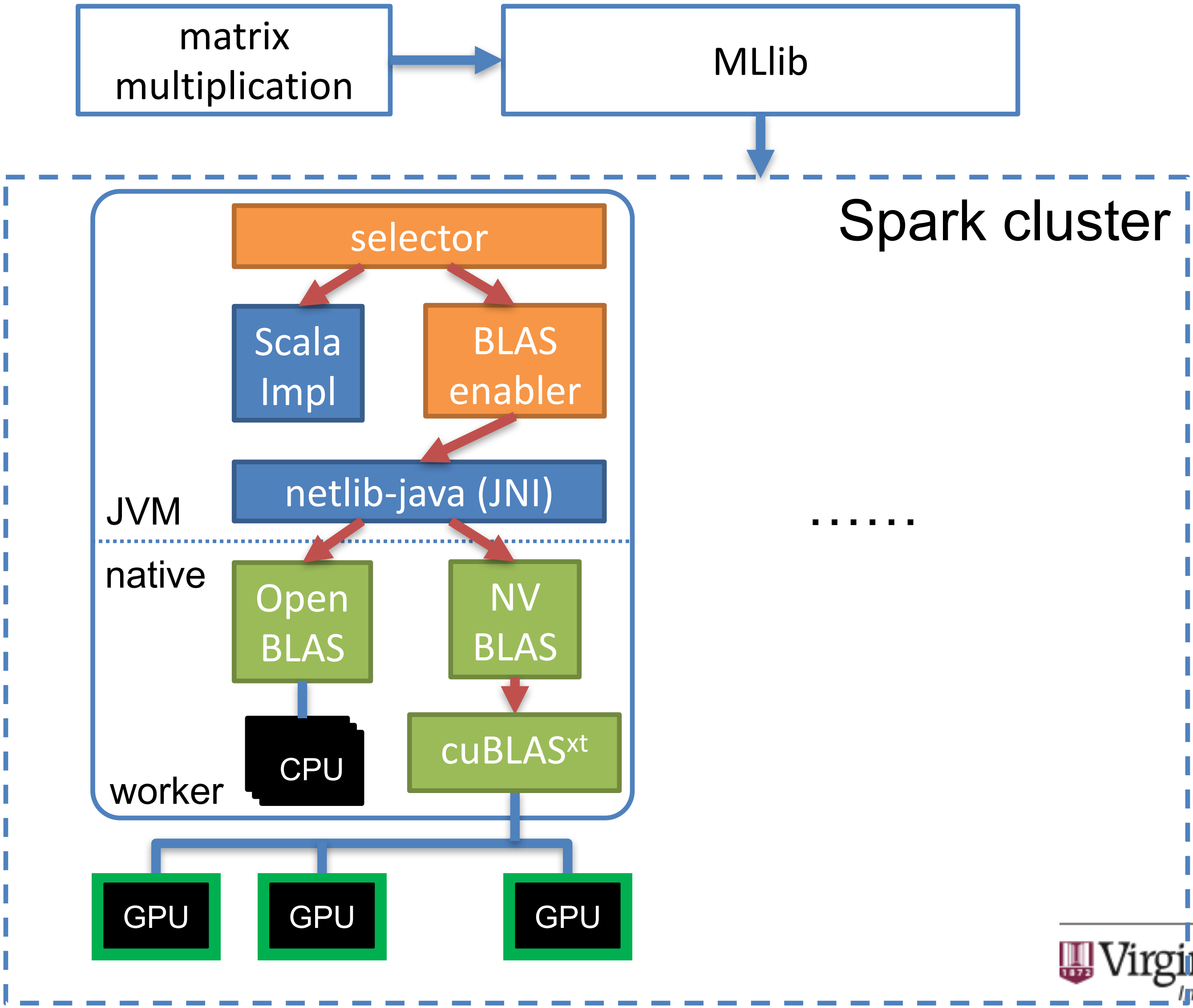
Agenda

- Introduction
- Background
- Design
- Evaluation
- Conclusion

Design considerations

- Enable user transparency
- Support scalable matrix multiplication
- Support dense and sparse matrices

