A View on Many-Task Big Data Processing: from GPUs to Clouds

Ana Lucia Varbanescu
University of Amsterdam
The Netherlands

Alexandru Iosup
Delft University of Technology
The Netherlands

Our team: **Undergrad** Tim Hegeman, … **Grad** Yong Guo, Mihai Capota, Bogdan Ghit **Staff** Henk Sips, Dick Epema, **Collaborators*** Ted Willke (Intel), Claudio Martella (Giraph), Kefeng Deng (NUDT, CN), David Villegas (IBM), …

* Not their fault for any mistakes in this presentation. Or so they wish.

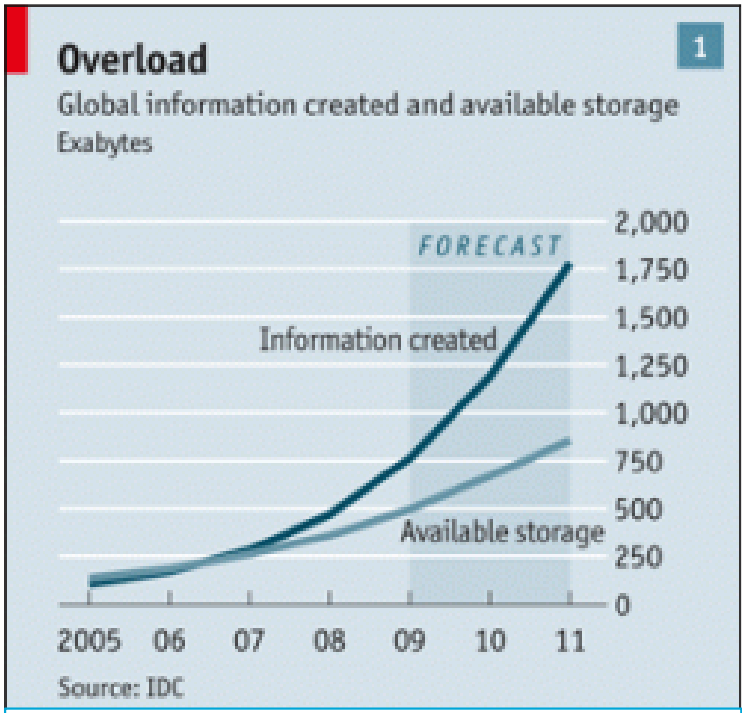
November 17, 2013

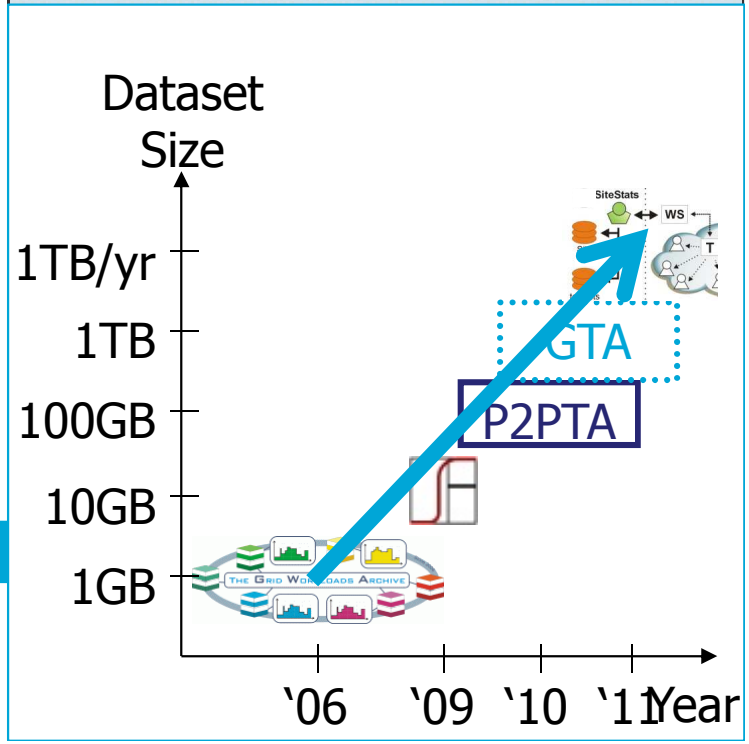
Technion, Haifa, Israel

TU Delft
Delft University of Technology

The Reality:
The Data Deluge

- **All human knowledge**
  - Until 2005: 150 Exa-Bytes
  - 2010: 1,200 Exa-Bytes
- **Online gaming (Consumer)**
  - 2002: 20TB/year/game
  - 2008: 1.4PB/year/game (only stats)
- **Public archives (Science)**
  - 2006: GBs/archive
  - 2011: TBs/year/archive





The Hype: The Three "V"s of Big Data
When you can, keep *and* process everything

- Volume
  - More data vs. better models
  - Data grows exponentially + iterative mod
  - Analysis in near-real time to extract value
  - Scalable storage and distributed queries
- Velocity
  - Speed of the feedback loop
  - Gain competitive advantage: fast recommendations
  - Identify fraud, predict customer churn faster
- Variety
  - The data can become messy: text, video, audio, etc.
  - Difficult to integrate into applications

Too big, too fast, does not comply with traditional DB

Adapted from: Doug Laney, "3D data management", META Group/Gartner report, Feb 2001. http://blogs.gartner.com/doug-laney/files/2012/01/ad949-3D-Data-Management-Controlling-Data-Volume-Velocity-and-Variety.pdf

TU Delft

The Reality: BTWorld, A Simple Use-Case

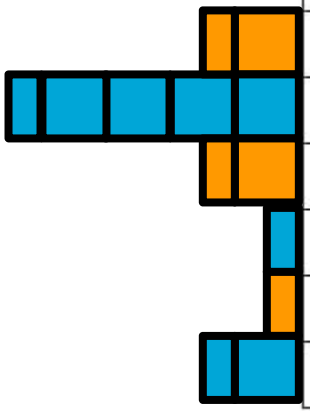
- BTWorld: a scientific project for collecting and **processing** time-based data
  - 3½ years of monitoring the BitTorrent network
  - Collected 14+ TB of data, 150bn records
- How to process such a dataset?

Smaller companies may also have problems with Big Data processing

The Science: Which Algorithms?

- (DONE) Our own survey, related to graph-processing
  - Academic publications (CIKM, ICDE, SIGKDD, SIGMOD, VLDB, CCGRID, HPDC, IPDPS, PPoPP, SC)

Graphitti



Class	Typical algorithms
General Statistics	Triangulation [36], Diameter [37], BC [38]
Graph Traversal	BFS, DFS, Shortest Path Search
Connected Components	MIS [39], BiCC [40], Reachability [41]
Community Detection	Clustering, Nearest Neighbor Search
Graph Evolution	Forest Fire Model [1], Preferential Attachment Model [42]
Other	Sampling, Partitioning

- (Ongoing) Our practitioner-scientist survey

http://goo.gl/TJwkTg

Ad: Help us gain this knowledge

2012-2013

Guo, Biczak, Varbanescu, Iosup, Martella, Wi
How Well do Graph-Processing Platforms Per
An Empirical Performance Evaluation and An

http://bit.ly/10hYdIU

The Science: Dataset sizes? Machines in cluster?

- (DONE) Our own survey, related to graph-processing
  - Academic publications (CIKM, ICDE, SIGKDD, SIGMOD, VLDB, CCGRID, HPDC, IPDPS, PPoPP, SC)

Graphitti

Platforms	Algorithms	Dataset type	Largest dataset	System
Neo4j, MySQL [40]	1 other	synthetic	100 KV	1 C
Neo4j, etc. [4]	3 others	synthetic	1 MV	1 C
Pregel [5]	1 other	synthetic	50 BV	300 C
GPS, Giraph [41]	CONN, 3 others	real	39 MV, 1.5 BE	60 C
			1 BV	16 C
			282 MV	90 C

Dataset size: 100sMB—10s GB

System size: <10—100s nodes

- (Ongoing) Our practitioner-scientist survey

http://goo.gl/TJwkTg

Ad: Help us gain this knowledge

2012-2013

Guo, Biczak, Varbanescu, Iosup, Martella, Wi
How Well do Graph-Processing Platforms Per
An Empirical Performance Evaluation and An

http://bit.ly/10hYdIU

The Data Deluge
The Professional World Gets Connected

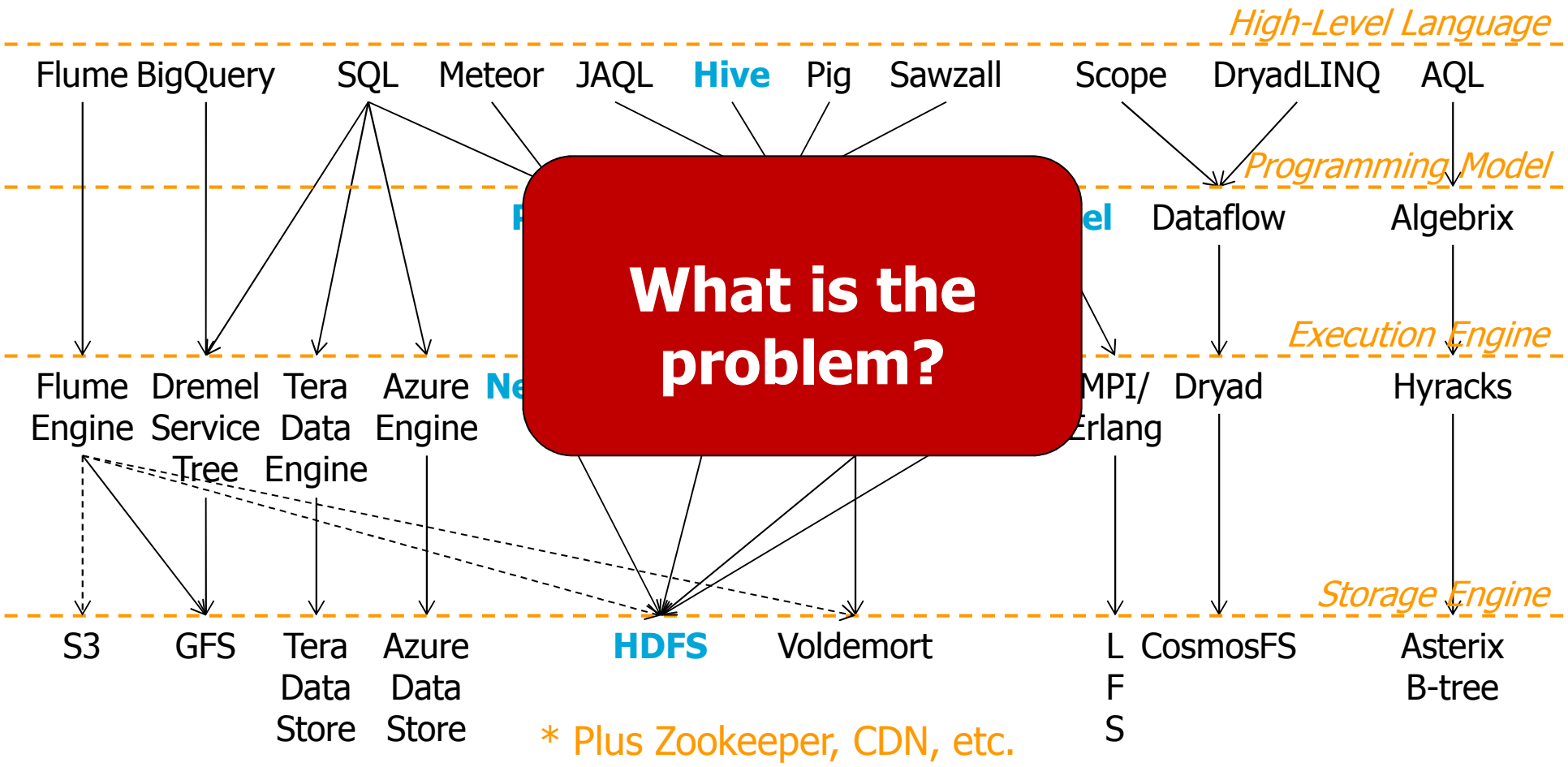
Source: Vincenzo Cosenza, The State of LinkedIn, http://vincos.it/the-state-of-linkedin/

TU Delft

Agenda

1. Introduction to Big Data
2. **Programming Models for Big Data**
3. Towards a General Many-Task Big-Data Architecture
4. Summary

Programming Models for Big Data: Systems of Systems (Why Big Data is Difficult)

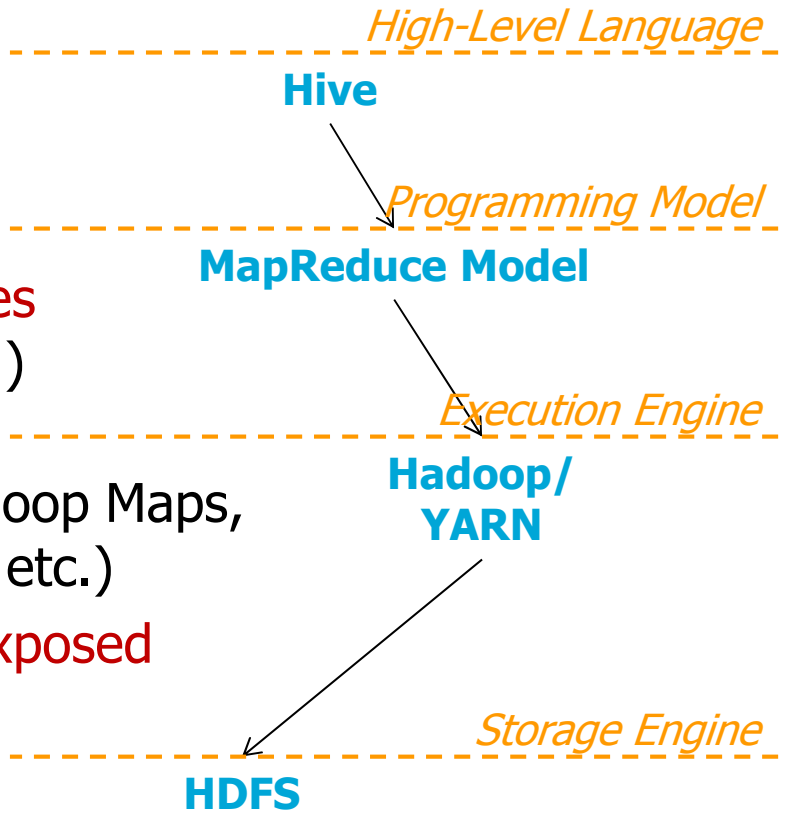


Adapted from: Dagstuhl Seminar on Information Management in the Cloud, http://www.dagstuhl.de/program/calendar/partlist/?semnr=11321&SUOG

TU Delft

The Problem: Monolithic Systems

- Monolithic
  - Highly integrated stack
    (we forgot 6 decades of sw.eng.)
  - Fixed set of homogeneous resources
    (we forgot 2 decades of distrib.sys.)
  - Execution engines do not coexist
    (we're running now MPI inside Hadoop Maps, Hadoop jobs inside MPI processes, etc.)
  - Little performance information is exposed
    (we forgot 4 decades of par.sys.)
  - …

