IEEE WCNC, Doha – April 5, 2016

Services in the Clouds and Big Data Era

Albert Y. Zomaya

Professor and Director

Centre for Distributed & High Performance Computing

School off Information Technologies

The University of Sydney

Australia

THE UNIVERSITY OF SYDNEY

Australian Government

Australian Research Council

amazon

Centre for Distributed and High Performance Computing

- A 40+ member group. Past and current funding from the Australian Research Council, CISCO, ERICSSON, IBM, Microsoft, Sun, Smart Internet CRC, NICTA, DSTO and CSIRO.
- The Centre's mission is to establish **a streamlined research, technology exploration and advanced training program**. It will be a leading centre to undertake collaborative multi-disciplinary research in support of *distributed* and *high performance computing* and related industry to enable advances in information technology and other application domains.
- The Centre focuses currently on several themes which build on existing strengths at Sydney University:
  - **Algorithmics and Data Mining**
  - **Cloud Computing and Green ICT**
  - **Internetworking**
  - **Service Computing**
  - **Distributed Computing Applications**
- The current work: **Mr. Omer Adam, Dr. Young Choon Lee**

A Tale to Tell

- Regardless of the **research discipline** (in academia) or **business** (in industry), many organizations are dealing with **data-related problems** and for different reasons, including:
  - The Increasing volume of data (generated daily)
  - **Urgency** to get **results/outcomes from data** to make time-constrained **decisions**
  - **Project deadlines** to meet when processing data
  - Monetary cost/budget **limitations** on processing such data
  - The need to **Increase profits/revenue** and **reduce cost** by **rapidly exploiting** these data, for example in:
    - Stock markets.
    - **Interdisciplinary scientific research** where **data** is the **major research driver** in many fields incl. biomedical, molecular biology (genomics), economics, info. science & knowledge discovery (data mining), etc.
    - **Data Analytics** to exploit available powerful computations to maximize insights/information **extracted** from data.
- The above problems (and others) **all** came to a point where **no ordinary computational capabilities** are needed to treat them effectively.

A Tale to Tell

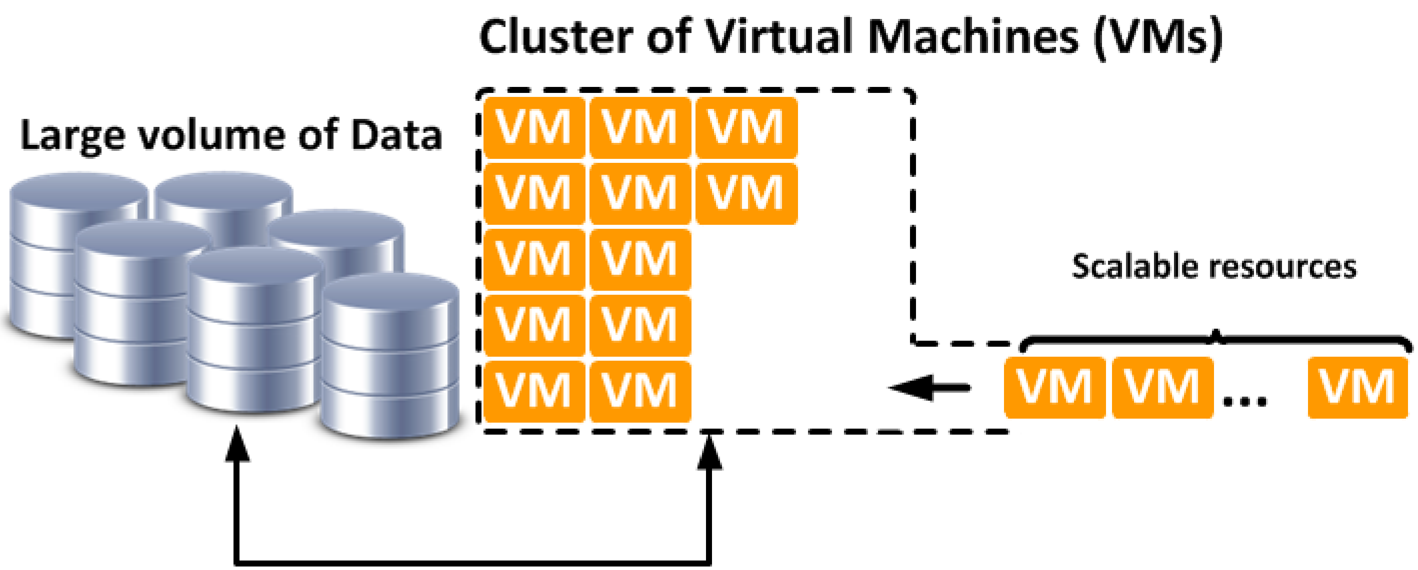
- Starting with High-Performance Computing (**HPC**) **clusters on-premise**.
  - Enabled by: the advances in distributed systems design.
  - Many 'legacy' paradigms emerged such as grid and cluster computing, etc.
  - **Expensive** Ownership.
  - **Rigid** paradigms for scalability of resource provisioning.

- A more efficient option is **Cloud Computing** that provides:
  - even **larger-scale distributed computations** with much **finer economical-control** over **resources.**
  - Public access to massive computing resources.
  - (**On-demand** / **At scale**) **Resource Allocations** for **clusters**.
  - No capital investment in IT infrastructure is needed anymore.

What is a Computing Cluster for?

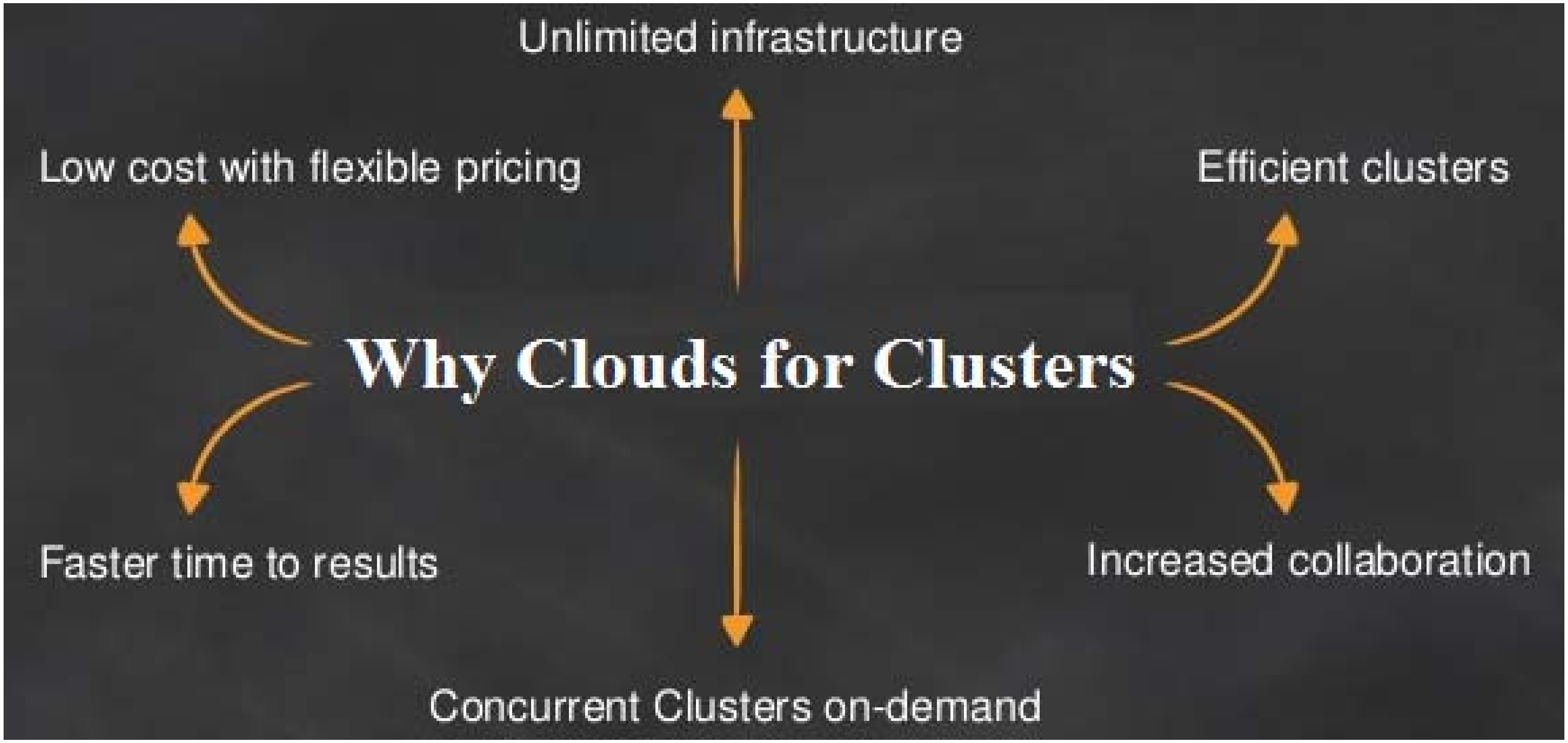
› Executing a **distributed application** (e.g. data analytics) over **massive** datasets requires **composing** a **scalable cluster** of **computing machines** that **cooperate**

- To perform **large-scale data analysis and manipulation** for scientific or business needs.
- To answer a query within a tolerable elapsed time, or
- To **process** certain volume of data for aggregation within a predetermined deadline.