System architecture



VirginiaTech
Invent the Future

Agenda

- Introduction
- Background
- Design
- Evaluation
- Conclusion

Methodology

Gramian matrix computation with GEMM

- XX^T
- Machine learning (PCA, SVD)
- Data analysis (all-pair similarity)

Methodology

Gramian matrix computation with GEMM

- XX^T

# of Rows (Cols)	Density	Raw size (GB)
4873	1	0.34
14684	1	3.1
24495	1	8.4
663331	1	77
4873	0.05	0.104
14684	0.05	2.6
24495	0.05	19
663331	0.05	41

Methodology

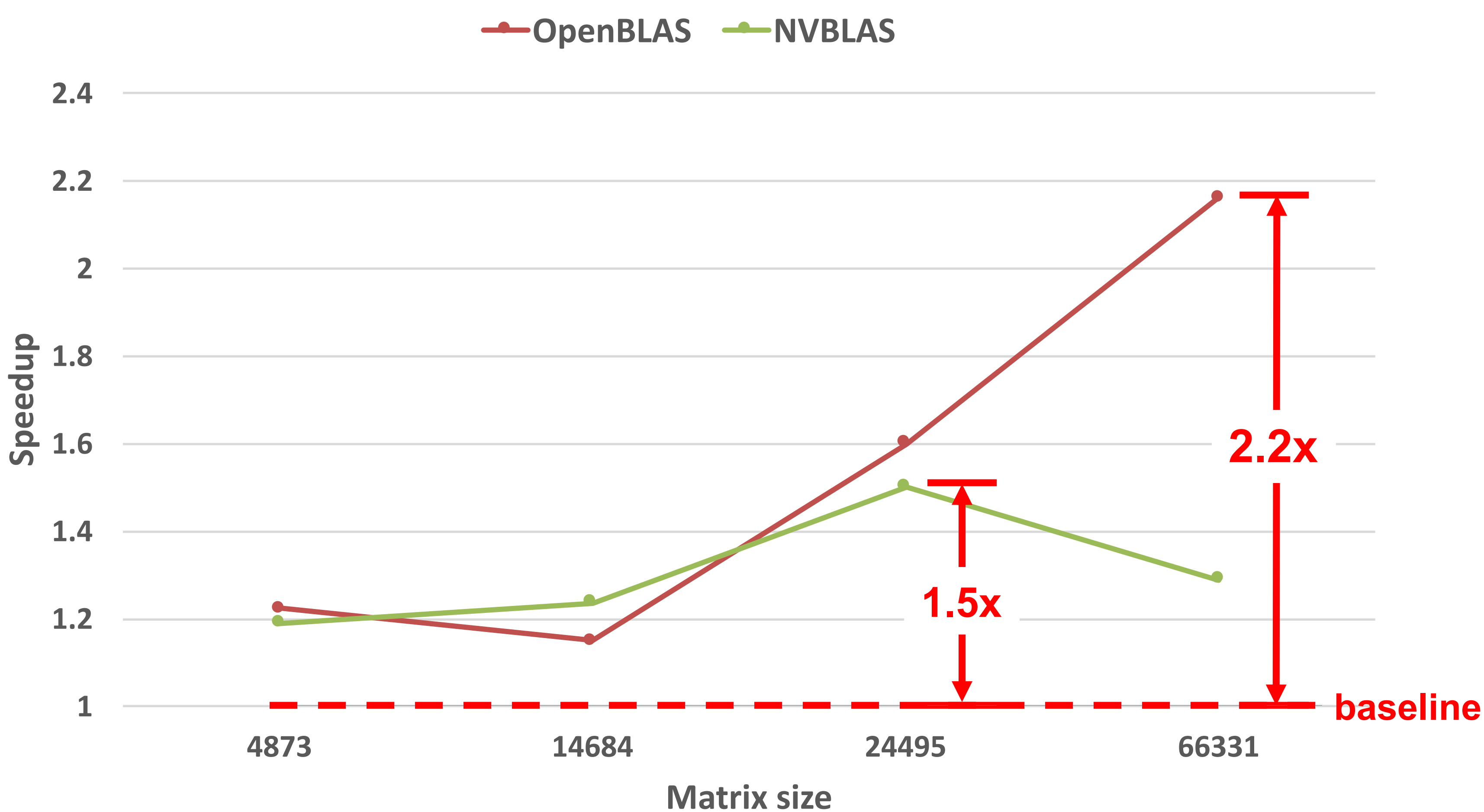
- System spec:

System	Rhea GPU node
CPU	Dual Intel Xeon E5
CPU cores	28 (56 HT)
Memory	1TB
GPU	Dual NVIDIA K80
GPU cores	4992 x 2
GPU memory	24 x 2 GB
CUDA	7.5

- Spark configuration:
  - Version: 1.6.1
  - 2 node cluster
  - One executor per node (56 cores, 800GB)
- BLAS configuration
  - OpenBLAS v0.2.19 (hand compile)
  - NVBLAS v7.5

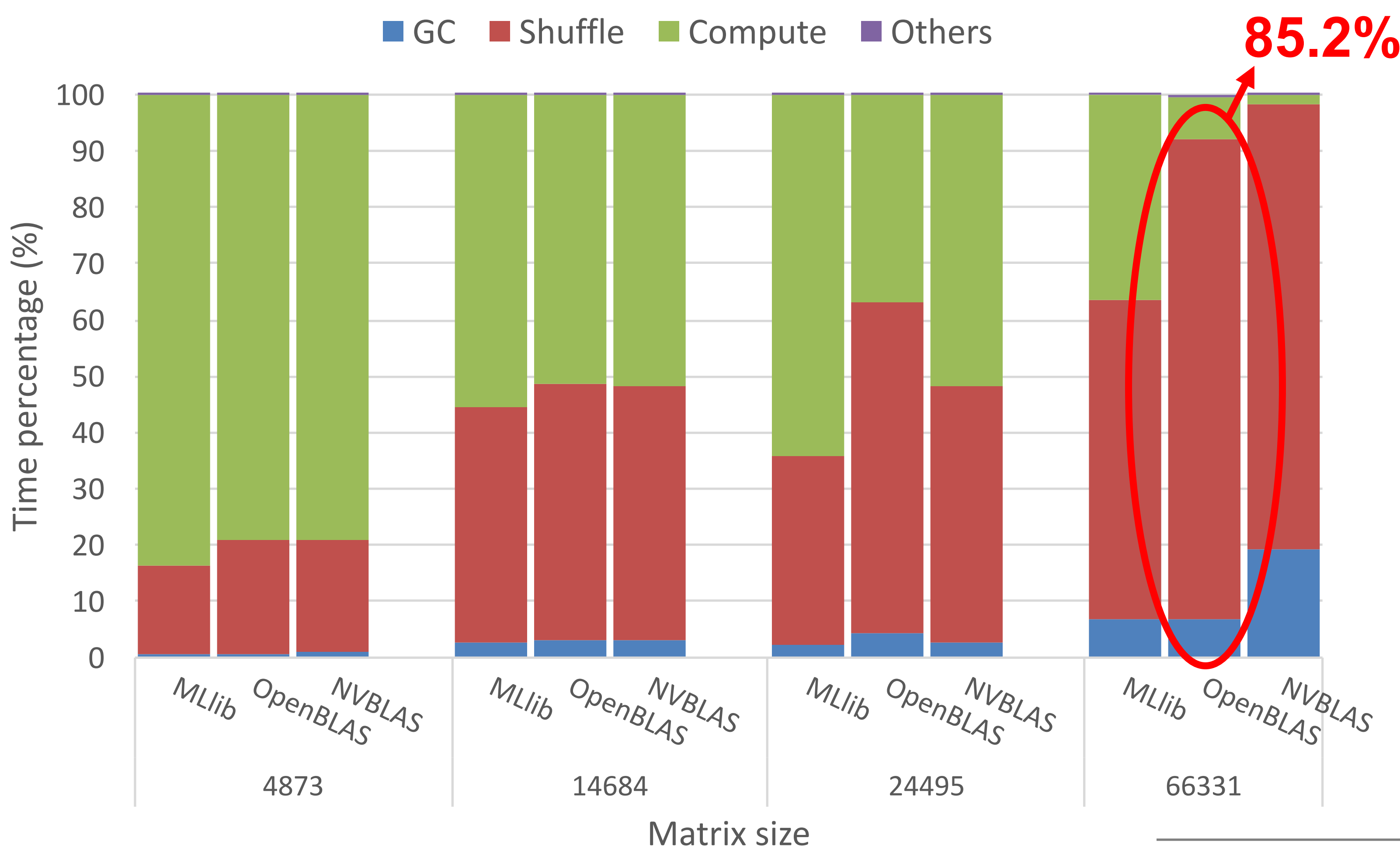
VirginiaTech Invent the Future

Overall performance: Dense Matrix



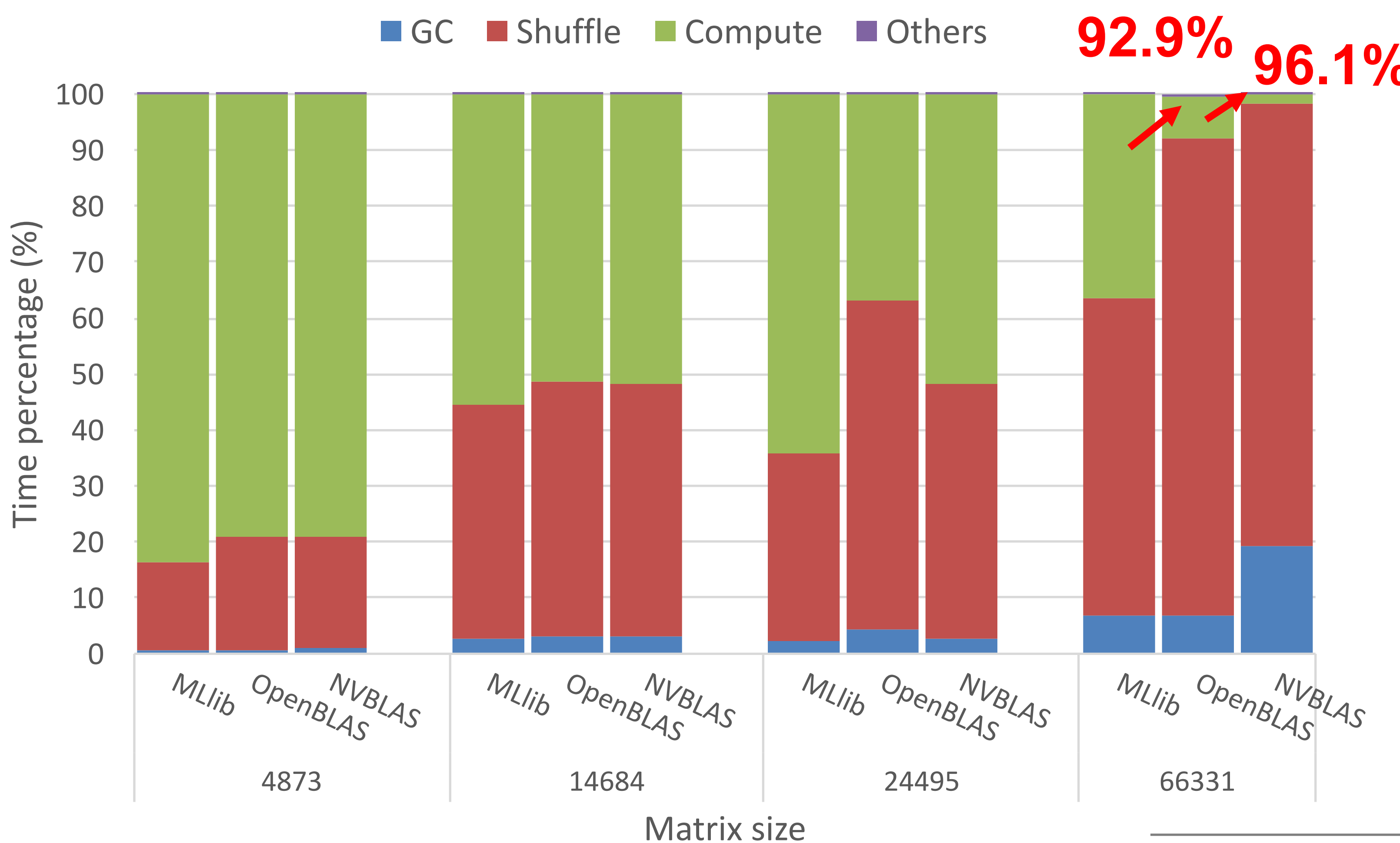