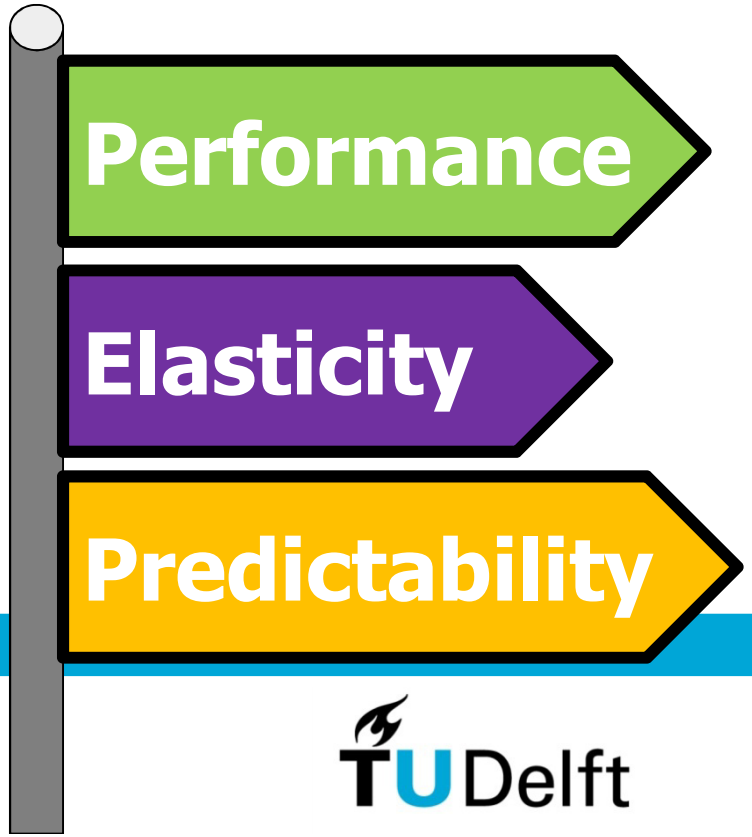


Pick your stack and you're stuck! (kid-level rhyme)

Adapted from: Dagstuhl Seminar on Information Management in the Cloud,
http://www.dagstuhl.de/program/calendar/partlist/?semnr=11321&SUOG

Agenda

1. Introduction
2. Programming Models for Big Data
3. **Big Data on GPUs and Clouds**
   1. **A General Approach**
   2. Performance
   3. Elasticity
   4. Predictability
4. Summary

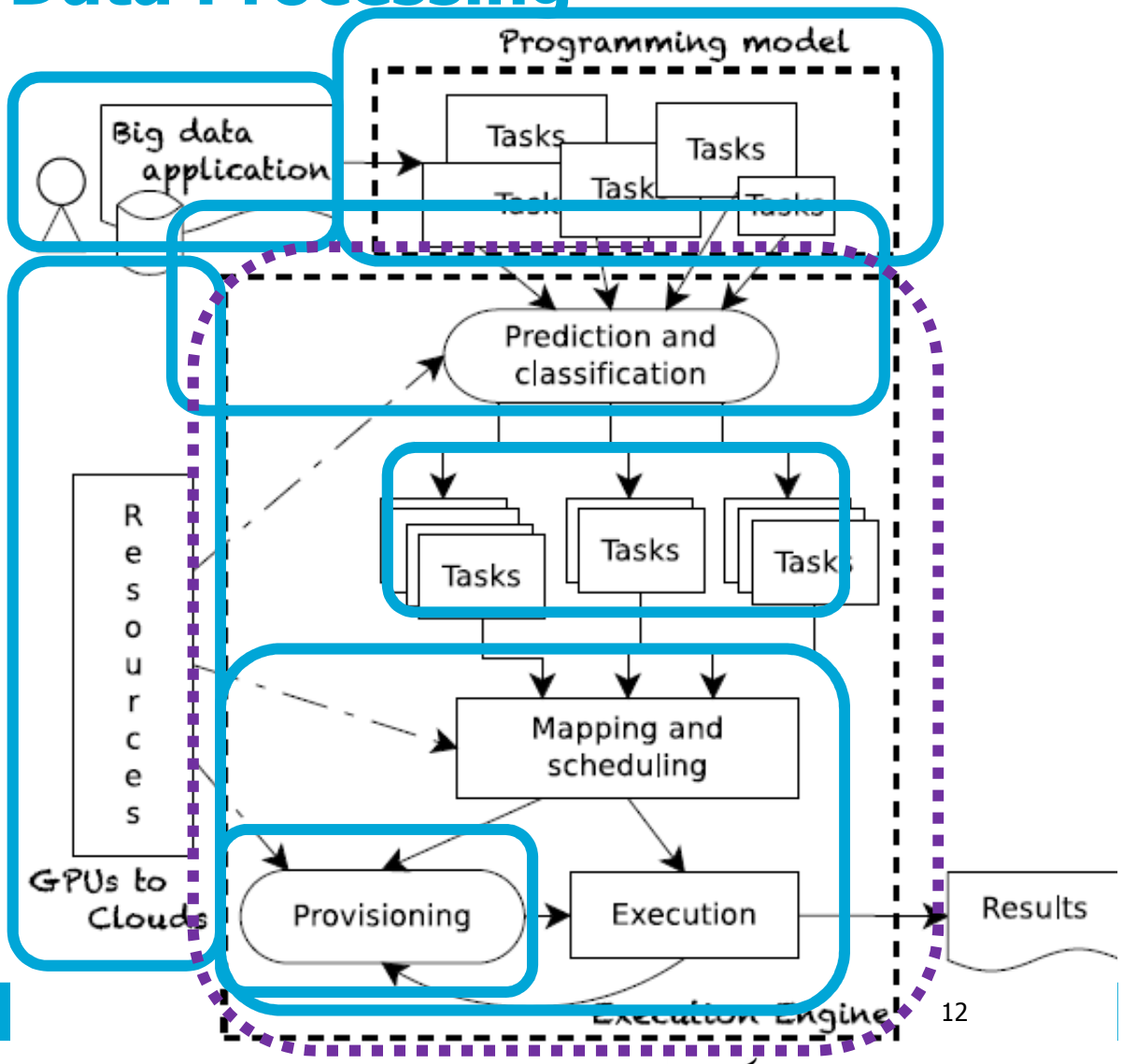


TU Delft

A Generic Architecture for Many-Task Big Data Processing

Execute Big Data apps as many tasks using mixed resouces:

1. High performance
2. Elasticity
3. Predictability
4. Compatibility



TU Delft

10 Main Challenges in 4 Categories*

* List not exhaustive

High Performance

1. **Parallel architectures and algorithms**—support from start
2. **Heterogeneous platforms**—application and data decomposition
3. Programmability by portability

Elasticity

1. **Performance and cost-awareness under elasticity**
2. **Portfolio scheduling**
3. Social awareness

Predictability

1. **Modeling**
2. **Benchmarking**

Compatibility

1. Interfacing with the application
2. Storage management

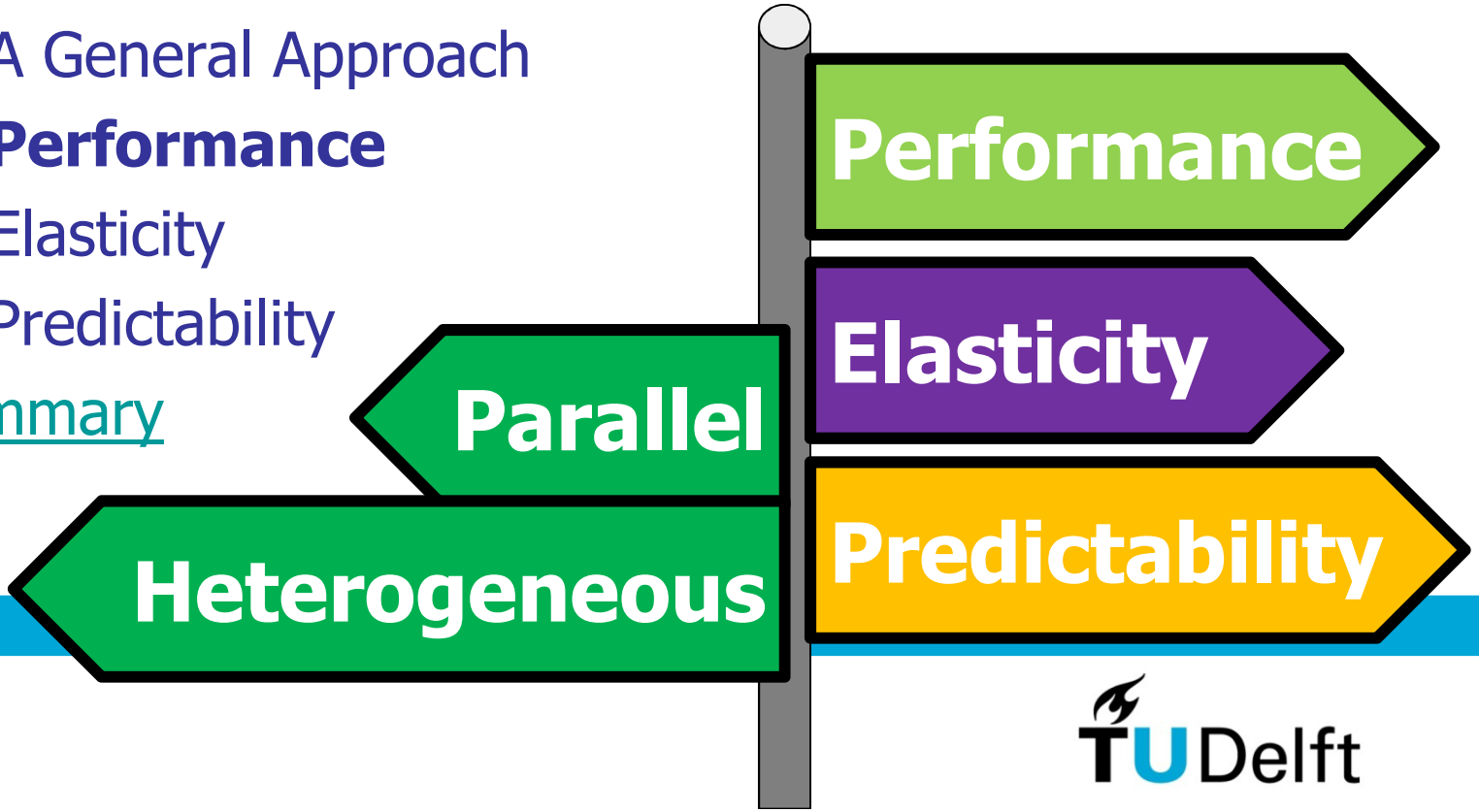
Varbanescu and Iosup, On Many-Task Big Data Processing: from GPUs to Clouds, MTAGS 2013. Proc. of SC13. (invited paper)

Ad: Read our article

TU Delft

Agenda

1. Introduction
2. Programming Models for Big Data
3. **Big Data on GPUs and Clouds**
   1. A General Approach
   2. **Performance**
   3. Elasticity
   4. Predictability
4. Summary

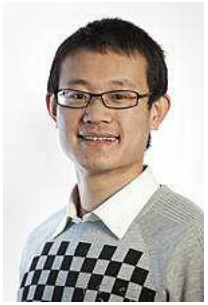


TU Delft

Performance: Our Team

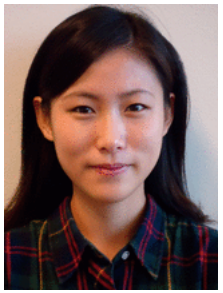
Ana Lucia Varbanescu
U. Amsterdam

Performance modeling
Parallel systems
Multi-core systems



Jianbin Fang
TU Delft

Parallel systems
Multi-core systems
Tianhe/Xeon Phi



Jie Shen
TU Delft

Performance evaluation
Parallel systems
Multi-core systems

Alexandru Iosup
TU Delft

Performance modeling
Performance evauluation

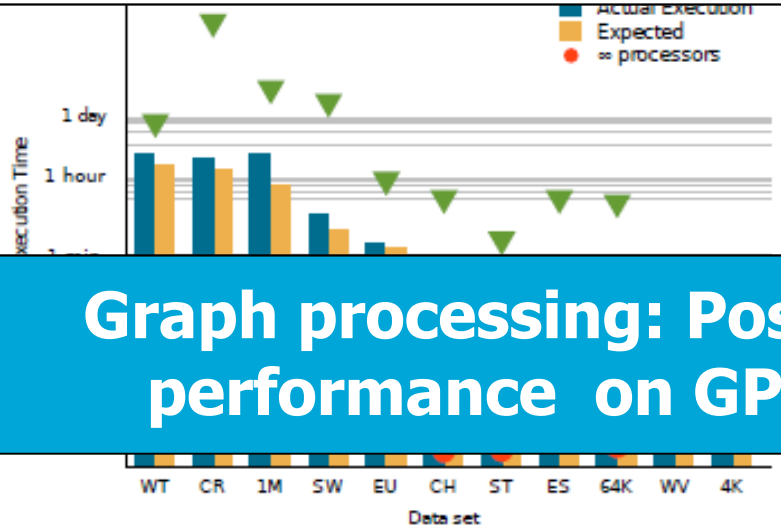
TU Delft

Parallel Architectures and Algorithms

- Unprecedented parallelism
- Instead of first large, sequential code-base, and only then parallelization
- Design and implement novel and efficient parallel algorithms from the beginning …
- And take into account many-task programming model

GPUs vs CPUs: All-Pairs Shortest Path

Pender and Varbanescu. MSc thesis at TU Delft. Jun 2012. TU Delft Library, http://library.tudelft.nl .

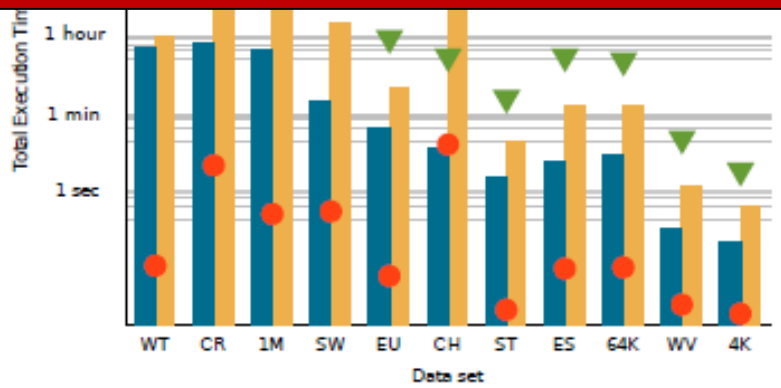


	Dataset
WT	Wikipedia Talk Network
CR	California Road Network
1M	Graph 1M
WV	Wikipedia Vote
4K	Graph 4K

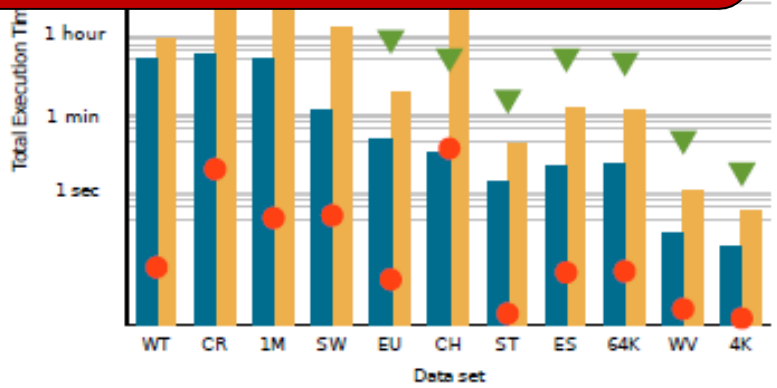
(a) Intel Xeon E5620

Graph processing: Possible to get better performance on GPUs than on CPUs

However, Algorithm and Dataset also determine performance



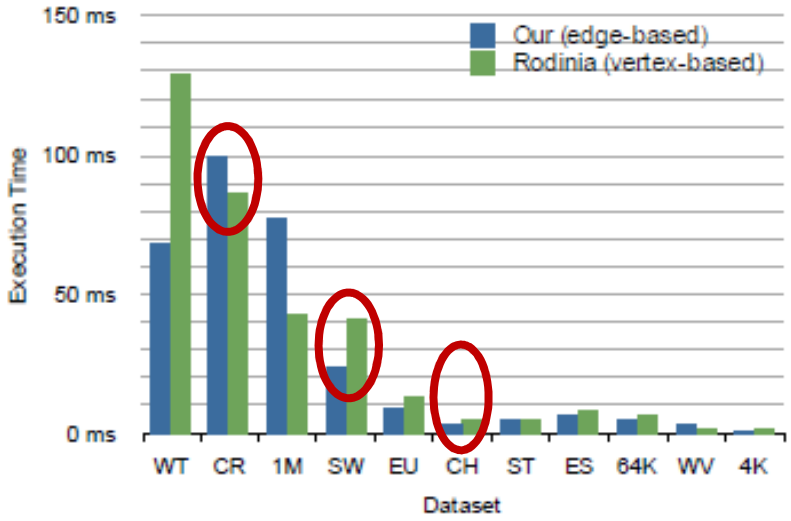
(c) Nvidia Tesla C2050/ C2070



(d) Nvidia GeForce GTX480

GPUs vs CPUs: BFS vs Data Format, E/V-based

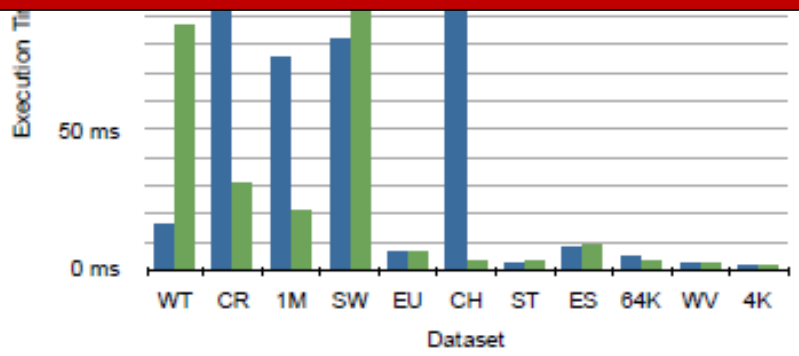
Pender and Varbanescu. MSc thesis at TU Delft. Jun 2012. TU Delft Library, http://library.tudelft.nl .