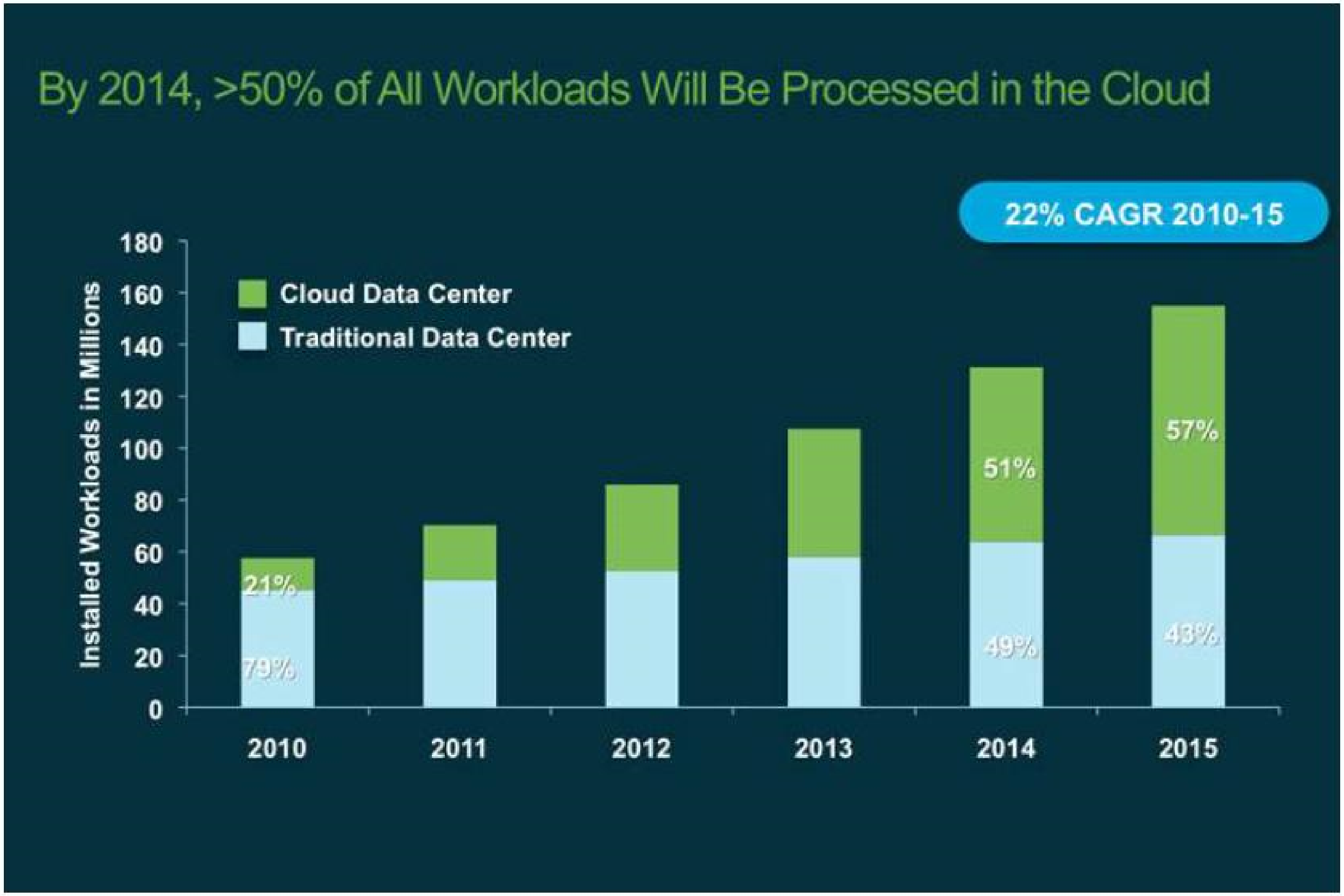
Why we need **Clouds** for Clusters?

- Public Clouds promise to give:



Rapid Growth of Cloud Adoption

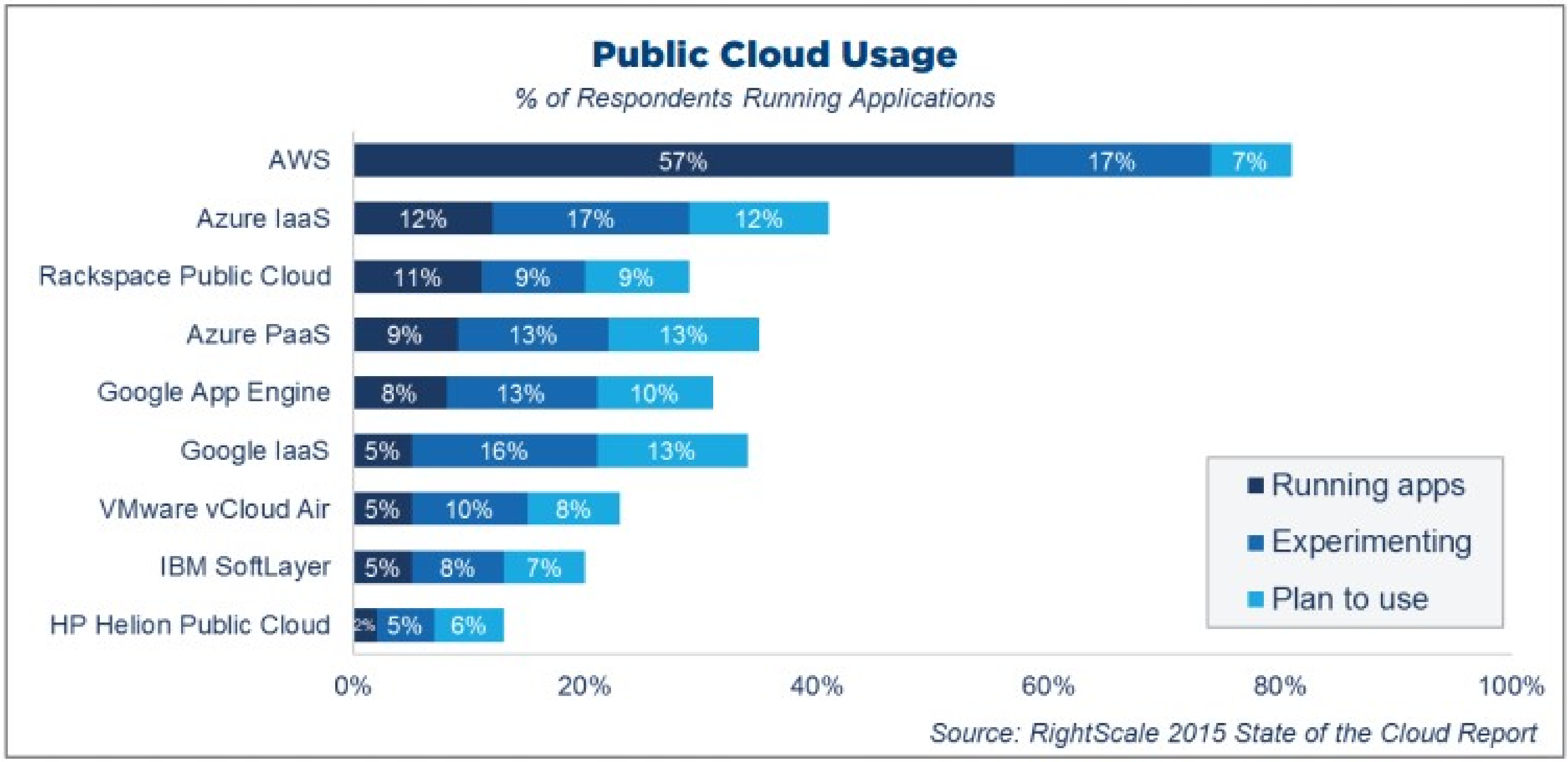
› Workloads/Applications distribution: 2010-2015



Source: Independent Analyst Shipment Data, Cisco Analysis

Cloud Providers

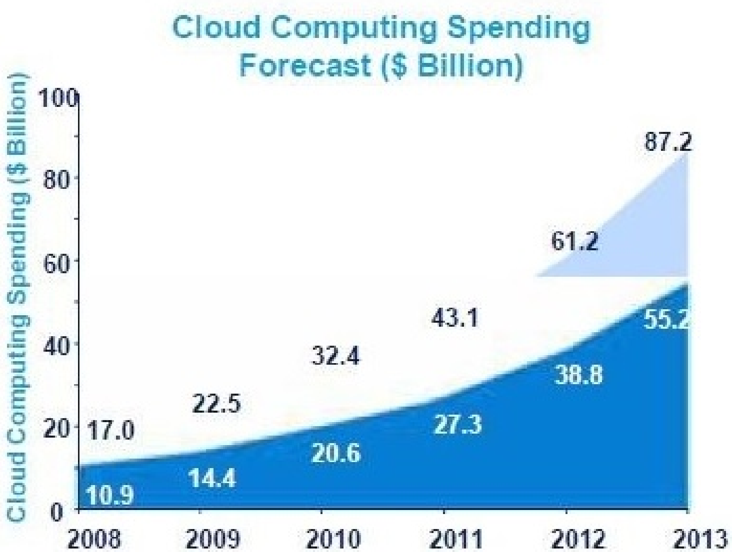
› With increasing interest in **Cloud adoption**, **application users** are given diverse options to construct their **computing clusters in clouds**.