VirginiaTech Invent the Future

Performance: Dense Matrix



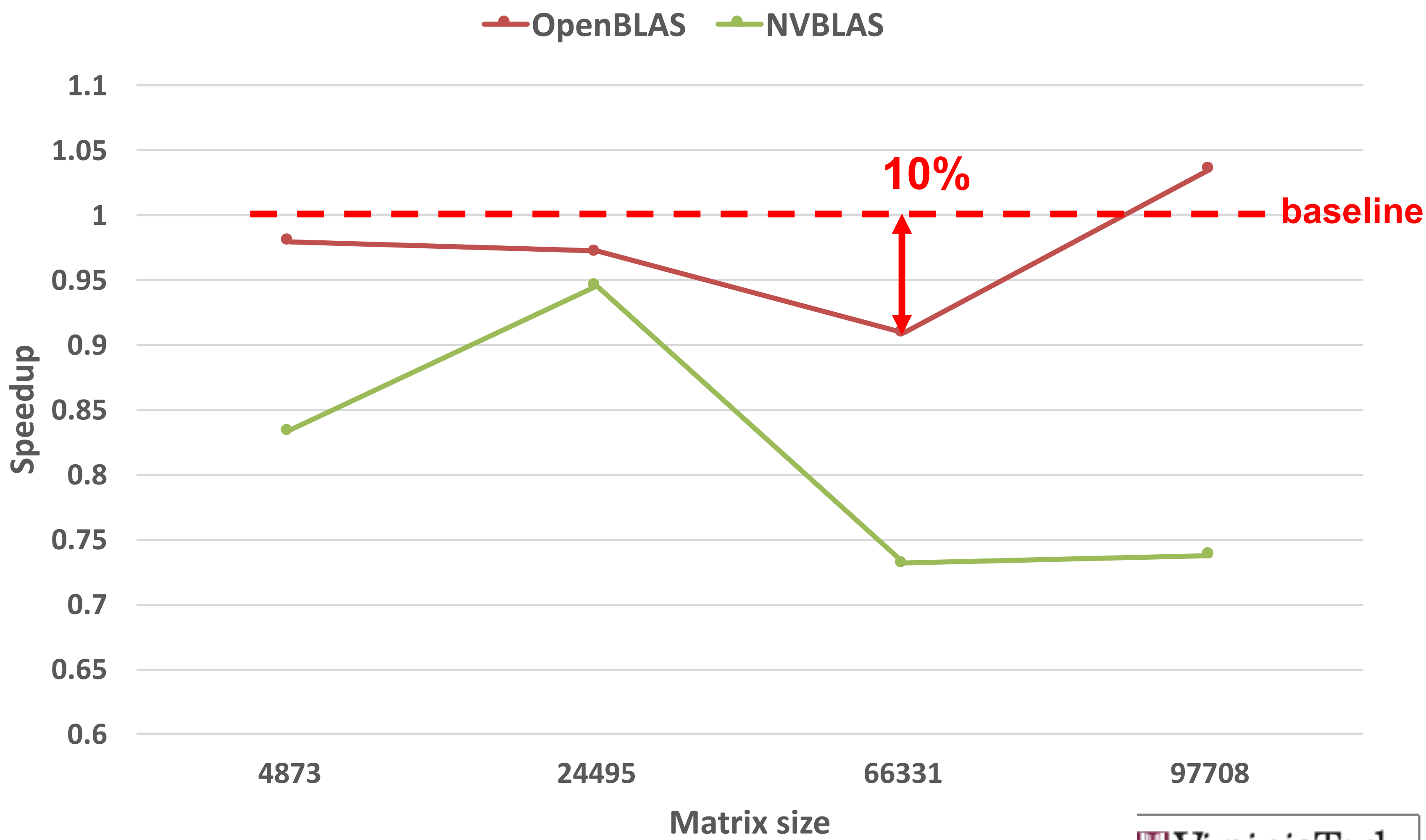
VirginiaTech Invent the Future

Performance: Dense Matrix

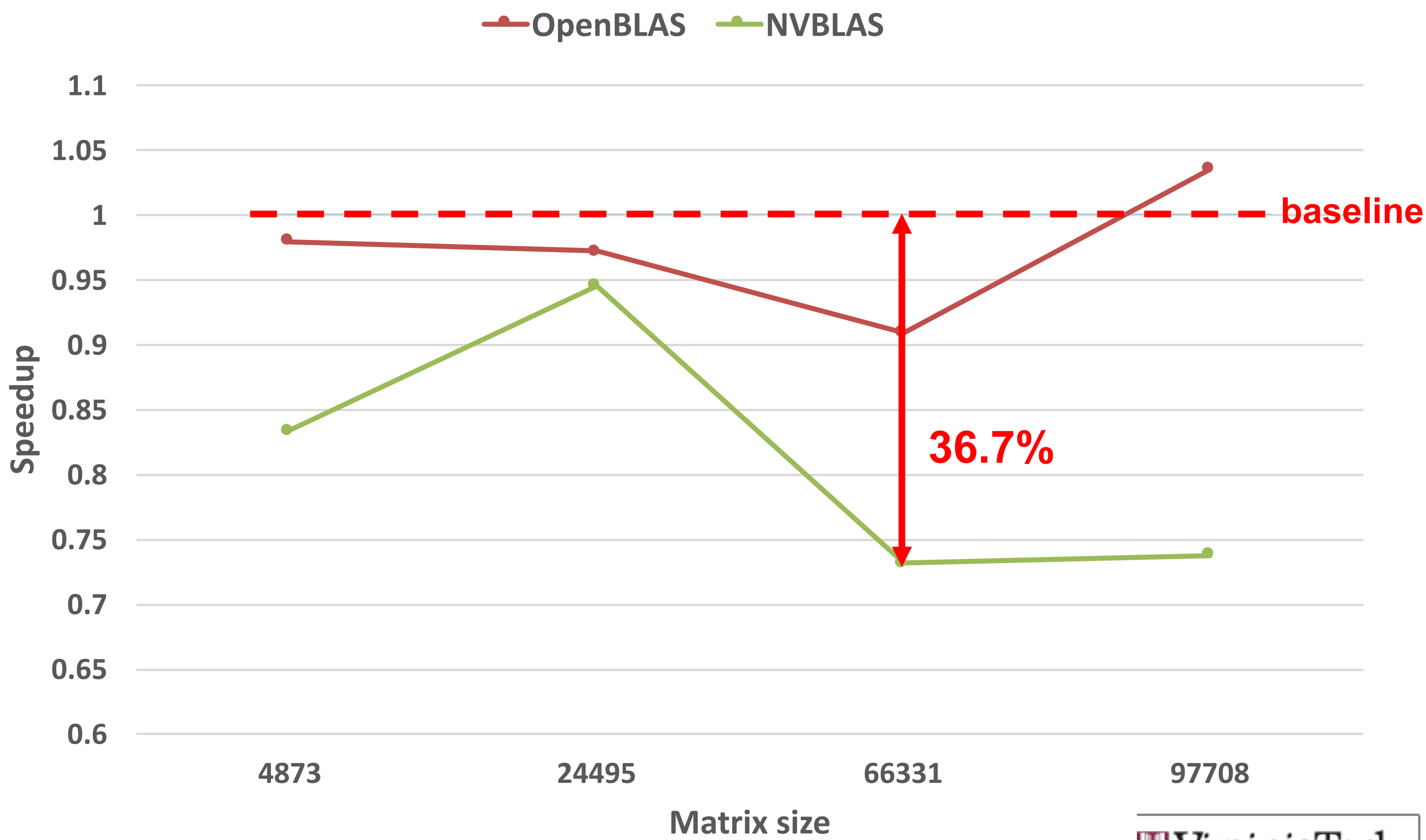


VirginiaTech Invent the Future