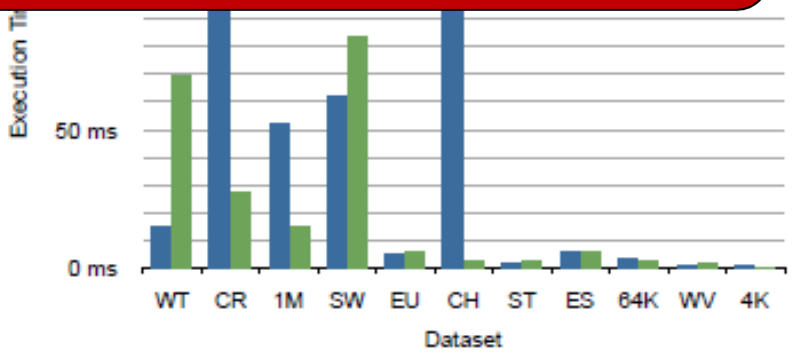
(a) Intel Xeon E5620

	Dataset
WT	Wikipedia Talk Network
CR	California Road Network
1M	Graph 1M
SW	Stanford Web Graph
EU	EU Email Communication Network
CH	Chain 100K
ST	Star 100K
ES	Epinions Social Network
64K	Graph 64K
WV	Wikipedia Vote
4K	Graph 4K

However, data format can also determine performance



(c) Nvidia Tesla C2050/ C2070

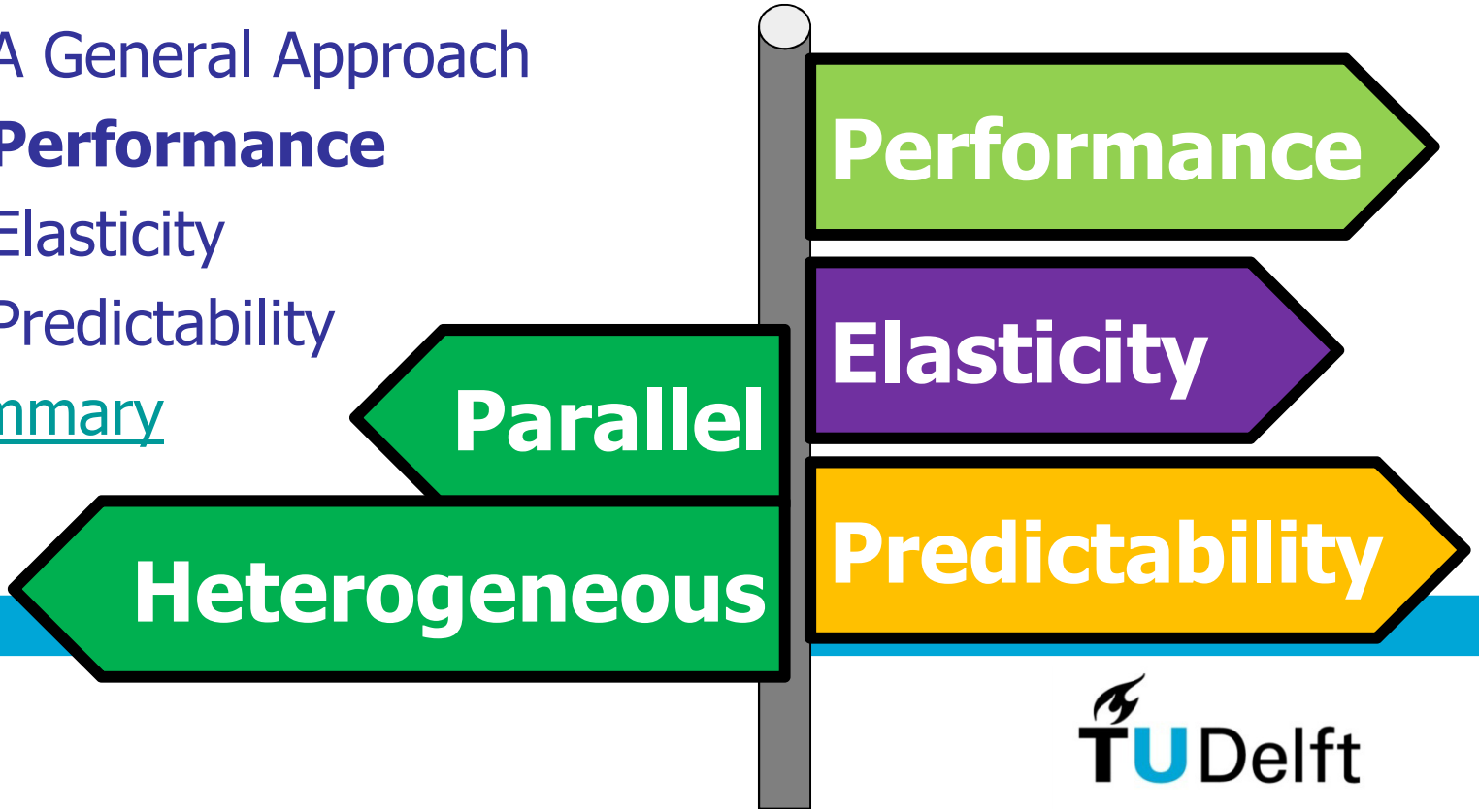


(d) Nvidia GeForce GTX480

TU Delft

Agenda

1. Introduction
2. Programming Models for Big Data
3. **Big Data on GPUs and Clouds**
   1. A General Approach
   2. **Performance**
   3. Elasticity
   4. Predictability
4. Summary



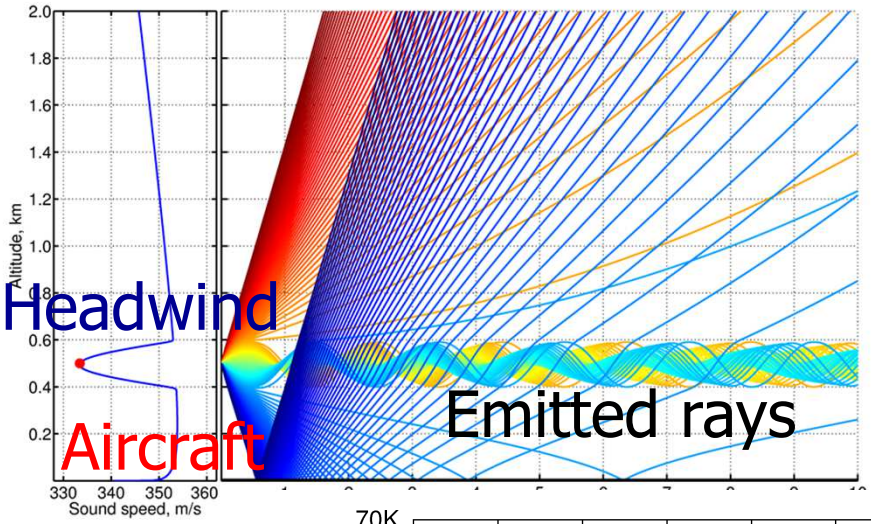
TU Delft

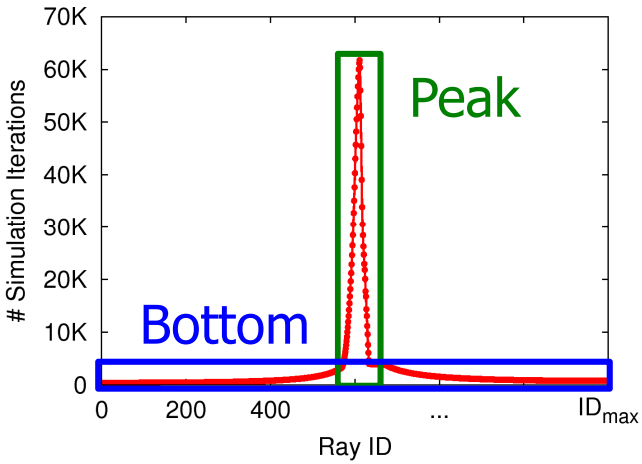
Heterogeneous Platforms

- Massive parallelism of modern architectures
- Many task programming model seems suitable to exploit massive parallelism, but …
- Need to address application (task) and data decomposition

Imbalanced Workloads on Fused Architectures

- Acoustic ray-tracing
- Fused architecture
  - Task + Data parallelism
  - Divide the whole workloads into
    - A bottom part (on the GPU)
    - A peak part (on the multi-core CPU)
  - multi-core CPU(s) **and** GPU(s)
- Experimental results
  - **10x better performance than traditional**
  - Auto-tuned soft real-time approaching hard real-time: ~30 ms

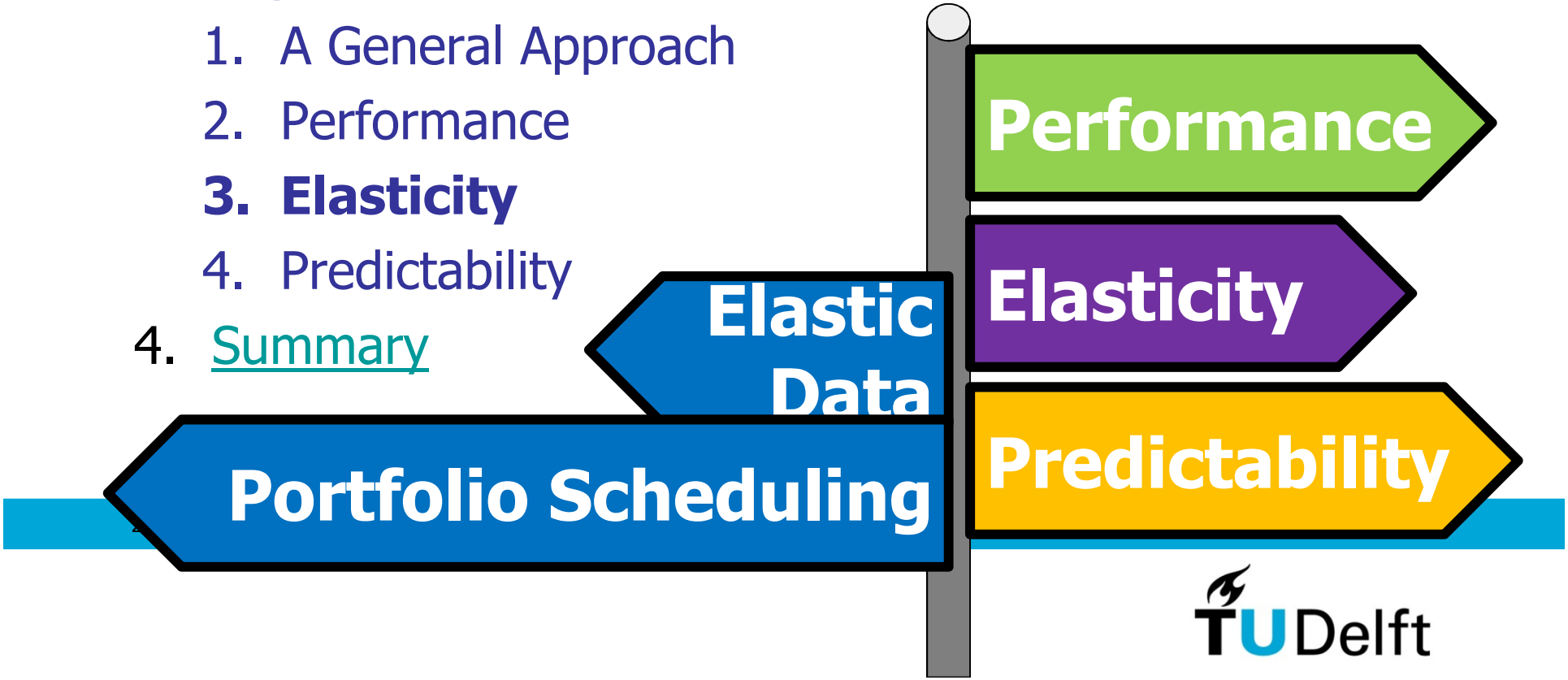




Shen et al. . Glinda: A Framework for Accelerating Imbalanced Applications on Heterogeneous Platforms. CF'13.

Agenda

1. Introduction
2. Programming Models for Big Data
3. **Big Data on GPUs and Clouds**
   1. A General Approach
   2. Performance
   3. **Elasticity**
   4. Predictability
4. Summary



TU Delft

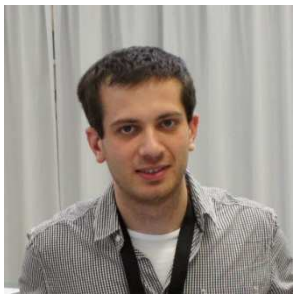
Elasticity: Our Team

Alexandru Iosup
TU Delft

Provisioning
Allocation
Elasticity
Portfolio Scheduling
Isolation
Multi-Tenancy

Dick Epema
TU Delft

Provisioning
Allocation
Koala



Bogdan Ghit
TU Delft

Provisioning
Allocation
Koala

Athanasios Antoniou
TU Delft

Provisioning
Allocation
Isolation
Utility

Kefeng Deng
NUDT
Portfolio Scheduling

Orna Agmon-Ben Yehuda
Technion
Elasticity, Utility

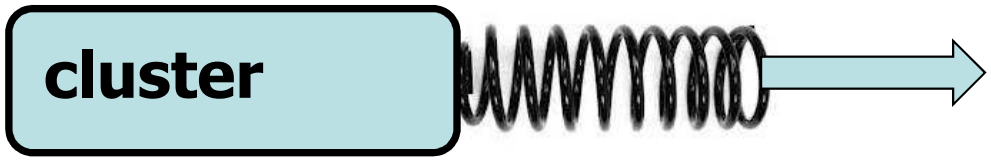
David Villegas
FIU/IBM
Elasticity, Utility

TU Delft

Elasticity, Performance and Cost-Awareness
Why Dynamic Data Processing Clusters?