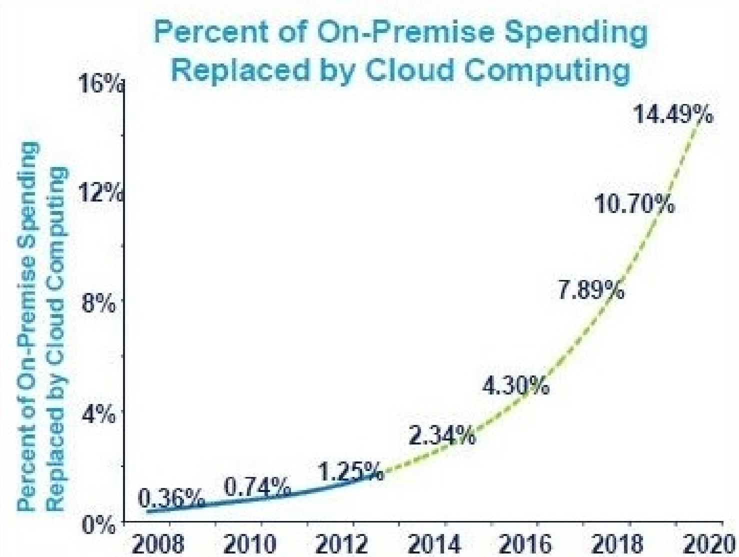


Inevitable Trend:
On-premise Clusters spending **Replaced** by Cloud computing

Impact of Cloud Computing on Enterprise IT Spending

Enterprise spending for on-premise solutions will fall, as cloud computing reduces the need for licenses, hardware and software.





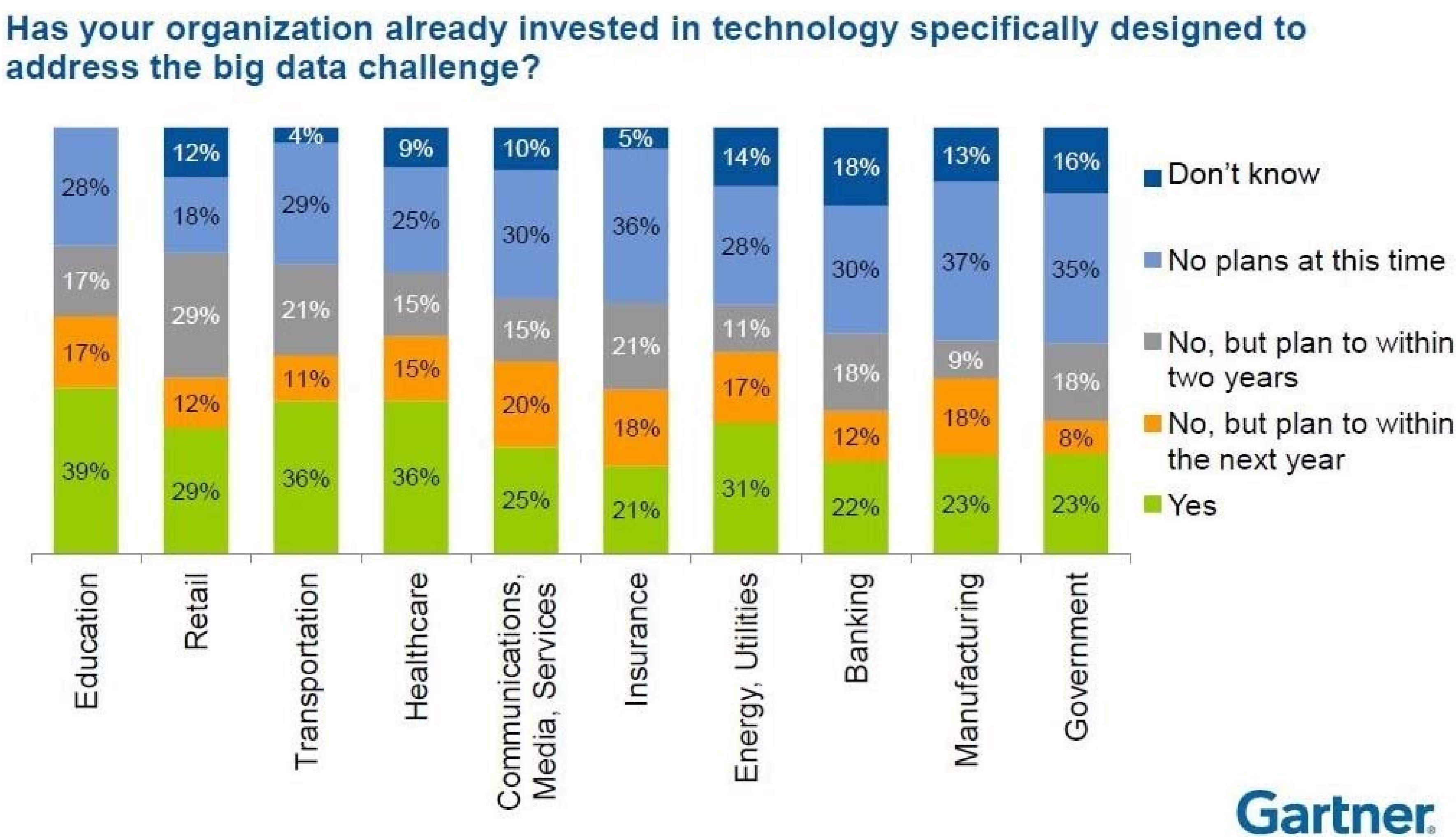
Extrapolation based on time-series analysis, assuming that factors promoting Cloud Computing through 2008-13 (——), remain constant beyond 2013 until 2020 (- - -).

The cloud computing market is forecast to grow at a rate of 36.6 percent during 2008-13 to $55.2 billion in 2013. It will probably reduce overall technology spending by $30.0-$39.4 billion in 2013, replacing 14.5 percent of global on-premise spending in 2020.
Source: Research and Innovation Estimates

Deloitte.
Australia

Business Investment:
Investment in **Computing Clusters** for Data Analytics

- With increasing trend: the need to **construct** cloud-based **computing clusters** for the purpose to run applications for data analysis becomes **increasingly demanded**.



The most common motivation for rapid Cloud Adoption: Emergence of Big Data problems (1)

› Cloud computing paradigm sells itself:

- By providing highly flexible billing model that is very convenient for everybody.
- Web services and Resources provisioning is **on-demand** and **at scale**.
- Remarkably reduce risk of investing in infrastructure.

- However, **this was NOT APPEALING enough to drive further cloud adoption** by some organizations.

› **Big Data problems** appear to be the **most common phenomena** that **pushes** more organizations to seriously consider adopting cloud computing to treat them:

- The increasing number of data sources/streams: is overwhelming.
- **90%** of the data in the world today has been created in the last two years alone (Source: IBM Big Data).
- Data can be stored. However, examining **raw big data** with the purpose of drawing conclusions/insights is a **challenge**, **nowadays**.

The most common motivation for rapid Cloud Adoption:
Cloud-based **Computing Clusters** for **Data Analytics** (2)

› Such **Big Data** problems ***advances*** *the design and development of* ***new distributed Data Analytics*** *applications* **to effectively exploit the power of public clouds,** and boost ‘time-to-solutions’:

- Data Analytics distributed applications require constructing **Computing Clusters** that are **scalable** and **resilient** to workload changes.
- In-house HPC clusters typically is **not effective** to provide **efficient Resource Allocation** mechanisms that is **scalable** with performance targets (e.g. deadlines).

› **Massively available** cloud **resources** enable to design ***efficient Resource Allocation mechanisms*** for **clusters** in clouds.

› ***Popular Hadoop Clusters***:

❑ **Google web Analytics**: analyses user browsing behaviour to better customise ads selection and placement.