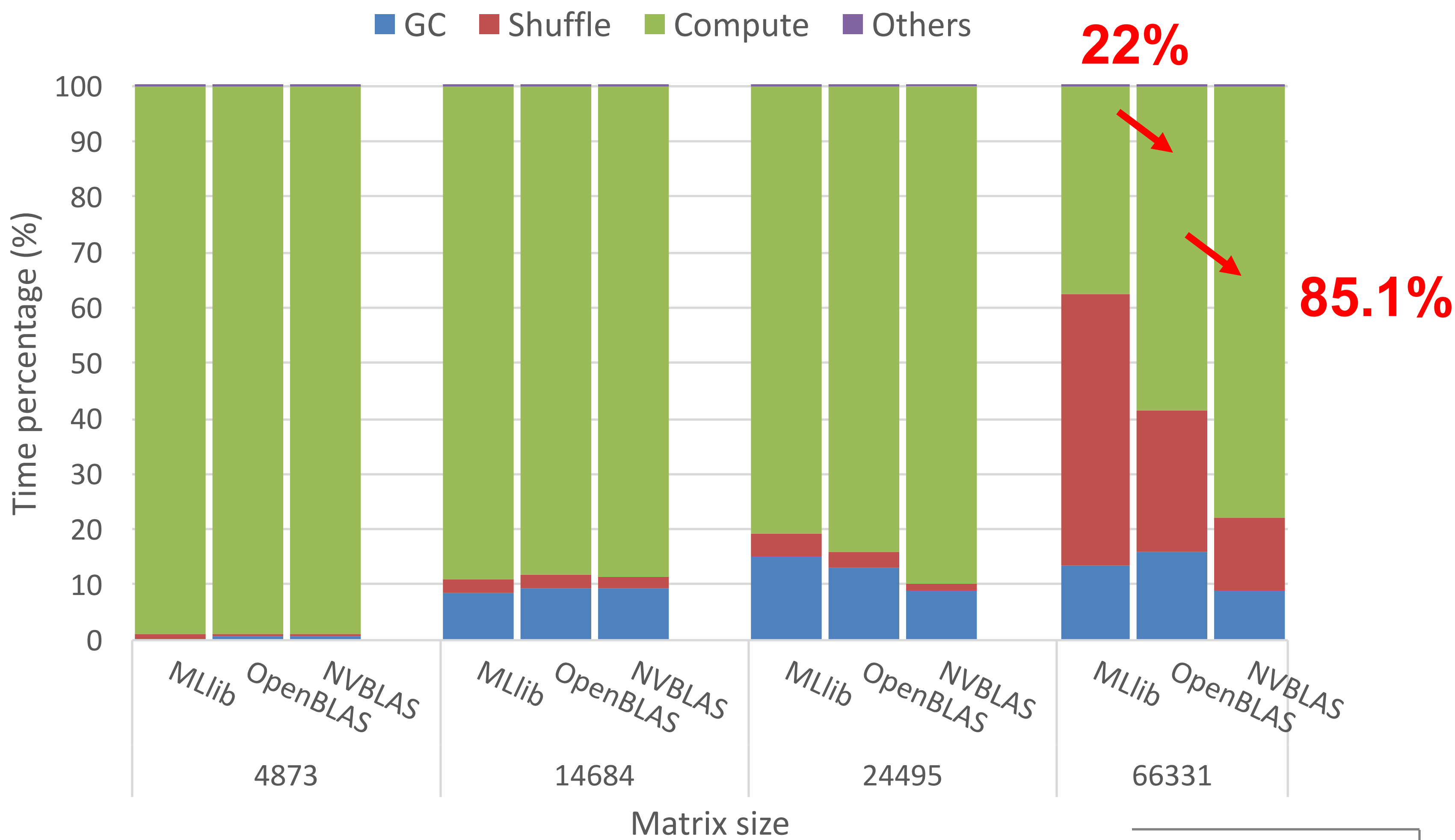
Overall performance: Sparse Matrix



Overall performance: Sparse Matrix



Performance: Sparse Matrix



Conclusion

- We employ scale-up accelerations for linear algebraic operations in Spark
- Our approach decides whether to use hardware accelerations based on matrix density
- The system improves overall performance up to more than 2x compared to default Spark
- Contact: Luna Xu, xuluna@cs.vt.edu
  DSSL: http://dssl.cs.vt.edu/

VirginiaTech
Invent the Future