- Improve resource utilization
  - Grow when the workload is too heavy
  - Shrink when resources are idle
- Fairness across multiple data processing clusters
  - Redistribute idle resources
  - Allocate resources for new MR clusters

Isolation
- Performance
- Failure
- Data
- Version

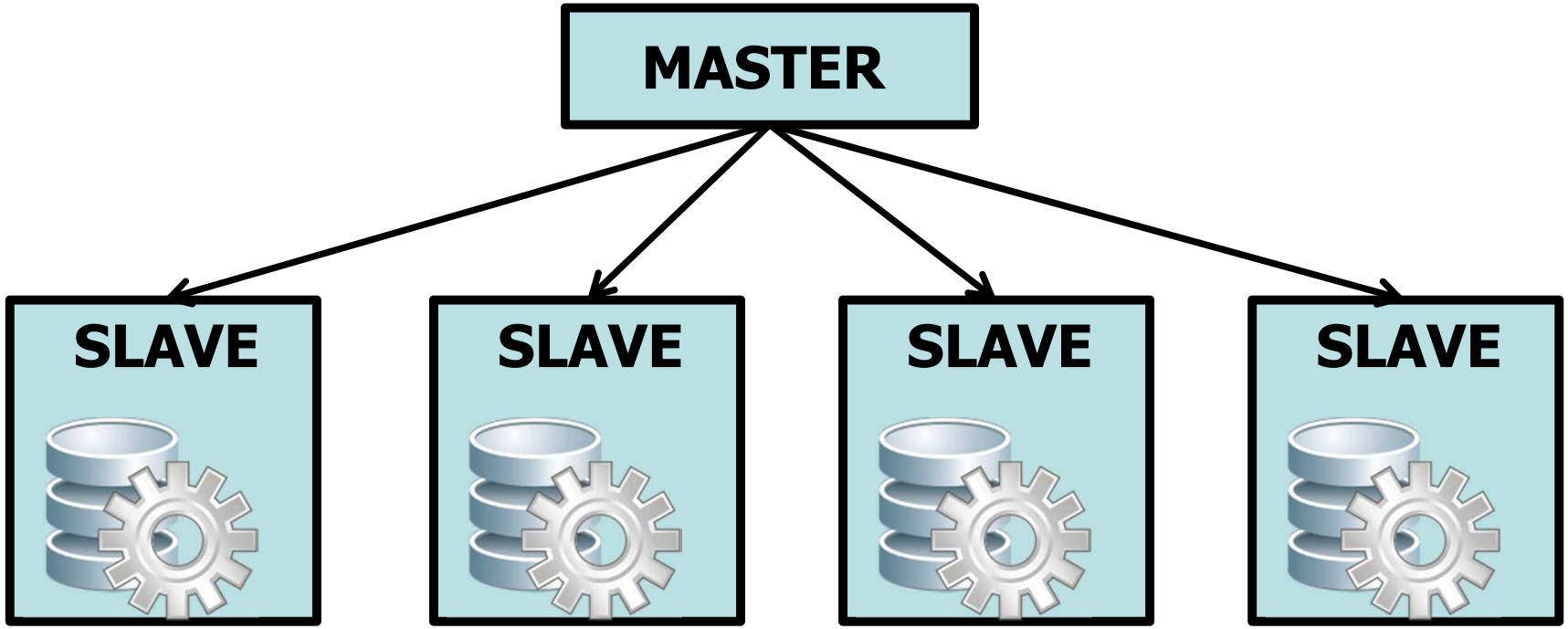


Ghit and Epema. Resource Management for Dynamic MapReduce Clusters in Multicluster Systems. MTAGS 2012. Best Paper Award.

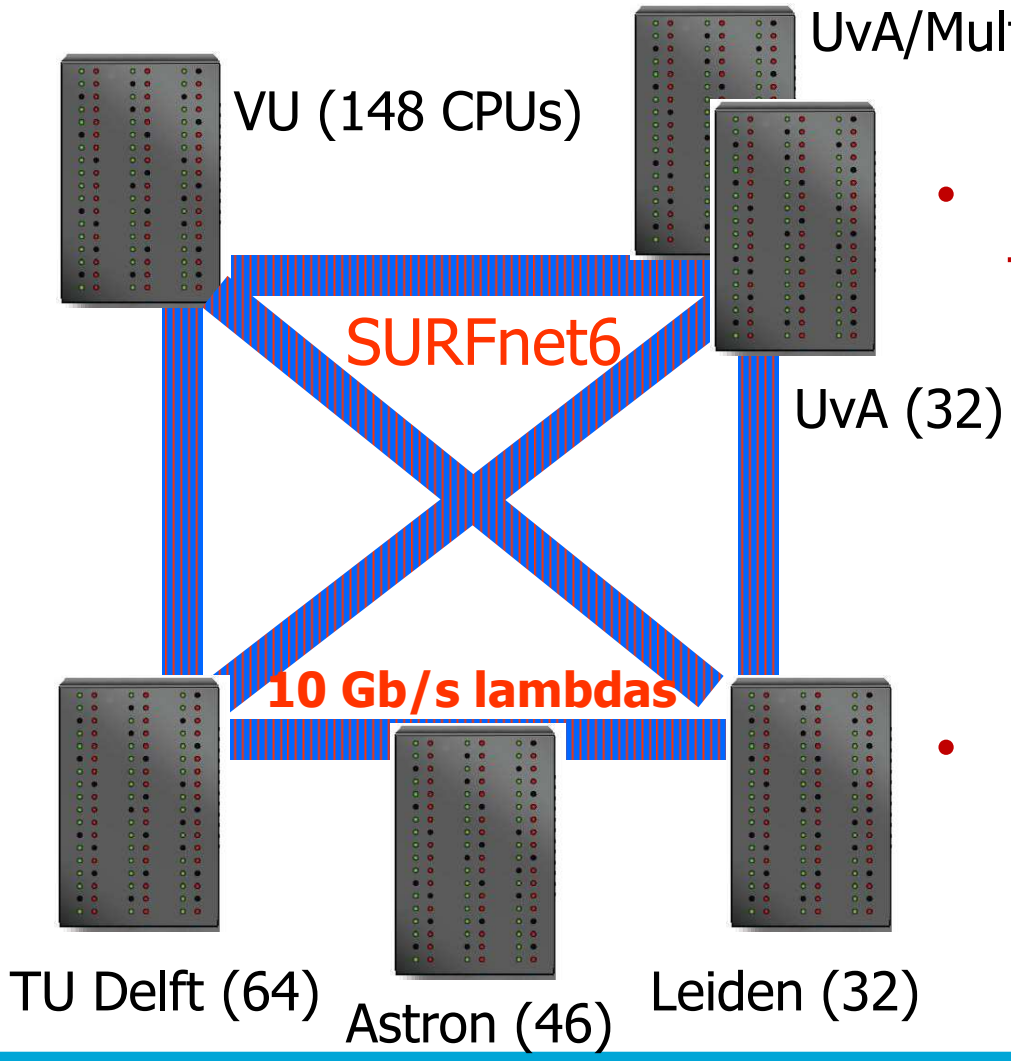
TU Delft

MapReduce Overview

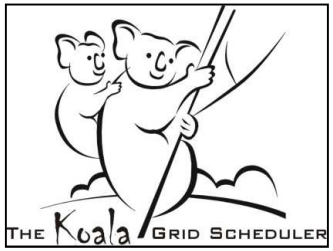
- MR cluster
  - Large-scale data processing
  - Master-slave paradigm
- Components
  - Distributed file system (storage)
  - MapReduce framework (processing)



The DAS-4 Infrastructure



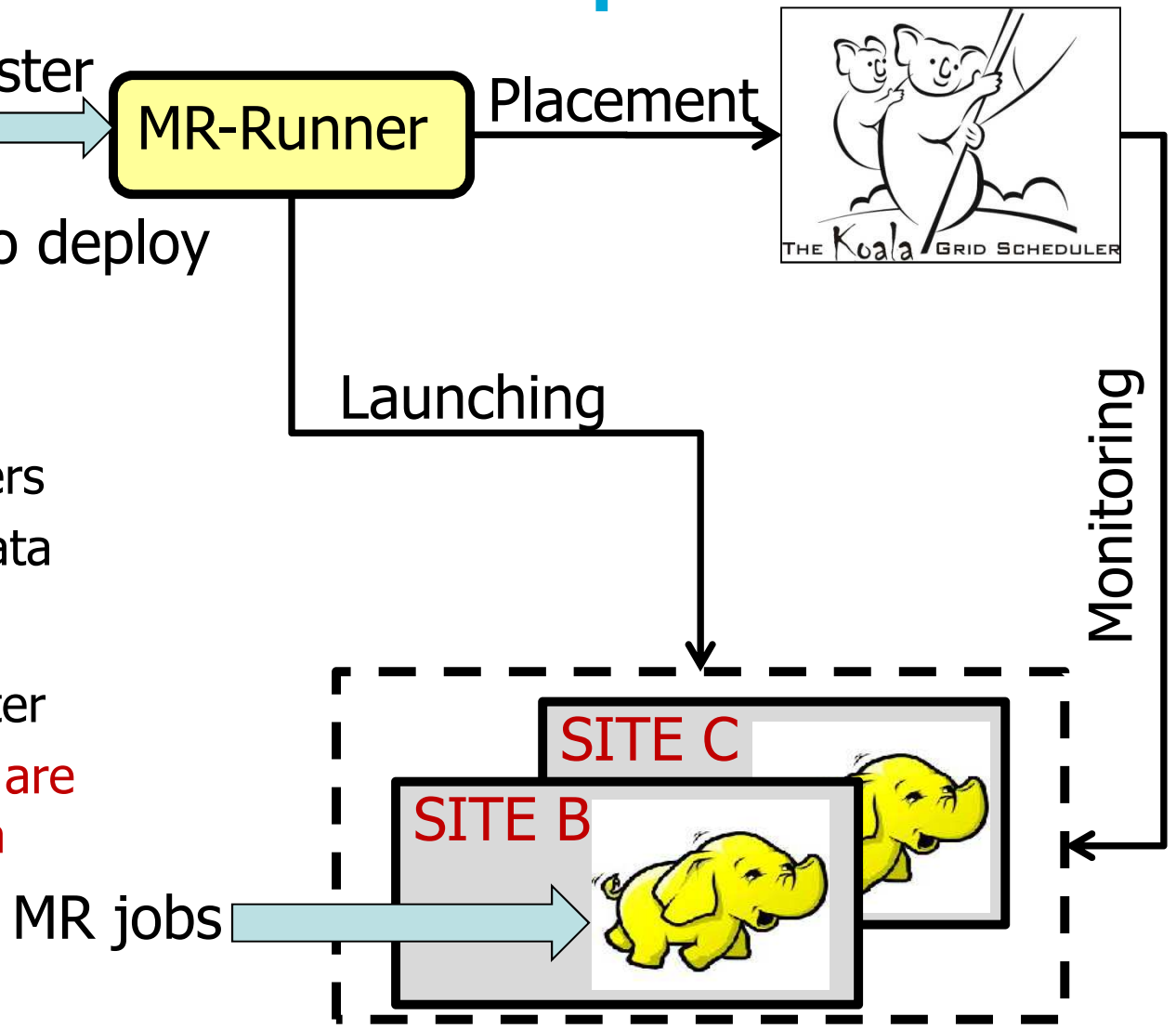
- Used for research in systems for over a decade
  - 1,600 cores (quad cores)
  - 2.4 GHz CPUs, GPUs
  - 180 TB storage
  - 10 Gbps Infiniband
  - 1 Gbps Ethernet
- Koala grid scheduler



TU Delft

KOALA Grid Scheduler and MapReduce

- Users submit jobs to deploy MR clusters
- Koala
  - Schedules MR clusters
  - Stores their meta-data
- MR-Runner
  - Installs the MR cluster
  - MR job submissions are transparent to Koala

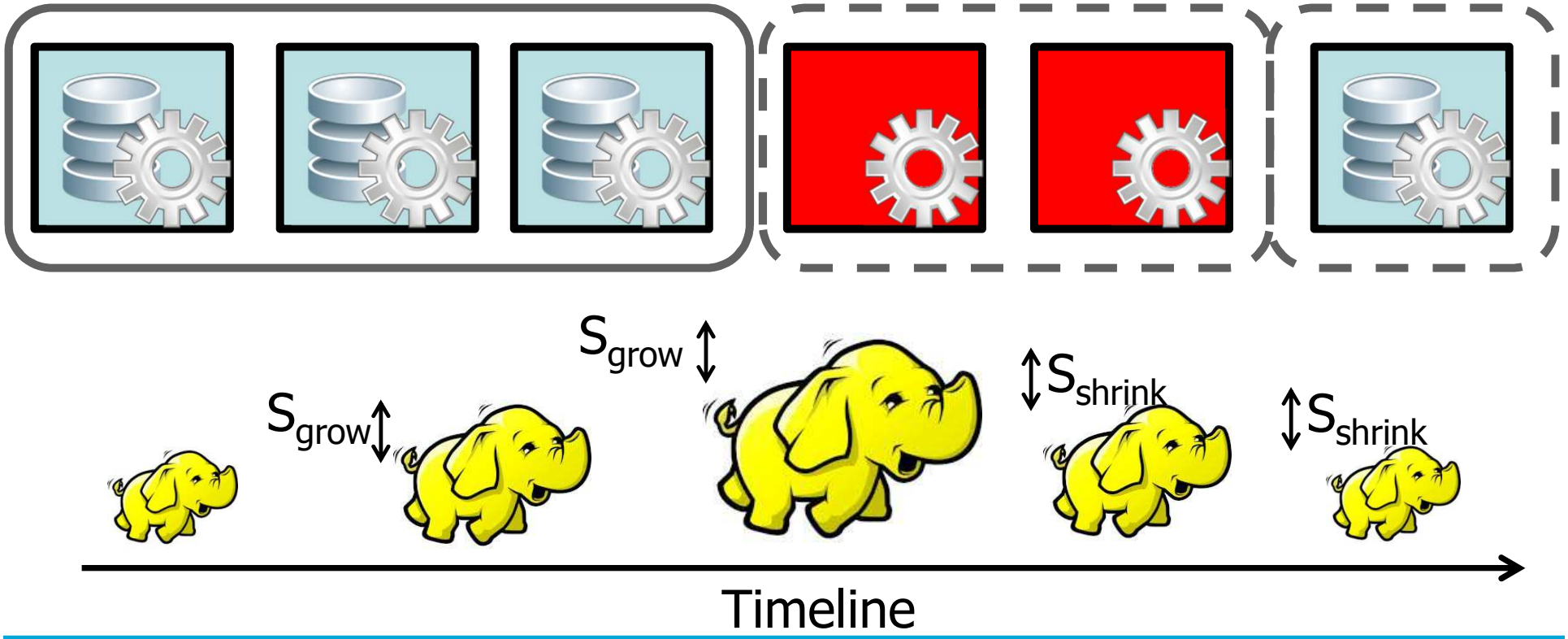


Ghit and Epema. Resource Management for Dynamic MapReduce Clusters in Multicluster Systems. MTAGS 2012. Best Paper Award.