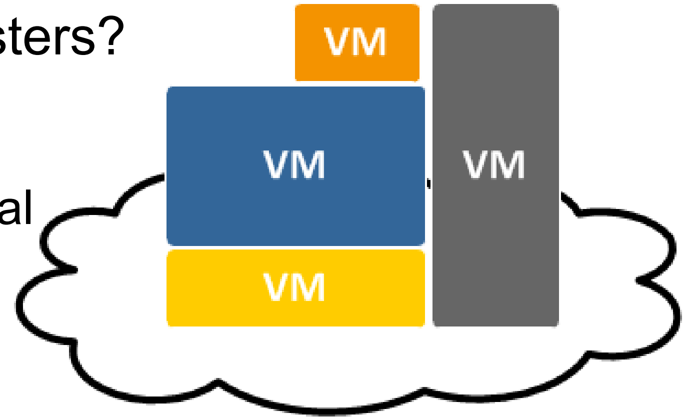
❑ **Facebook Recommendation**: analyses connections in **huge graphs of friendships** to recommend new ones.

facebook RECOMMENDATIONS

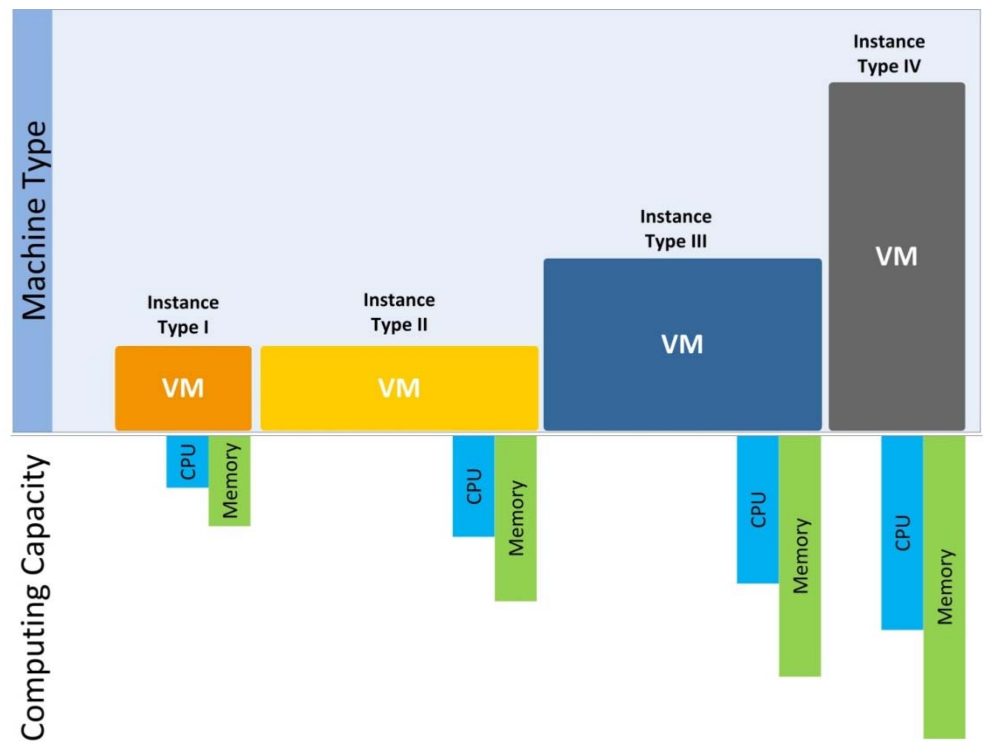
Excited enough? Or, Have you encountered the need to go for **clusters** on clouds?

› What does cloud provide us with to construct clusters?

- **Unlimited** Computing Resources **available**:
  - (Different machine types with different computational capacities)

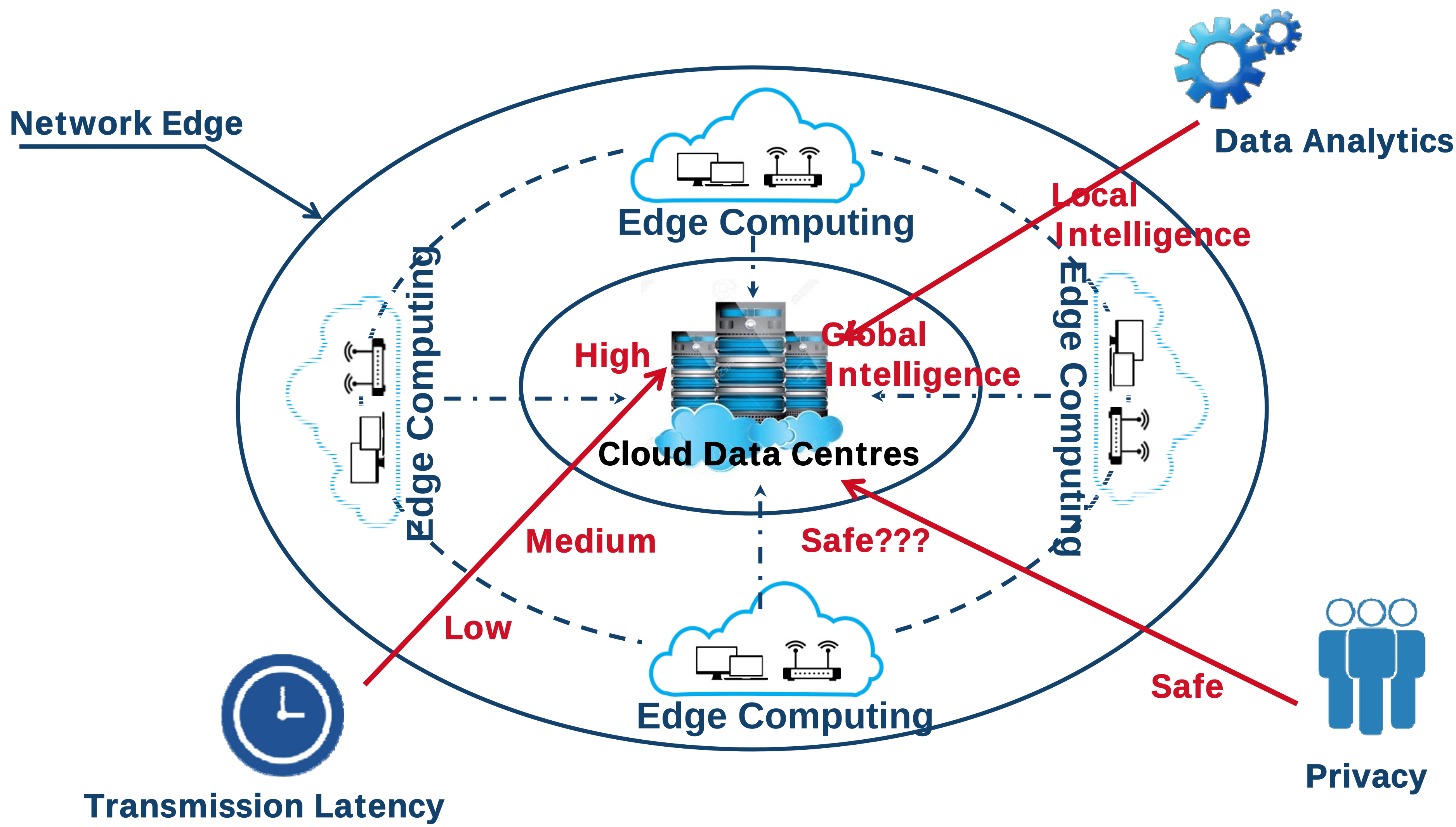


- A machine type is determined by amount of CPU/Memory.
  - It defines its **computing capability**
  - It also defines the **pricing model**, thus, controlling the **cost**.



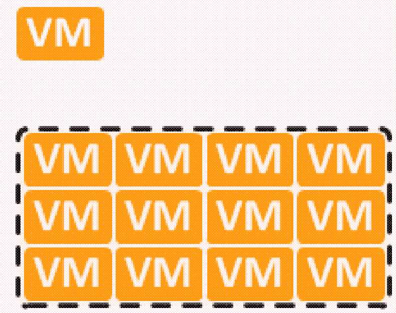
› Which one to select for a cluster ? and, Why?

A Glance of Edge Computing

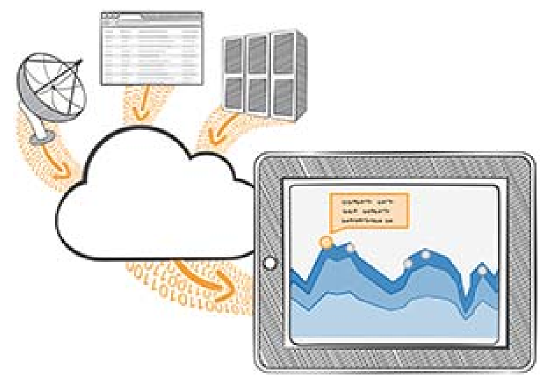


What is the current **state-of-the-art** for constructing Clusters on clouds?

- The **current practice** tends to construct **<u>homogeneous</u> clusters** that are built up using **only one instance type (VM type)**.
  - *Example*: A cluster consists of **12 instances (VMs)** of type **m3.medium** on AWS. When **auto-scaling**, it adds only the **same instance type** to the cluster when needed.



- *Examples*:
  - ***Amazon Kensis***: a platform for real-time **streaming data** on AWS cloud, offering powerful services to make it easy to load and analyse streaming data (similar to the **open-source Apache Storm**)
    - **Use Case**: Real-time stream processing Content Recommendations similar to one in **Facebook**.



  - ***Amazon Elastic MapReduce clusters (EMR)***: simplifies **Big data processing**, providing a managed **Hadoop** framework that makes it easy, fast, and cost-effective to <u>distribute</u> and <u>process</u> vast **<u>amounts of data</u>** across dynamically **scalable** Amazon **EC2 instances**.
    - **Use Case**: Batch processing conducted by **Google** to analyse users' behaviour for better ads selection and placement.

Why is constructing **Homogeneous Clusters** a common practice?

- Possible reasons are:
  - **Straightforward** and **Easy** to implement **without** the need of **informed performance-based decisions**.
  - **No** sophisticated **resource allocation mechanisms** are needed.
  - **Cheaper** for cloud providers.
  - Performance of a homogeneous cluster is predictable, simply because it is built up from identical VMs (**only one instance type** is used).
  - ***Avoiding*** the hassle of **ensuring cluster performance predictability** that would arise when considering ***heterogeneous resources*** to *allocate*.

- Example:
  - AWS and Microsoft Azure clouds **still** provide **auto-scaling templates** where *only one instance type* is allowed to be *selected* to allocate when auto-scaling is trigged.

THE UNIVERSITY OF SYDNEY

Are they Sufficient for Applications Users Needs?

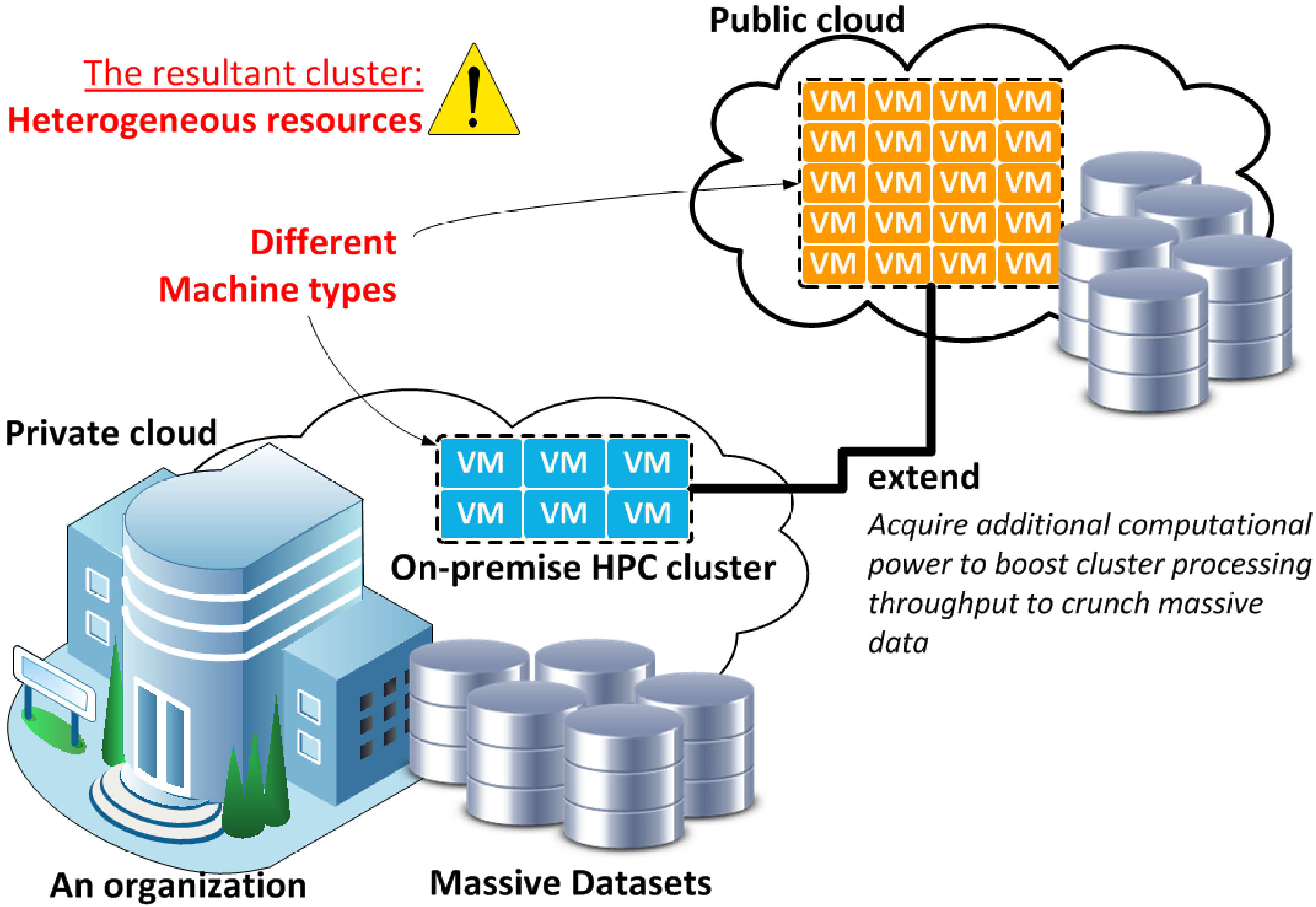
Do we need to consider ***Clusters of heterogeneous resources***? (1)

- Is this sufficient? **No**,
- Consider the following:

1. Distributed data analytics applications, feature workload patterns that have **heterogeneous resource demands**, for which, accounting for performance heterogeneity of cloud resources would be highly advantageous to the application performance.
2. Besides that, all public clouds intrinsically provide **different** machine types (heterogeneous VMs):
   - Can they be **exploited** for the benefits of application performance? e.g. to address situations where a computing cluster is no longer homogeneous, especially during/after scaling processes (scaling -in/-out).
   - Why only use a single instance type for the whole cluster?
3. Performance **unpredictability** in the cloud may lead to Service-level Agreement (SLA) **violations**:

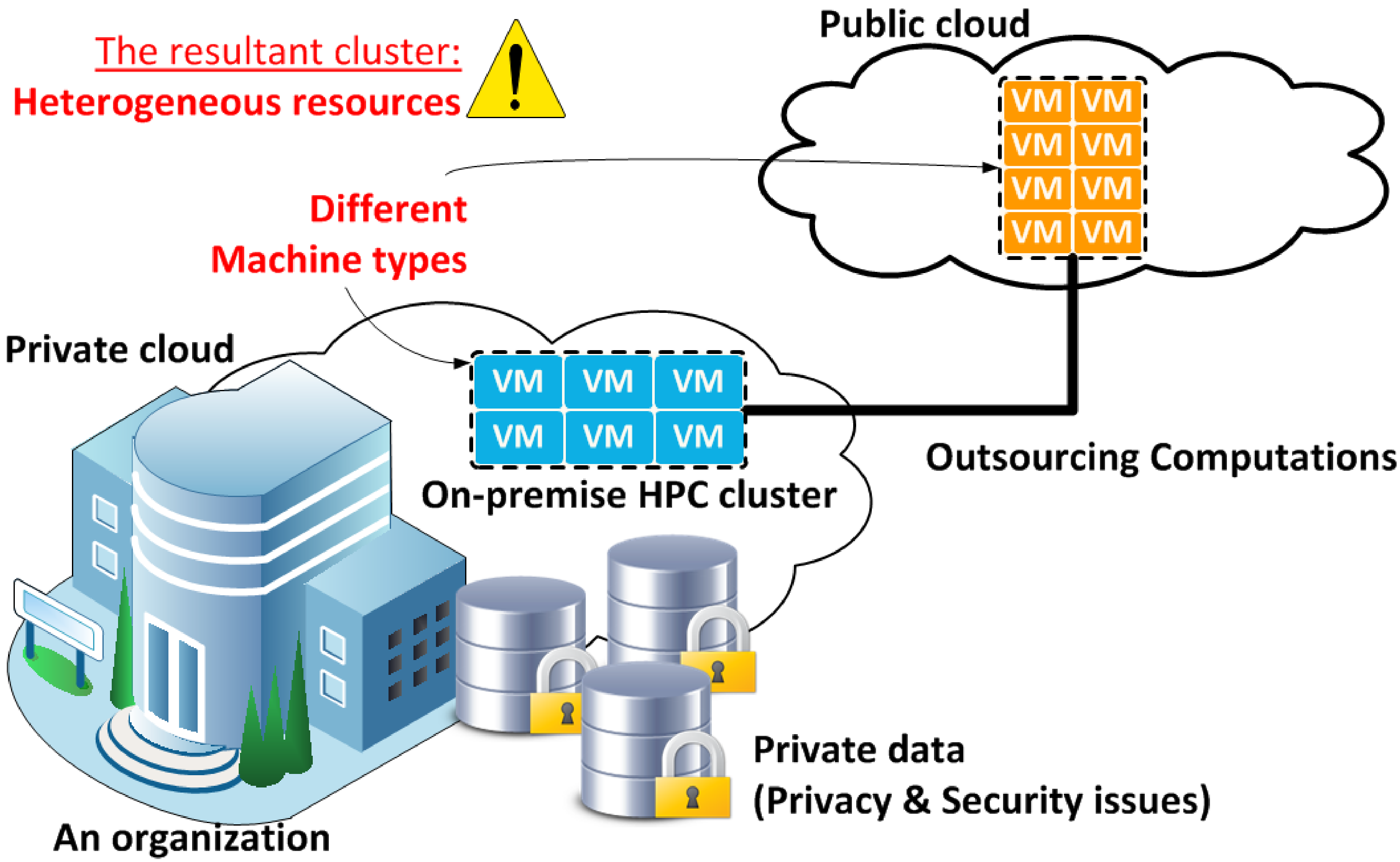
Private-to-public Cluster extension (1)

› The need to acquire additional computational capacity for your local private cloud by hiring additional resources from public clouds (to meet new deadlines, absorb surge workload changes, etc.)