TU Delft

Elastic MapReduce, TUD version

- Two types of nodes
  - Core nodes: TaskTracker and DataNode
  - Transient nodes: only TaskTracker



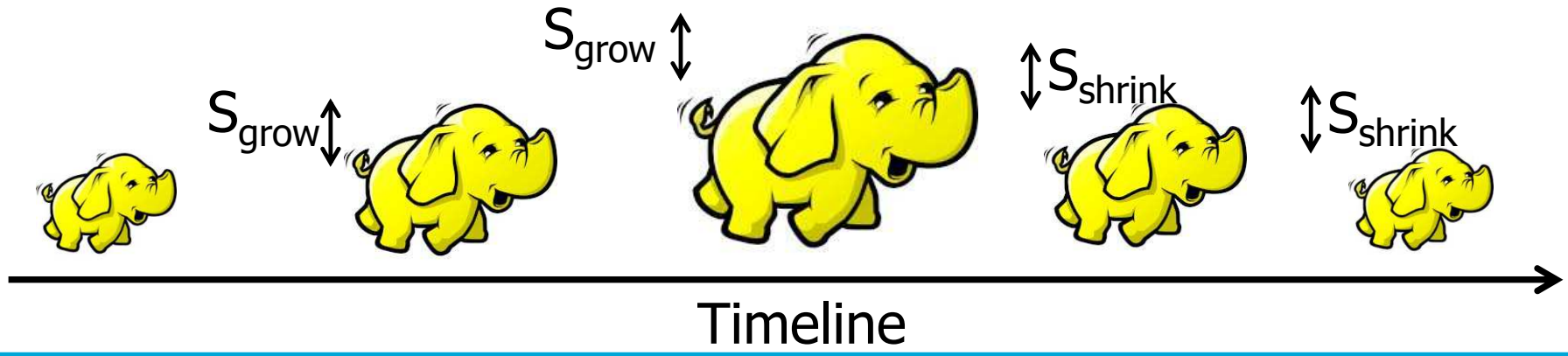
Ghit and Epema. Resource Management for Dynamic MapReduce Clusters in Multicluster Systems. MTAGS 2012. Best Paper Award.

TU Delft

Resizing Mechanism

- Two-level provisioning
  - Koala makes resource offers / reclaims
  - MR-Runners accept / reject request

- Grow-Shrink Policy (GSP)
  - MR cluster utilization: $F_{\min} \leq \frac{totalTasks}{availSlots} \leq F_{\max}$
  - Size of grow and shrink steps: **S_{grow}** and **S_{shrink}**

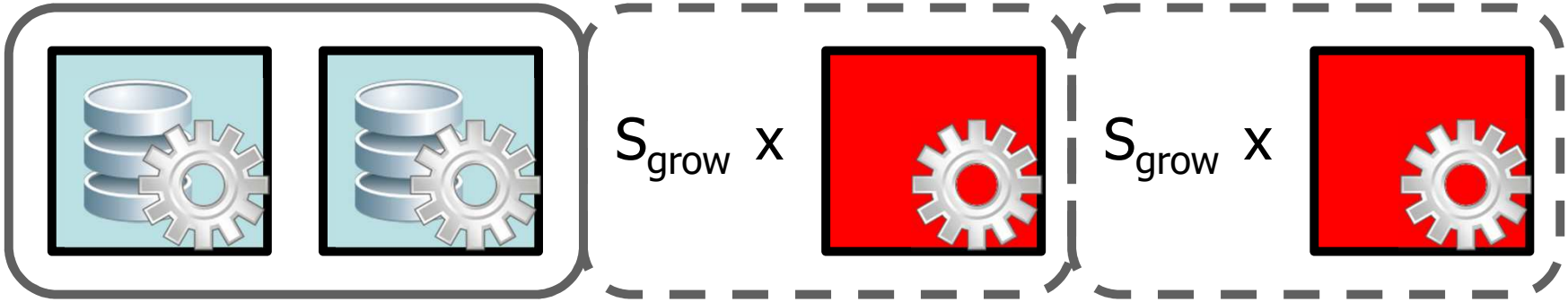


Ghit and Epema. Resource Management for Dynamic MapReduce Clusters in Multicluster Systems. MTAGS 2012. Best Paper Award.

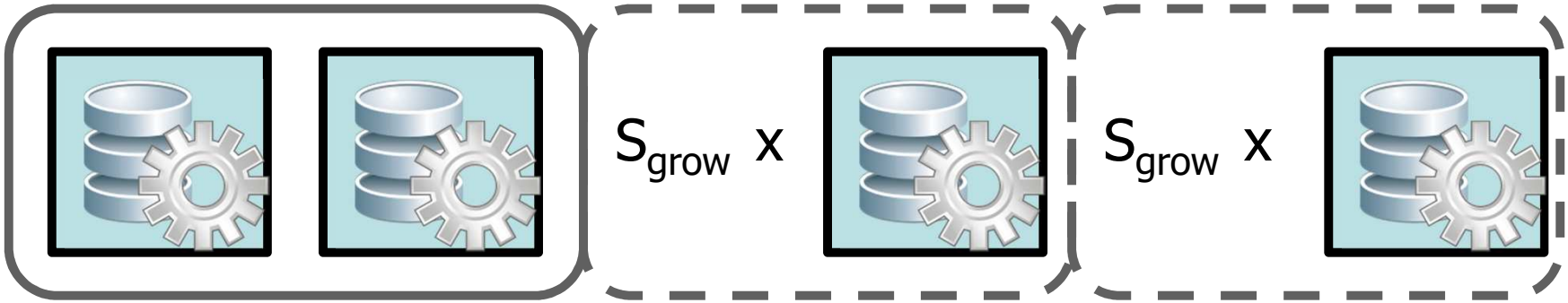
TU Delft

Baseline Policies

- Greedy-Grow Policy (GGP)—only grow with transient nodes:



- Greedy-Grow-with-Data Policy (GGDP)—grow, core nodes:



Ghit and Epema. Resource Management for Dynamic MapReduce Clusters in Multicluster Systems. MTAGS 2012. Best Paper Award.

Setup

- *98% of jobs @ Facebook take less than a minute*
- *Google reported computations with TB of data*

- DAS-4
- Two applications: Wordcount and Sort

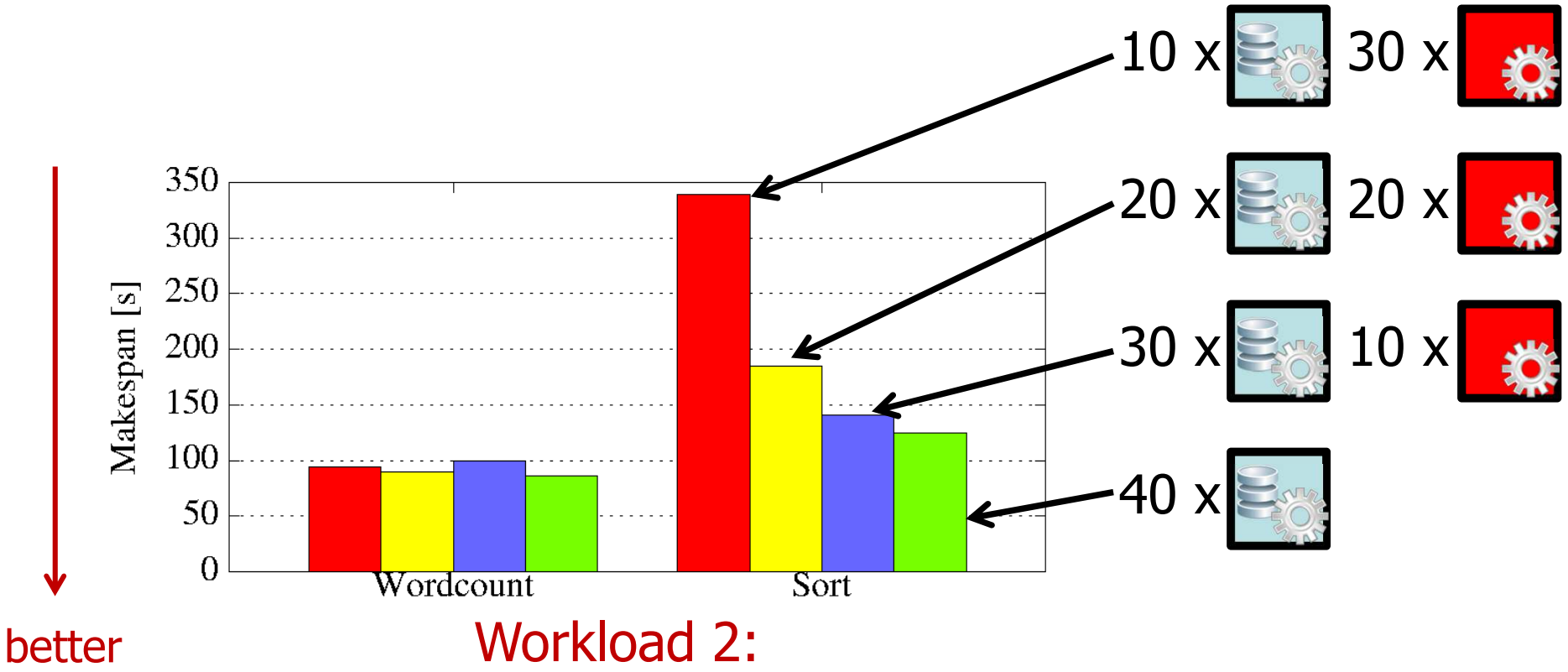
Workload 1
- Single job
- 100 GB
- Makespan

Workload 2
- Single job
- 40 GB, 50 GB
- Makespan

Workload 3
- Stream of 50 jobs
- 1 GB → 50 GB
- Average job execution time

Ghit and Epema. Resource Management for Dynamic MapReduce Clusters in Multicluster Systems. MTAGS 2012. Best Paper Award.

Transient Nodes



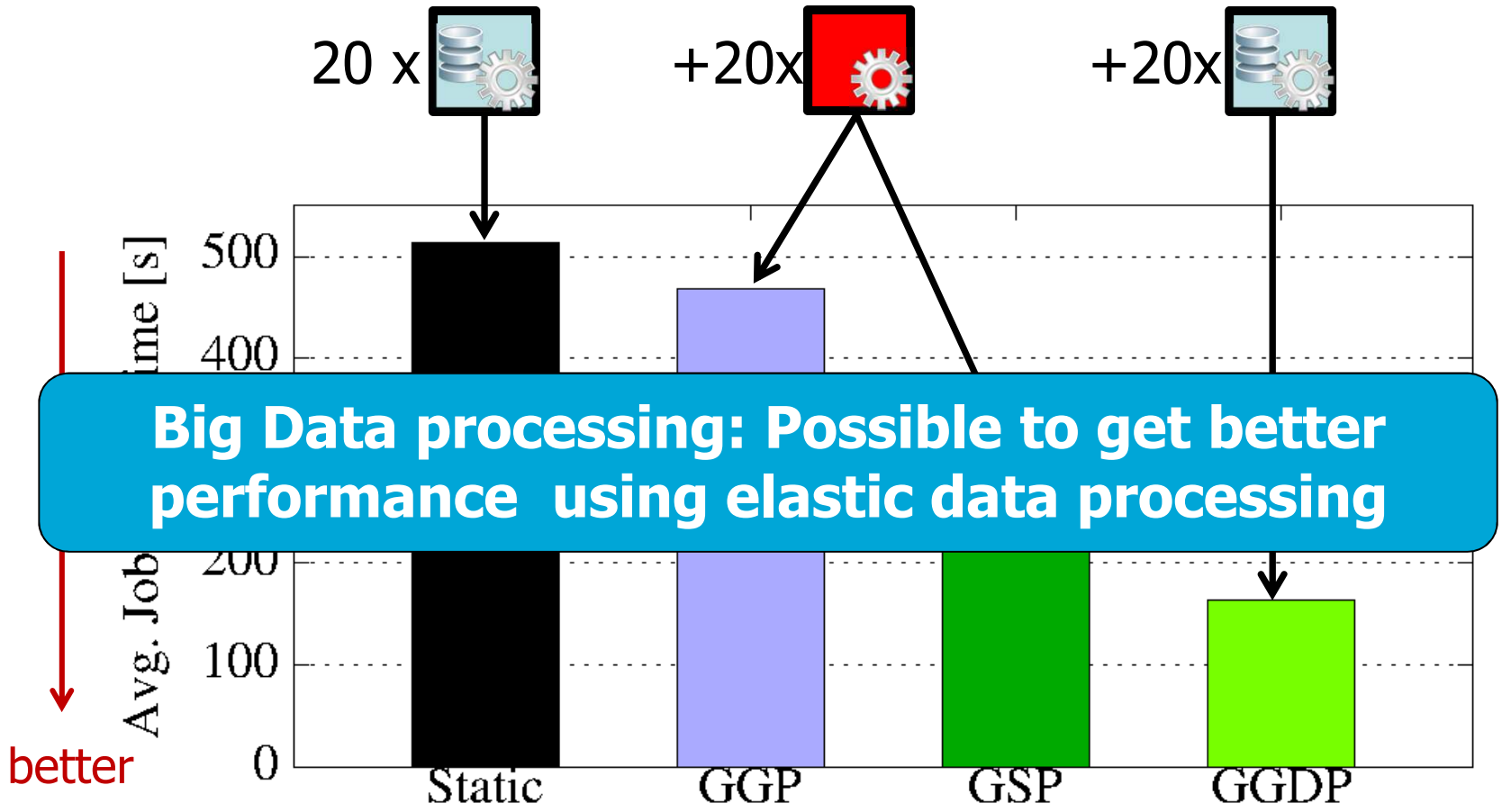
Workload 2:
40GB, 50GB

- Wordcount scales better than Sort on transient nodes

Ghit and Epema. Resource Management for Dynamic MapReduce Clusters in Multicluster Systems. MTAGS 2012. Best Paper Award.

TU Delft

Performance of Resizing using Static, Transient, and Core Nodes



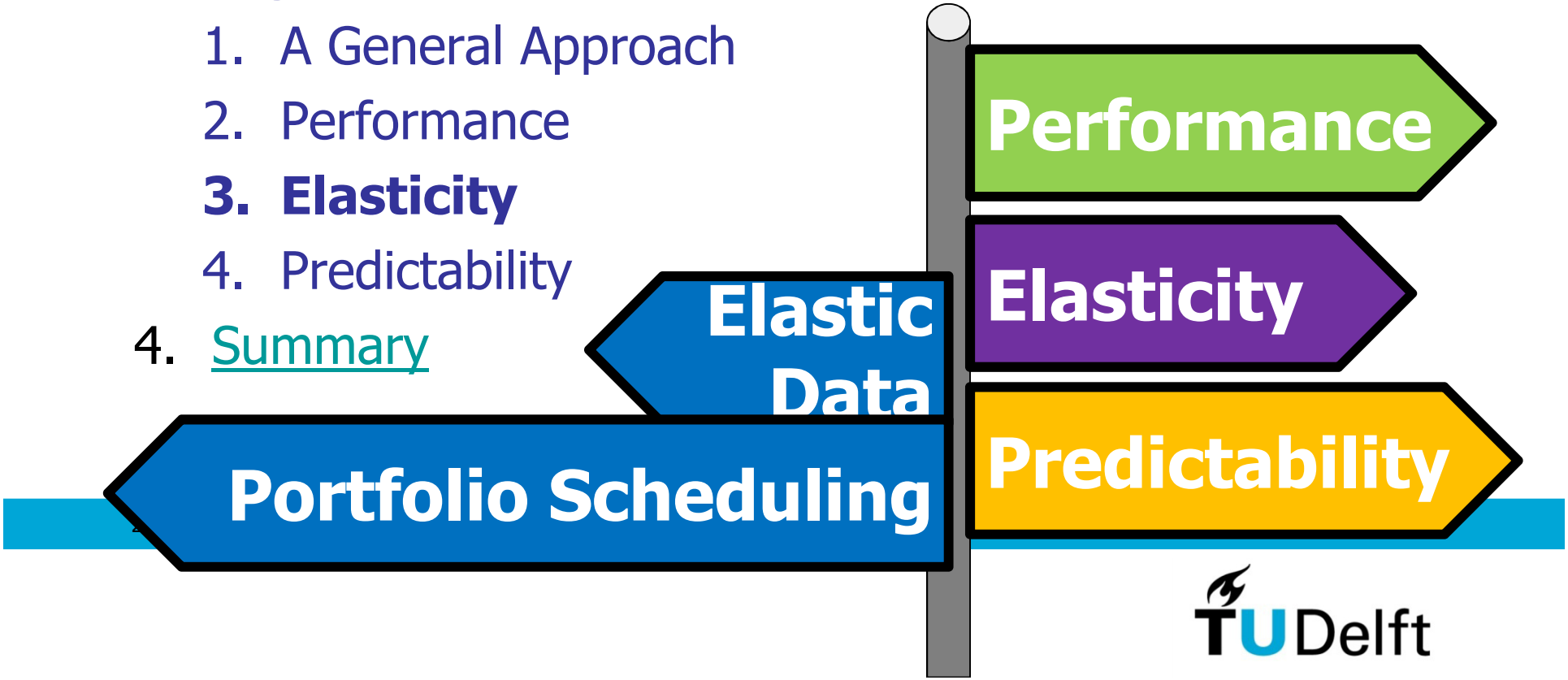
Sort + WordCount
(50 jobs, 1-50GB)

Ghit and Epema. Resource Management for Dynamic MapReduce Clusters in Multicluster Systems. MTAGS 2012. Best Paper Award.