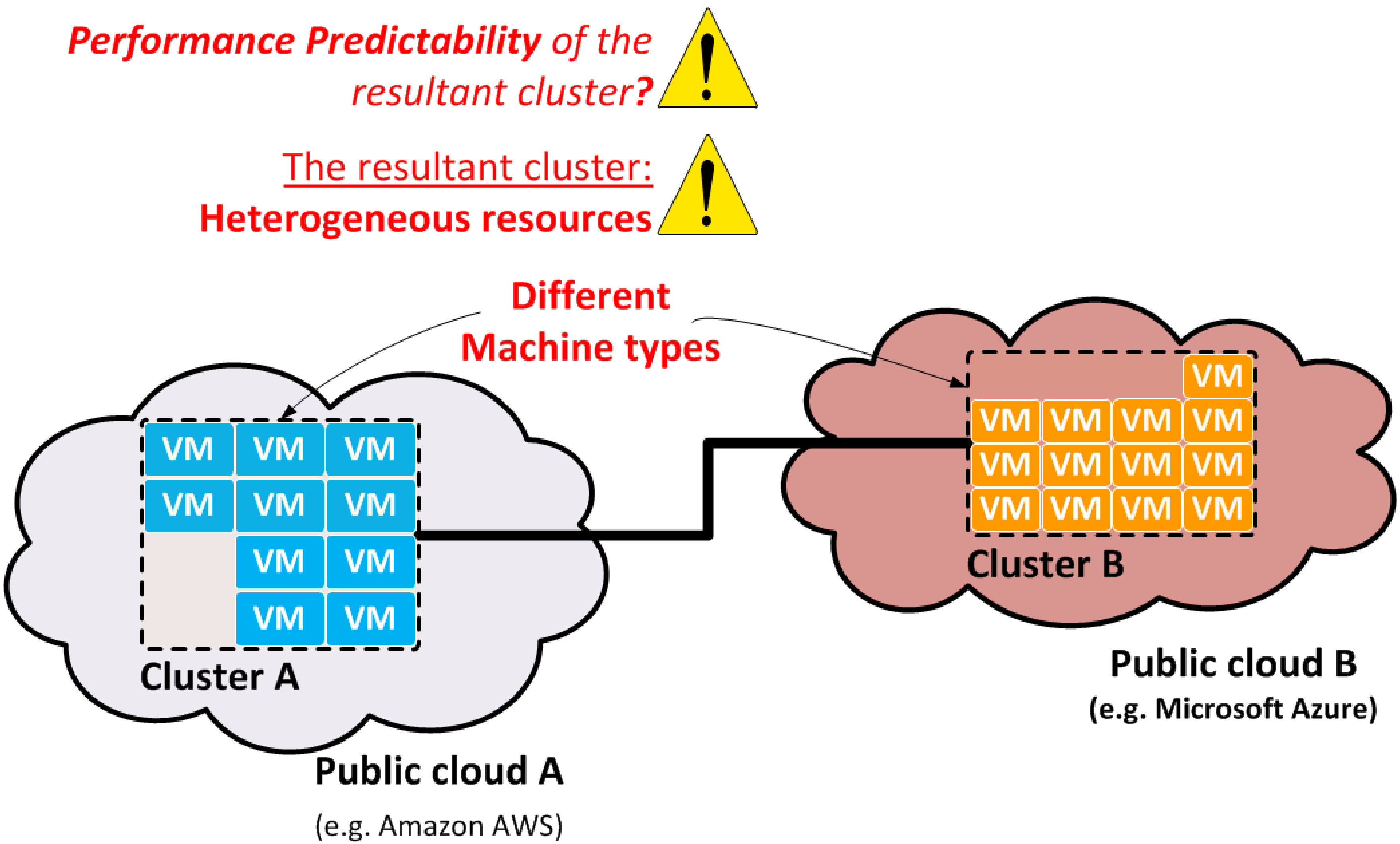
Private-to-public Cluster extension (2)

› The need to keep your private data locally on a private cloud for ensured security (with on-premise clusters for local computations), while it is urgent to hire extra resources from public cloud to boost your local cluster processing throughput for enhanced time-to-solutions.



Public-to-public Clusters merging

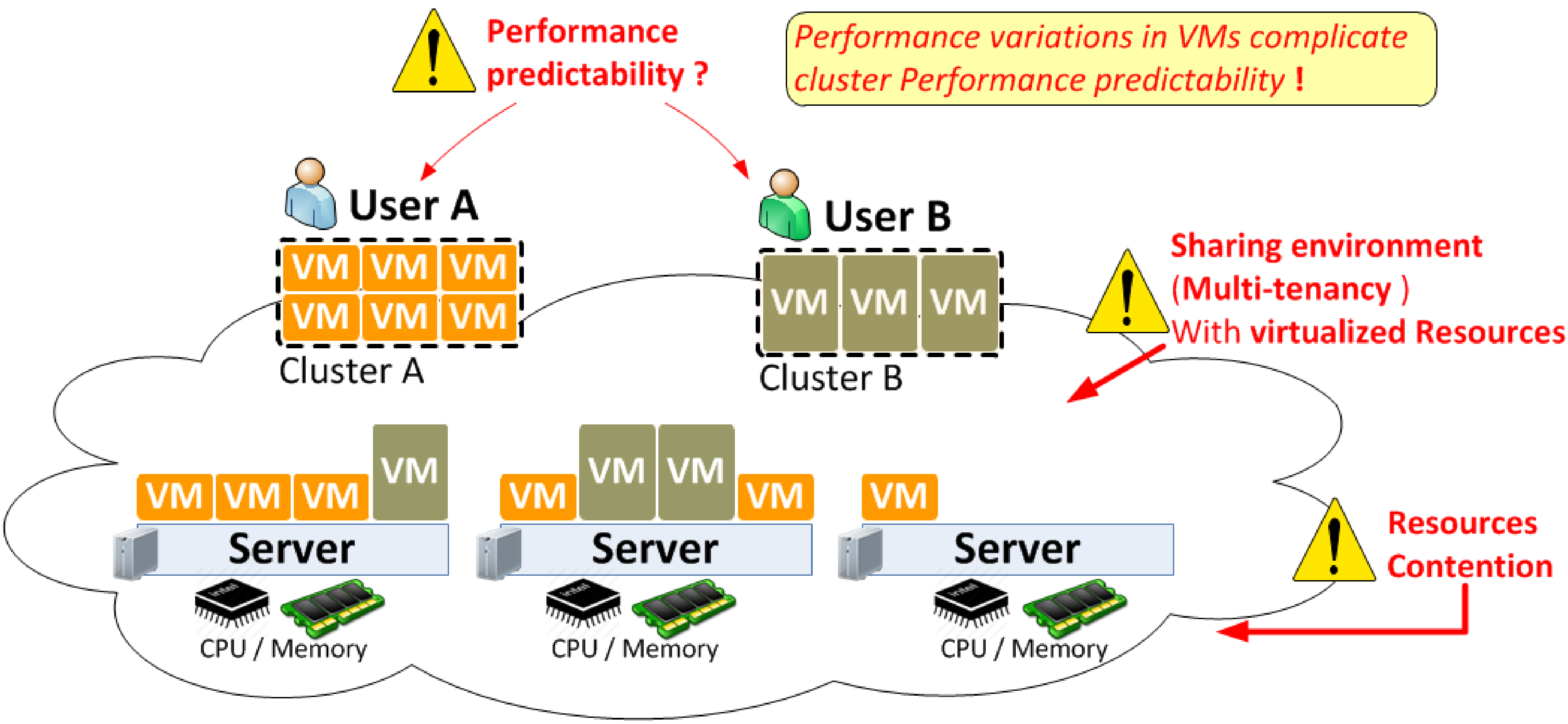
› The need to **merge** existing **homogeneous clusters**, that are possibly initially built from **different instance types,** in two or more **different** cloud providers.



Inevitable Performance Variations

- Cluster **VMs** may still exhibit **inevitable Performance Variations** due to:

1. **Multi-tenancy cloud environment**: Cluster VMs of one application are **co-located** on physical servers with VMs of other applications. As *VM placement* is unknown to application users.
2. **Resource contention**: Sharing **physical resources** of underlying servers (CPU/Memory).
3. Also, VMs of **same type** might experience **performance variations** as a single cloud provider has several datacenters across different geographical locations.



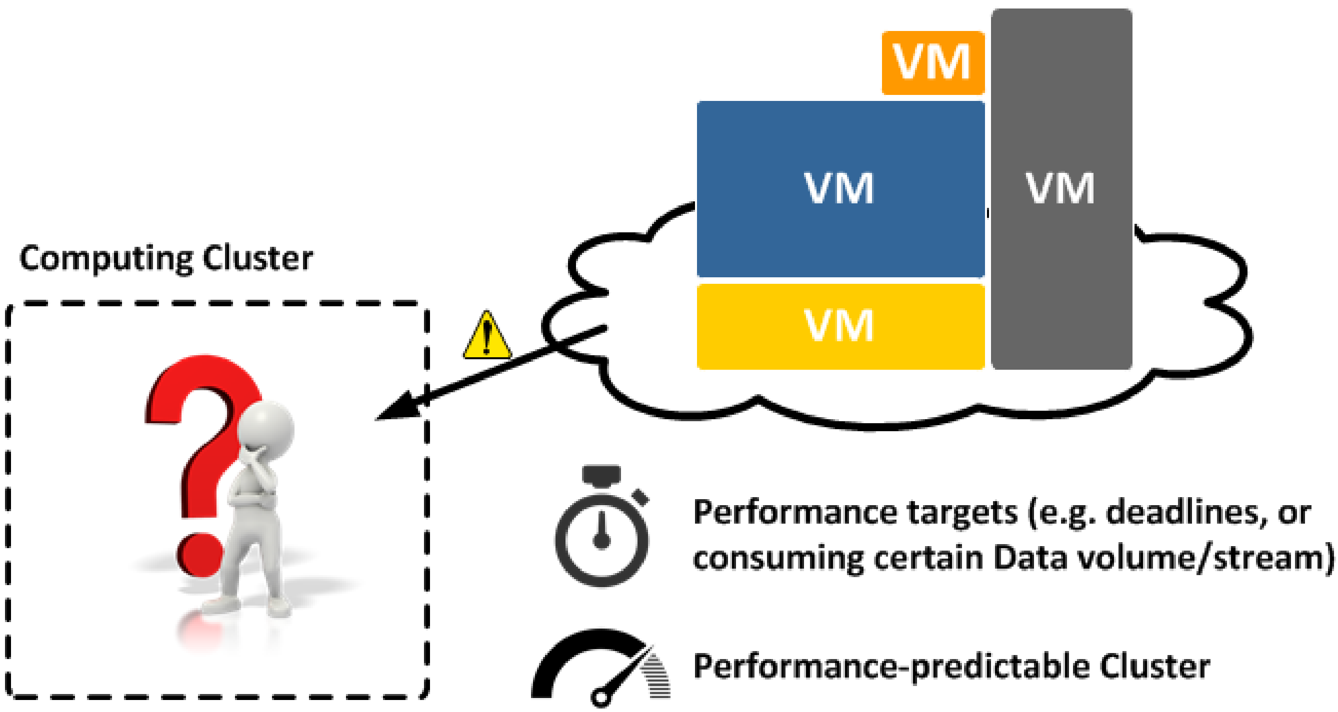
Concerns with **Homogeneous** Clusters (1): *Initial Construction of a Cluster*

› **Initial Construction**:

- Even with Homogeneous clusters, applications users **still need to make decision** on which **machine type** to select and whether it would be sufficient to construct cluster that would **meet the deadline** or **consume specified data volume in a tolerable time.**

› **For example:**

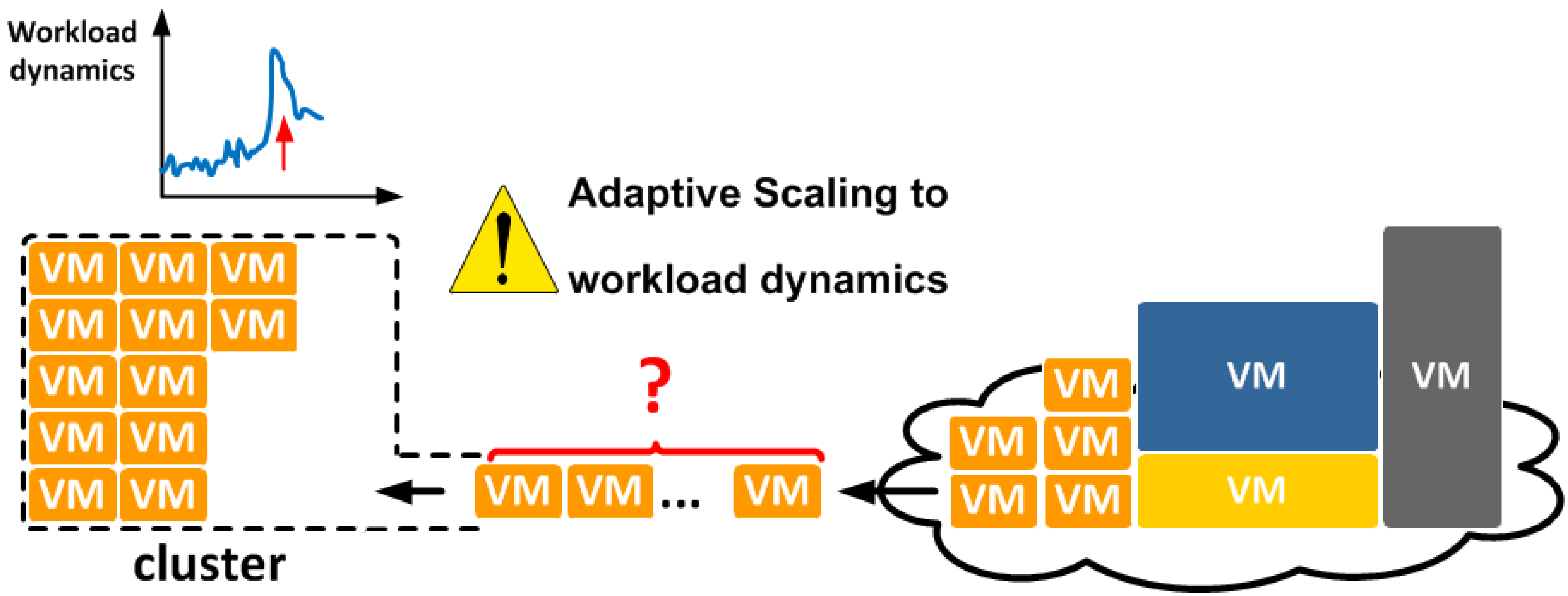
- **Cluster users** at **Google** currently **select their instances** merely based on a **combination** of **intuition**, **trial-and-error**, and **prior experience** to ***achieve their performance targets.***



Concerns with **Homogeneous** Clusters (2): ***Adaptive Scaling to workload dynamics***

› **Adaptive Scaling to workload dynamics**:

- For clusters that are constructed for long-term runs, still need a **performance-based resource allocation mechanism** to cope up with changes in workloads.



Do we need to consider *Clusters of heterogeneous resources*?

› A problem arises: **Performance Predictability** of clusters is ***no longer*** ensured.

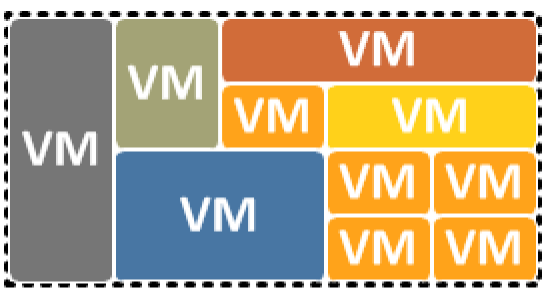
- All these scenarios end up with a cluster of VMs that are no longer homogeneous. As a result, ***the*** *cluster performance* ***has become a real challenge to*** *ensure*.
- Application users are left with burden to design and make their own decisions on constructing such clusters.

› So, we **do need** to have **performance-based resource allocation mechanisms**.

Cluster with homogeneous resources

Due to Performance-Varying Resources in the cluster:

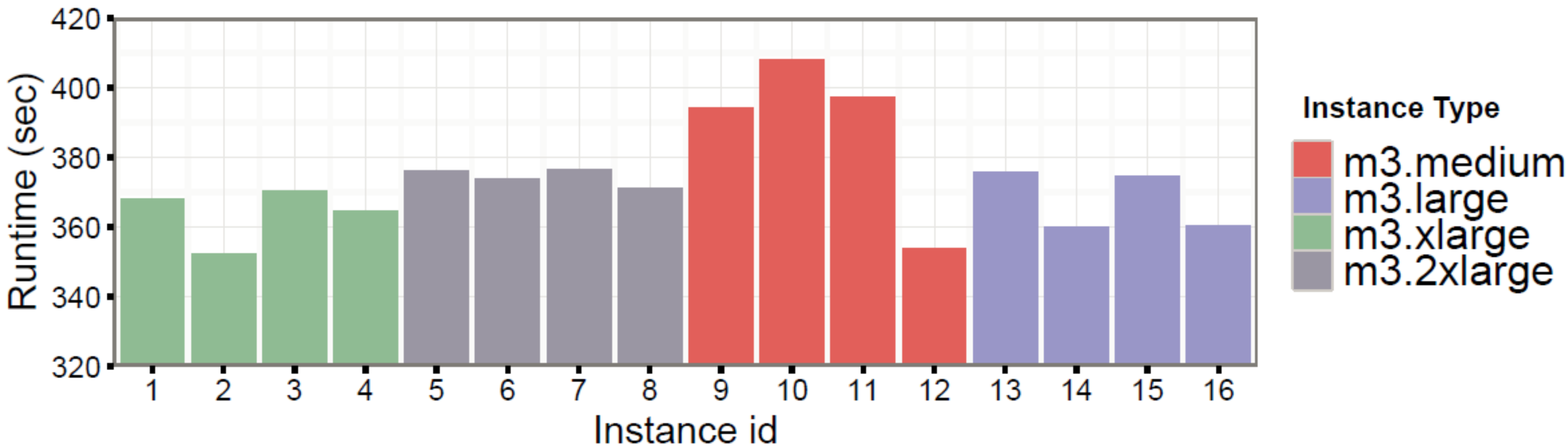
Performance Predictability?