TU Delft

Agenda

1. Introduction
2. Programming Models for Big Data
3. **Big Data on GPUs and Clouds**
   1. A General Approach
   2. Performance
   3. **Elasticity**
   4. Predictability
4. Summary



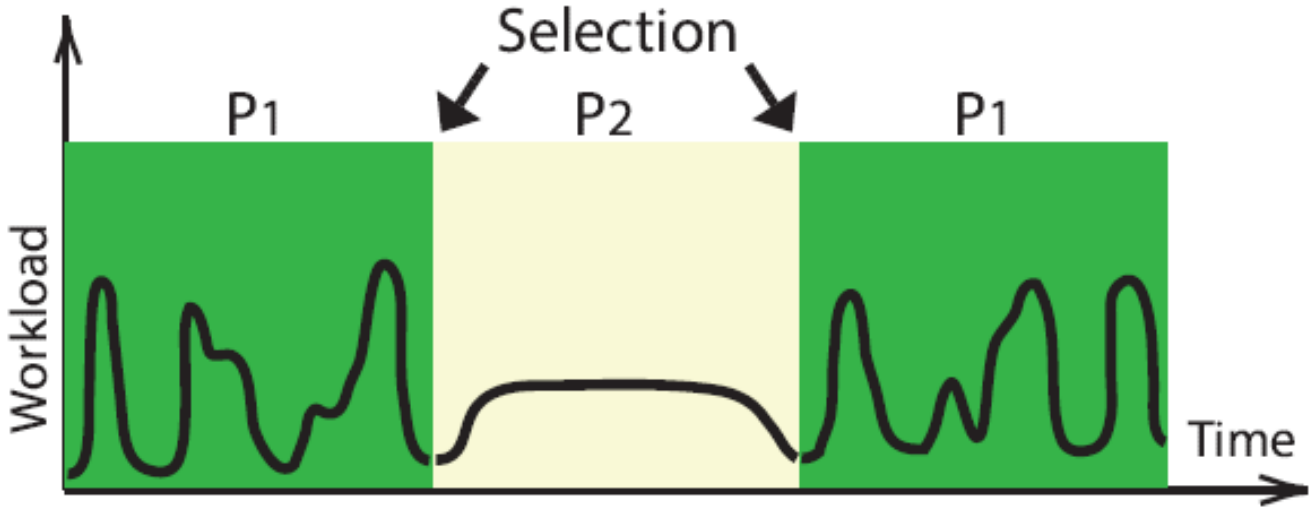
TU Delft

Elasticity, Portfolio Scheduling
Why Portfolio Scheduling?

- **Old scheduling aspects**
  - Hundreds of approaches, each targeting specific conditions—which to choose? How to configure?
  - No one-size-fits-all policy
- **New scheduling aspects**
  - New workloads, e.g., pretty much all Big Data
  - New data center architectures
  - New cost models, e.g., moving workloads to IaaS clouds

- **Developing a scheduling policy is risky and ephemeral**
- **Selecting a scheduling policy is risky and difficult**

What is Portfolio Scheduling?
In a Nutshell, for Elastic Big Data Processing



- Create a set of scheduling policies
  - Resource provisioning and allocation policies
- Online selection of the active policy, at important moments
  - Periodic selection, for example
- Same principle for other changes: pricing model, system, …

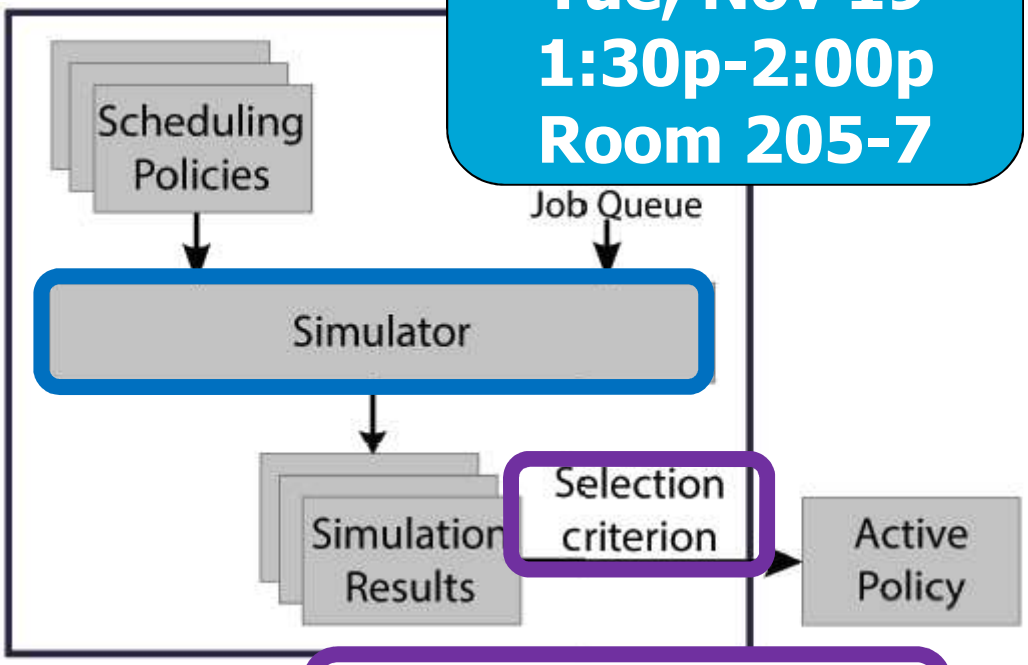
Portfolio Scheduling (Technical)

SC|13
Tue, Nov 19
1:30p-2:00p
Room 205-7

- Periodic execution

- Simulation-based selection
- Utility function

- Alternatives simulator
  - Expert human knowledge
  - WL sample in real env.
  - Mathematical analysis
- Alternatives utility function
  - Well-known and exotic functions



α=β=1
K=100

$$U = \kappa \cdot \left(\frac{R_J}{R_V}\right)^{\alpha} \cdot \left(\frac{1}{S}\right)^{\beta}$$

R_J: Total Runtime of Jobs
R_V: Total Runtime of VMs
S: Slowdown

Deng, Verboon, Iosup. A Periodic Portfolio Scheduler for Scientific Computing in the Data Center. JSSPP'13.

Deng, Song, Ren, Iosup. Exploring portfolio scheduling for long-term execution of scientific workloads in IaaS clouds. SC|13.

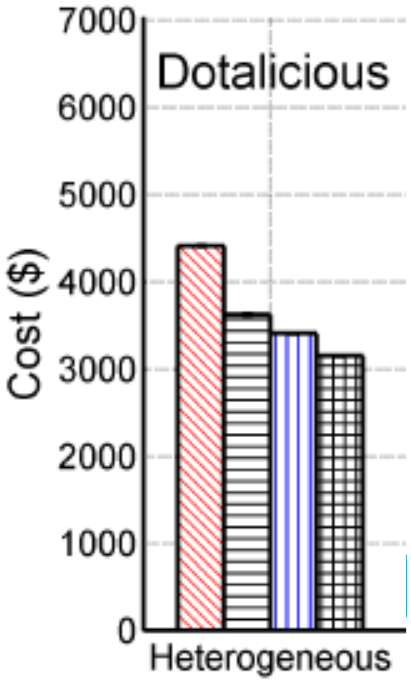
Agmon Ben-Yehuda, Schuster, Sharov, Silberstein, Iosup. ExPERT: pareto-efficient task replication on grids and a cloud. IPDPS'12.

Portfolio Scheduling for Online Gaming
(also for Scientific Workloads)

- **CoH =** Cloud-based, online, Hybrid scheduling
  - Intuition: keep rental cost low by finding good mix of machine configurations and billing options, use **on-demand cloud VMs**
  - Main idea: run *both* solver of an Integer Programming Problem and various heuristics, **pick best schedule periodically (at deadline)**
  - Additional feature: Can use **reserved cloud instances**

Gaming (and scientific) workloads

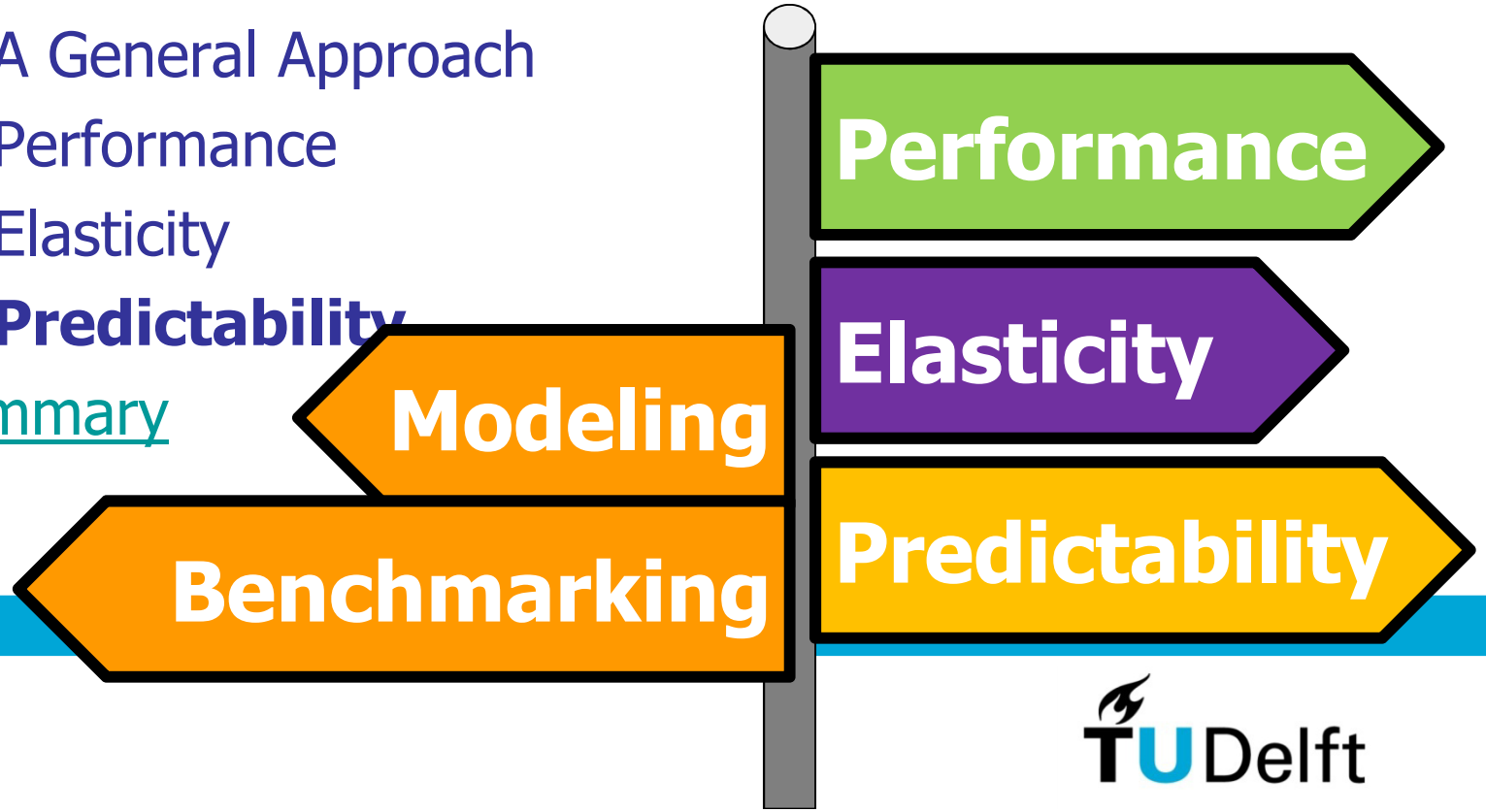
Trace	#jobs	average runtime [s]
Grid5000	200,450	2728
LCG	188,041	8971
DotaLicious	109,251	2231



Shen, Deng, Iosup, and Epema. Scheduling Jobs in the Cloud Using On-demand and Reserved Instances, EuroPar'13.

Agenda

1. Introduction
2. Programming Models for Big Data
3. **Big Data on GPUs and Clouds**
   1. A General Approach
   2. Performance
   3. Elasticity
   4. **Predictability**
4. Summary

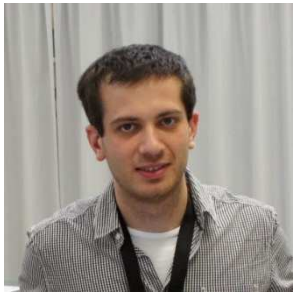


TU Delft

Predictability: Our Team

Alexandru Iosup
TU Delft
Modeling
Benchmarking

Dick Epema
TU Delft
Modeling
Benchmarking

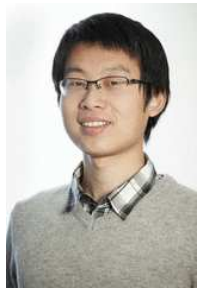


Bogdan Ghit
TU Delft
Modeling
Benchmarking

Ana Lucia Varbanescu
U. Amsterdam
Graph processing
Benchmarking

Claudio Martella
VU Amsterdam
All things Giraph

Mihai Capota
TU Delft
Big Data apps
Benchmarking



Yong Guo
TU Delft
Graph processing
Benchmarking

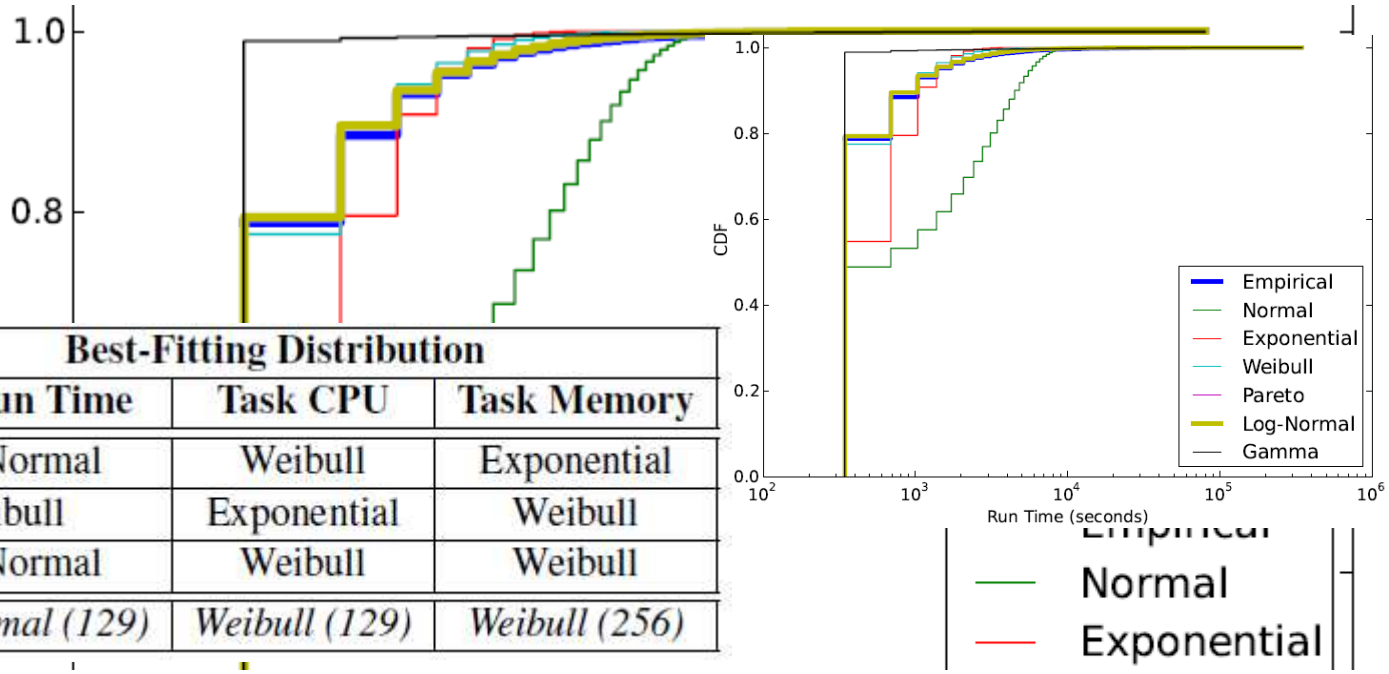
Marcin Biczak
TU Delft
Cloud Computing
Performance Eval.
Development

Modeling

- Characterization of big data applications, both algorithm and dataset
- Characterization of system
- Model performance or any other attribute as function of algorithm, data, data processing model, and (transient) resource substrate

When Long-Term Traces Exist
Our Statistical MapReduce Models

- Real traces
  - Yahoo
  - Google
  - 2 x Social N



		Best-Fitting Distribution		
Job	**Task Count**	**Task Run Time**	**Task CPU**	**Task Memory**
1	1	Log-Normal	Weibull	Exponential
2	128	Weibull	Exponential	Weibull
3	128	Log-Normal	Weibull	Weibull
Overall Best Fit		*Log-Normal (129)*	*Weibull (129)*	*Weibull (256)*

Model	Tasks	Correlation	Map/Reduce Modeled	Sign. Level	Indirect Distr. Sel.
Complex Model	Indirect	Run time – Disk	Separately	0.05	Best fits
Relaxed Complex Model	Indirect	Run time – Disk	Separately	0.02	All fits
Safe Complex Model	Direct	Run time – Disk	Separately	0.05	–
Simple Model	Direct	–	Together	0.05	–

de Ruiter and Iosup. A workload model for MapReduce. MSc thesis at TU Delft. Jun 2012. Available online via TU Delft Library, http://library.tudelft.nl .

The BTWorld Use Case (When Long-Term Traces Do Not Exist)

Collected Data

- BitTorrent: swarms of people sharing files
  - 100M users
  - At some point 35% of total internet traffic
- Data-driven project: data first, ask questions later
- Over 14TB of data, 1 file/tracker/sample
- Timestamped, multi-record files
  - Hash: unique id for file
  - Tracker: unique id for tracker
  - Information per file: seeders, leechers

Wojciechowski, Capota, Pouwelse, and Iosup. BTWorld: Towards observing the global BitTorrent file-sharing network. HPDC 2010

The BTWorld Use Case (When Long-Term Traces Do Not Exist)

Analyst Questions

- How does the number of peers evolve over time?
- How long are files available?
- Did the legal bans and tracker take-downs impact BT?
- How does the location of trackers evolve over time?
- Etc.

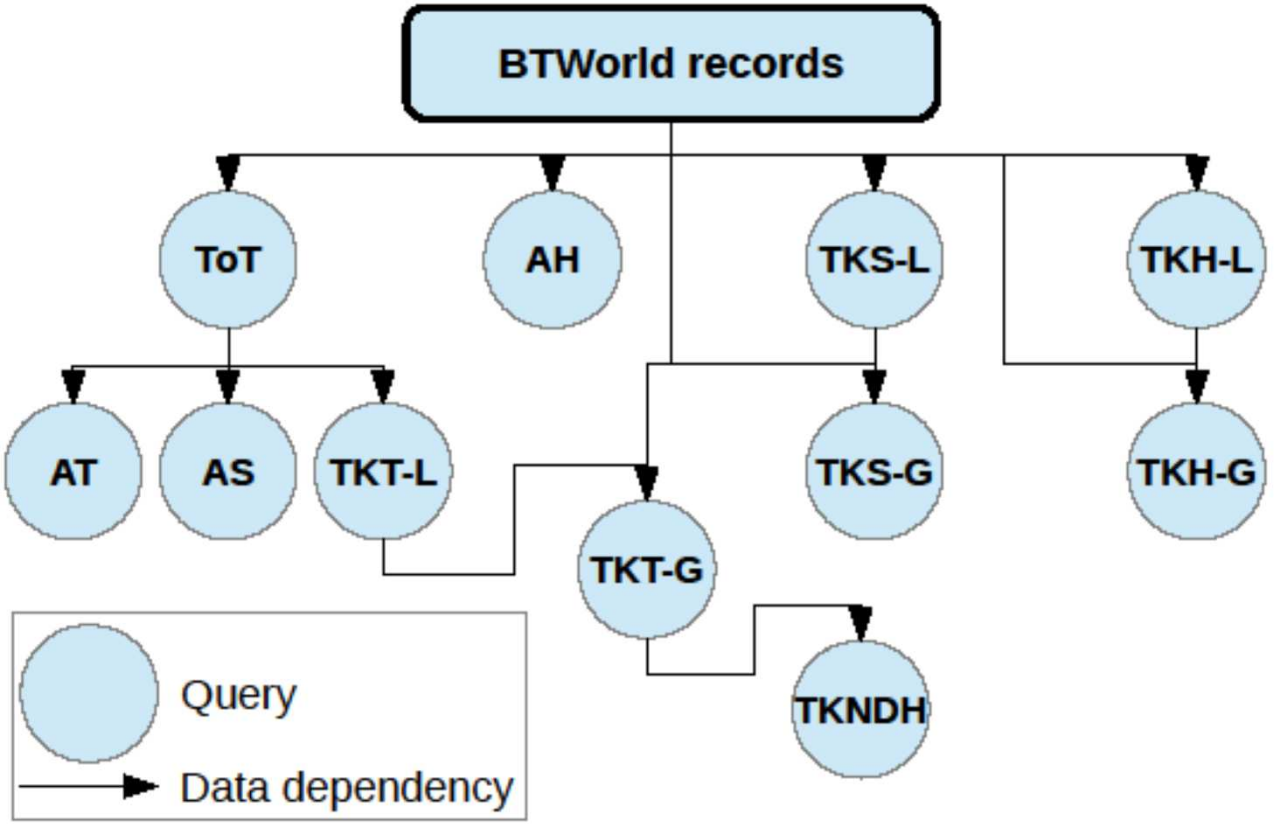
These questions need to be translated into queries

Hegeman, Ghit, Capotã, Hidders, Epema, Iosup. The BTWorld Use Case for Big Data Analytics: Description, MapReduce Logical Workflow, and Empirical Evaluation.IEEE BigData'13

MapReduce-based Workflow for the BTWorld Use Case

Overview

Complex workflow with inter-query dependencies



Hegeman, Ghit, Capotã, Hidders, Epema, Iosup. The BTWorld Use Case for Big Data Analytics: Description, MapReduce Logical Workflow, and Empirical Evaluation.IEEE BigData'13

MapReduce-based Workflow for the BTWorld Use Case

Query Diversity

- Queries use different operators, stress different parts of system
- Workflow is **not** modeled well by single-application benchmarks

Global Top K Trackers (TKT-G):

```
SELECT *
FROM logs
NATURAL JOIN (
    SELECT tracker
    FROM TKTL
    GROUP BY tracker
    ORDER BY MAX(sessions) DESC
    LIMIT k);
```

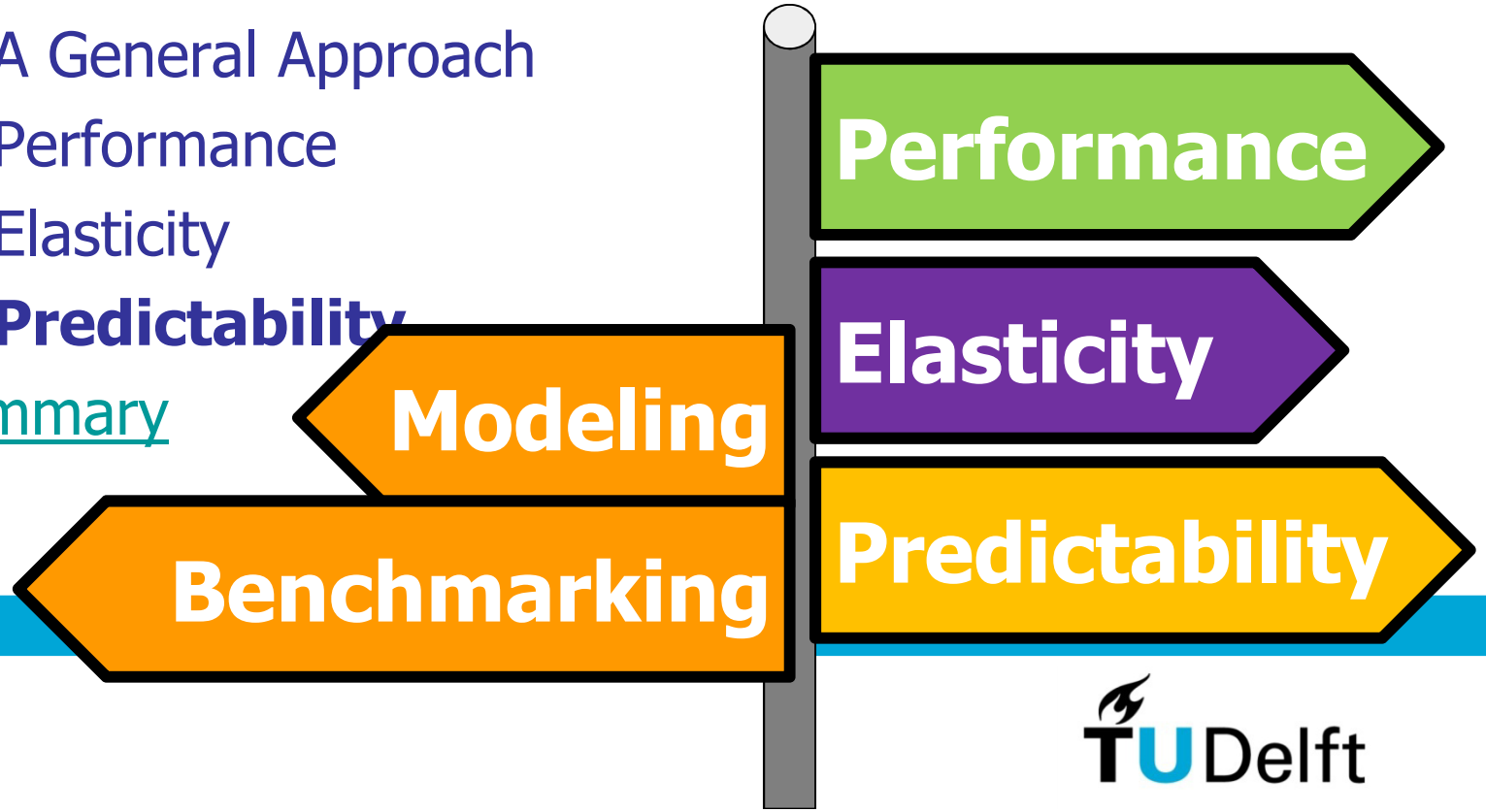
Active Hashes (AH):

```
SELECT timestamp, COUNT(DISTINCT(hash))
FROM logs
GROUP BY timestamp;
```

Hegeman, Ghit, Capotã, Hidders, Epema, Iosup. The BTWorld Use Case for Big Data Analytics: Description, MapReduce Logical Workflow, and Empirical Evaluation.IEEE BigData'13

Agenda

1. Introduction
2. Programming Models for Big Data
3. **Big Data on GPUs and Clouds**
   1. A General Approach
   2. Performance
   3. Elasticity
   4. **Predictability**
4. Summary



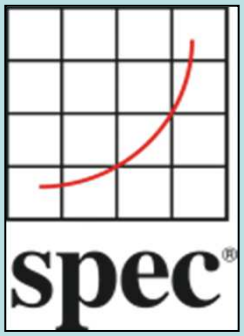
TU Delft

Benchmarking

- From single kernel or solitary-kernel suite to …
  Big Data processing workflow
- Derived from modeling …
  Intra-query, intra-job, and inter-job data dependencies
- Can benchmarking be
  - Realistic?
  - Cost- and time-effective?
  - Fair?

SPEC Research Group (RG)

The Research Group of the
Standard Performance Evaluation Corporation



Mission Statement

- Provide a **platform for** collaborative research efforts in the areas of computer benchmarking and quantitative system analysis
- Provide metrics, tools and benchmarks for evaluating early prototypes and research results as well as full-blown implementations
- Foster interactions and collaborations btw. industry and academia

Ad: Join us!

More information: *http://research.spec.org*

TU Delft
Delft University of Technology

Our Method

A benchmark suite for
performance evaluation of graph-processing platforms

1. Multiple Metrics, e.g.,
   - Execution time
   - Normalized: EPS, VPS
   - Utilization

2. Representative graphs with various characteristics, e.g.,
   - Size
   - Directivity
   - Density

3. Typical graph algorithms, e.g.,
   - BFS
   - Connected components

http://bit.ly/10hYdIU

Guo, Biczak, Varbanescu, Iosup, Martella, Willke:
How Well do Graph-Processing Platforms Perform?
An Empirical Performance Evaluation and Analysis

Graphitti

Benchmarking suite
Data sets

Graphs	# V	# E	d ($\times 10^{-5}$)	$\bar{D}$	Size	Directivity
Amazon	262.1 K	1.2 M	1.8	4.7	18 MB	directed
WikiTalk	2.4 M	5.0 M	0.1	2.1	87 MB	directed
KGS	293.3 K	16.6 M	38.5	112.9	210 MB	undirected
Citation	3.8 M	16.5 M	0.1	4.4	297 MB	directed
DotaLeague	61.2 K	50.9 M	2,719.0	1,663.2	655 MB	undirected
Synth	2.4 M	64.2 M	2.2	53.6	964 MB	undirected
Friendster	65.6 M	1.8 B	0.1	55.1	31 GB	undirected

SNAP

Graph500
http://www.graph500.org/

The Game Trace Archive
http://gta.st.ewi.tudelft.nl/

Guo, Biczak, Varbanescu, Iosup, Martella, Willke.
How Well do Graph-Processing Platforms Perform?
An Empirical Performance Evaluation and Analysis

Graphitti

Q4

Benchmarking suite
Platforms and Process

- Platforms

- Process
  - Evaluate baseline (out of the box) and tuned performance
  - Evaluate performance on fixed-size system
  - Future: evaluate performance on elastic-size system
  - Evaluate scalability

http://bit.ly/10hYdIU

Guo, Biczak, Varbanescu, Iosup, Martella, Willke:
How Well do Graph-Processing Platforms Perform?
An Empirical Performance Evaluation and Analysis

Graphitti

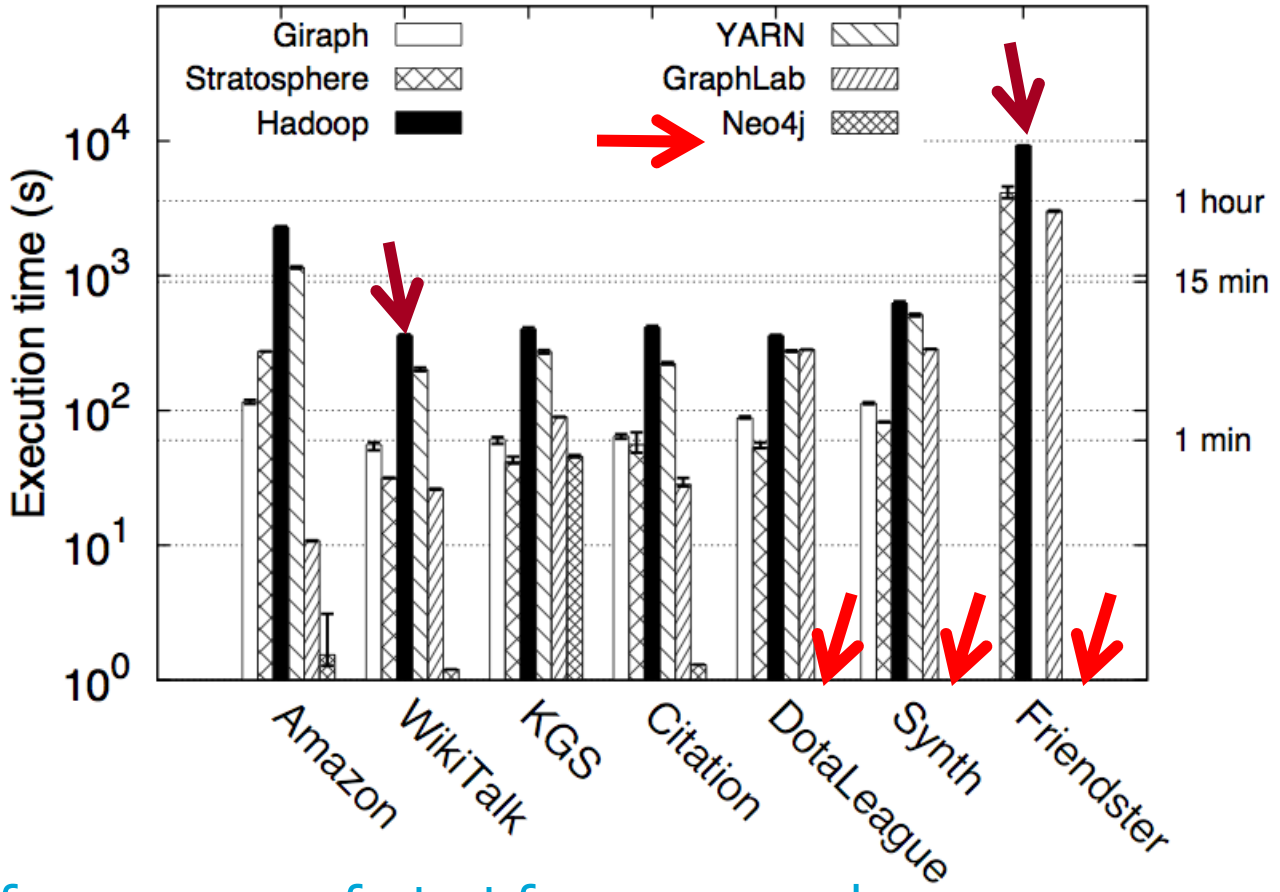
Experimental setup

- Size
  - Most experiments take 20 working nodes
  - Up to 50 working nodes

- DAS4: a multi-cluster Dutch grid/cloud
  - Intel Xeon 2.4 GHz CPU (dual quad-core, 12 MB cache)
  - Memory 24 GB
  - 10 Gbit/s Infiniband network and 1 Gbit/s Ethernet network
  - Utilization monitoring: Ganglia

- HDFS used here as distributed file systems

BFS: results for all platforms, all data sets



- No platform can runs fastest for every graph
- Not all platforms can process all graphs
- Hadoop is the worst performer

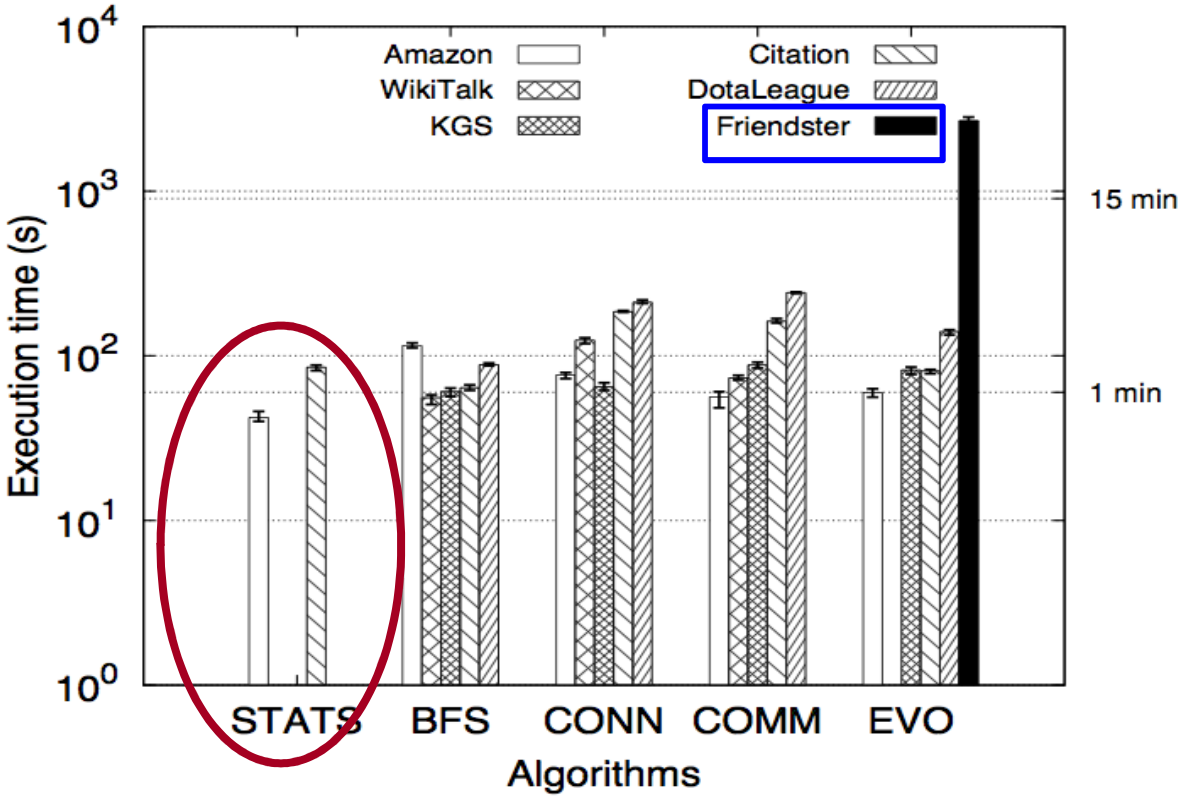
http://bit.ly/10hYdIU

Guo, Biczak, Varbanescu, Iosup, Martella, Willke:
How Well do Graph-Processing Platforms Perform?
An Empirical Performance Evaluation and Analysis

Graphitti

Giraph: results for all algorithms, all data sets

http://bit.ly/10hYdIU



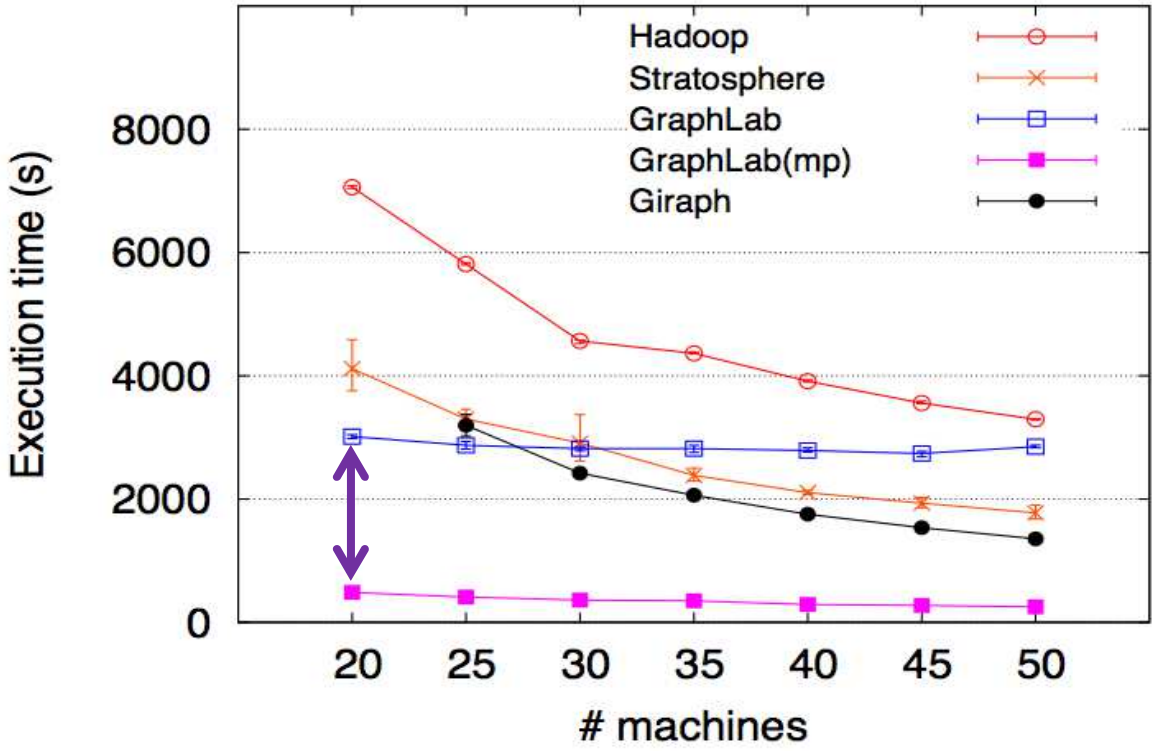
- Storing the whole graph in memory helps Giraph perform well
- Giraph may crash when graphs or messages become larger

Guo, Biczak, Varbanescu, Iosup, Martella, Willke.
How Well do Graph-Processing Platforms Perform?
An Empirical Performance Evaluation and Analysis

Graphitti

Horizontal scalability: BFS on Friendster (31 GB)

http://bit.ly/10hYdIU



- Using more computing machines can reduce execution time
- Tuning needed for horizontal scalability, e.g., for GraphLab, split large input files into number of chunks equal to the number of machines

Guo, Biczak, Varbanescu, Iosup, Martella, Willke.
How Well do Graph-Processing Platforms Perform?
An Empirical Performance Evaluation and Analysis

Graphitti

Additional Overheads
Data ingestion time

http://bit.ly/10hYdIU

- Data ingestion
  - Batch system: one ingestion, multiple processing
  - Transactional system: one ingestion, one processing

- Data ingestion matters even for batch systems

	Amazon	DotaLeague	Friendster
HDFS	1 second	7 seconds	5 minutes
Neo4J	4 hours	6 **days**	**n/a**

Guo, Biczak, Varbanescu, Iosup, Martella, Willke.
How Well do Graph-Processing Platforms Perform?
An Empirical Performance Evaluation and Analysis

Graphitti

Conclusion and ongoing work

- Performance is f(Data set, Algorithm, Platform, Deployment)
- Cannot tell yet which of (Data set, Algorithm, Platform) the most important (also depends on Platform)
- Platforms have their own drawbacks
- Some platforms can scale up reasonably with cluster size (horizontally) or number of cores (vertically)

- Ongoing work
  - *Benchmarking* suite
  - Build a performance boundary model
  - Explore performance variability

http://bit.ly/10hYdIU

Guo, Biczak, Varbanescu, Iosup, Martella, Willke:
How Well do Graph-Processing Platforms Perform?
An Empirical Performance Evaluation and Analysis

Graphitti

~~Conclusion~~ Take-Home Message

- **Programming Models for Big Data**
  - **Big data programming models have ecosystems**
  - **Pick your stack and you're stuck!**
  - Many trade-offs, many programming models, many problems
- **A Generic Big-Data Processing System**
  - Looking at Execution Engine
  - Performance challenges: parallel from the beginning, fused architectures
  - Elasticity challenges: elastic data processing, portfolio scheduling, etc.
  - Pedictability challenges: modeling, benchmarking, etc.
  - …
- **Conclusion: a thousand flowers already bloomed, through our general approach they may become fruitful**

http://www.flickr.com/photos/dimitrisotiropoulos/4004766418/

TU Delft

Thank you for your attention! Questions? Suggestions? Observations?

More Info:

- http://www.st.ewi.tudelft.nl/~iosup/research.html
- http://www.st.ewi.tudelft.nl/~iosup/research_cloud.html
- http://www.pds.ewi.tudelft.nl/

Alexandru Iosup

A.Iosup@tudelft.nl
http://www.pds.ewi.tudelft.nl/~iosup/ (or google "iosup")
Parallel and Distributed Systems Group
Delft University of Technology

Survey Big Data Usage

http://goo.gl/TJwkTg

Reading Material

SC|13
Tue, Nov 19
1:30p-2:00p
Room 205-7

- **Programming Models for Big Data**
  - Jeffrey Dean, Sanjay Ghemawat: MapReduce: Simplified Data Processing on Larg
  - Jeffrey Dean, Sanjay Ghemawat: MapReduce: a flexible data processing tool. Co
  - Matei Zaharia, Andy Konwinski, Anthony D. Joseph, Randy Katz, and Ion Stoica. 2008. Improving MapReduce performance in heterogeneous environments. In Proceedings of the 8th USENIX conference on Operating systems design and implementation (OSDI'08). USENIX Association, Berkeley, CA, USA, 29-42.
  - Matei Zaharia, Dhruba Borthakur, Joydeep Sen Sarma, Khaled Elmeleegy, Scott Shenker, Ion Stoica: Delay scheduling: a simple technique for achieving locality and fairness in cluster scheduling. EuroSys 2010: 265-278
  - Tyson Condie, Neil Conway, Peter Alvaro, Joseph M. Hellerstein, Khaled Elmeleegy, Russell Sears: MapReduce Online. NSDI 2010: 313-328
  - Grzegorz Malewicz, Matthew H. Austern, Aart J. C. Bik, James C. Dehnert, Ilan Horn, Naty Leiser, Grzegorz Czajkowski: Pregel: a system for large-scale graph processing. SIGMOD Conference 2010: 135-146
  - Dominic Battré, Stephan Ewen, Fabian Hueske, Odej Kao, Volker Markl, Daniel Warneke: Nephele/PACTs: a programming model and execution framework for web-scale analytical processing. SoCC 2010: 119-130

Pender and Varbanescu. MSc thesis at TU Delft. Jun 2012. TU Delft Library, http://library.tudelft.nl .

Shen et al. Glinda: A Framework for Accelerating Imbalanced Applications on Heterogeneous Platforms. CF'13.

Guo, Biczak, Varbanescu, Iosup, Martella, Wi
How Well do Graph-Processing Platforms Per
An Empirical Performance Evaluation and An

http://bit.ly/10hYdIU

Hegeman et al. The BTWorld Use Case for Big Data Analytics: Description, MapReduce Logical Workflow, and Empirical Evaluation.IEEE BigData'13.

Deng, Song, Ren, Iosup. Exploring portfolio scheduling for long-term execution of scientific workloads in IaaS clouds. SC|13.