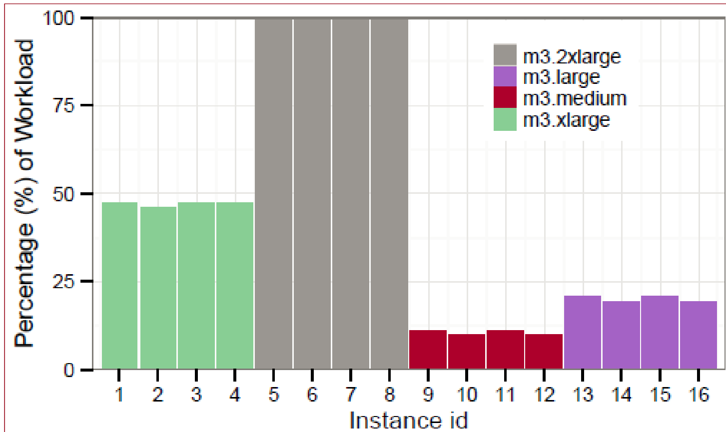


Cluster with heterogeneous resources

Example: Distributed Application on 16-node Cluster

› A MapReduce job aggregates 64GB of data in 6.83 minutes on Amazon EC2 cluster of 16 instances. **Computational Heterogeneity in cluster's nodes** influences performance of real applications and **complicates** their **performance predictability**:





Instance Type	m3.medium	m3.large	m3.xlarge	m3.2xlarge
workload assigned	5.8%	11.4%	26.4%	56.4%

Response-Table 1: Percentages are relative to the total number of map tasks dispatched for execution across the cluster, over the entire runtime.

Workload distribution

Performance Predictability of Cluster built from Performance-Varying Resources

› We frame performance variability phenomenon around three fundamental performance properties:

i. **Performance stability**: Cloud resources provisioned for application execution should remain '**constant**' over time. That is, the performance of an individual virtual instance is expected to be **stable irrespective** of other instances **sharing** the same physical host.

ii. **Performance homogeneity**: The performance of virtual instances drawn from distinct types are meant to be heterogeneous. However, instances of the ***same type*** are supposed to ***perform similarly***.

iii. **Performance predictability:** Computing instances provisioned on clouds would **not be reliable** if the performance of user application **cannot be predicted**. This predictability necessarily requires **both *stability*** and ***homogeneity*** in the performance of instances provisioned.

The Research Problem

› We address **resource allocation problem** of **data analytics clusters** in the cloud. For a **given performance target**, searching for the optimal combination from **performance-varying heterogeneous resources**, is **no ordinary** task.

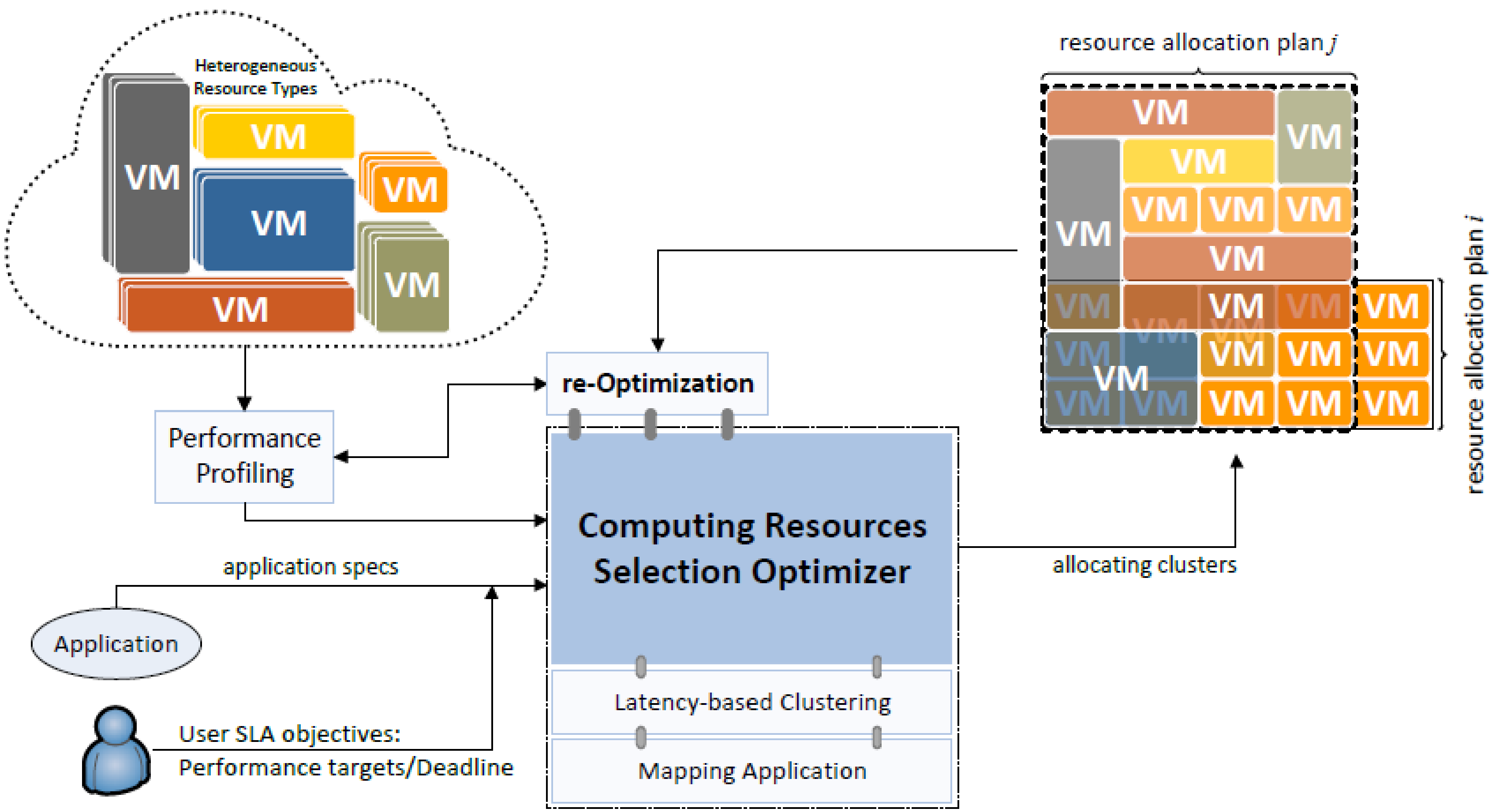
- Given the **computational characteristics** of various types of cloud instances available:
  - We generate a **resource allocation plan** that defines a **compute cluster** of **mixed-type instances** such that the **targeted average cluster performance** is **predictable** and thus **attainable**.

› In addition, a previously generated **resource allocation plan** is **re-optimised** during application execution to adjust to **new performance requirements** as it is dictated by **changes** in **resource demands**.

Proposed Resource Allocation Mechanism

› We develop a **resource allocation mechanism** that **benefits** from the **heterogeneity of cloud infrastructures** to attain **user-defined average cluster performance**.

› The selection of appropriate combination of instances provides the **necessary and sufficient computing capacity** to handle performance **under-/over-estimation** and workload **fluctuations**.



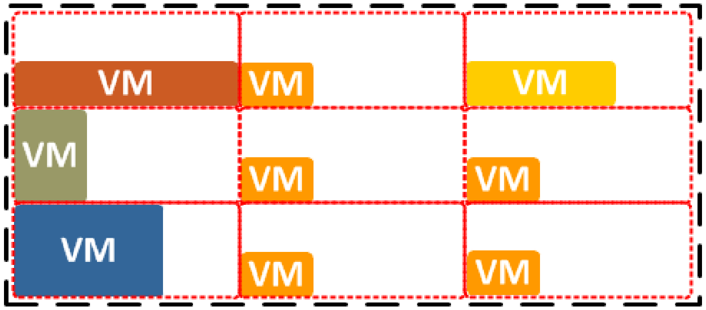
A unified performance metric for a Cluster: *Overall Average Performance* in MIP model

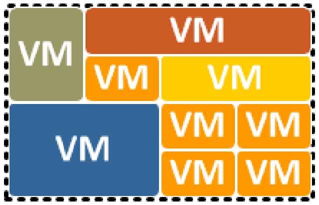
› The variation in performance observed by a distributed application can be perceived as a collective result of micro variances occurring at different system levels; including performance variabilities in CPU, memory, disk I/O, and network.

› For a cluster, we wanted a measure that can cover these variabilities from each instance **individually**, and at the same time, can capture the variability across all instances **altogether**.

› As a result, we propose a unified performance metric, namely **Overall Average Performance (û)** for a cluster:

- It works in a way as if each instance in a cluster has the **same** average performance as specified by **û.**
- It alleviates the complexity that arises by considering **resources heterogeneity** in allocation, as it simplifies the calculation of the predicted performance of a cluster.
- It promotes performance predictability of cluster of heterogeneous resources.





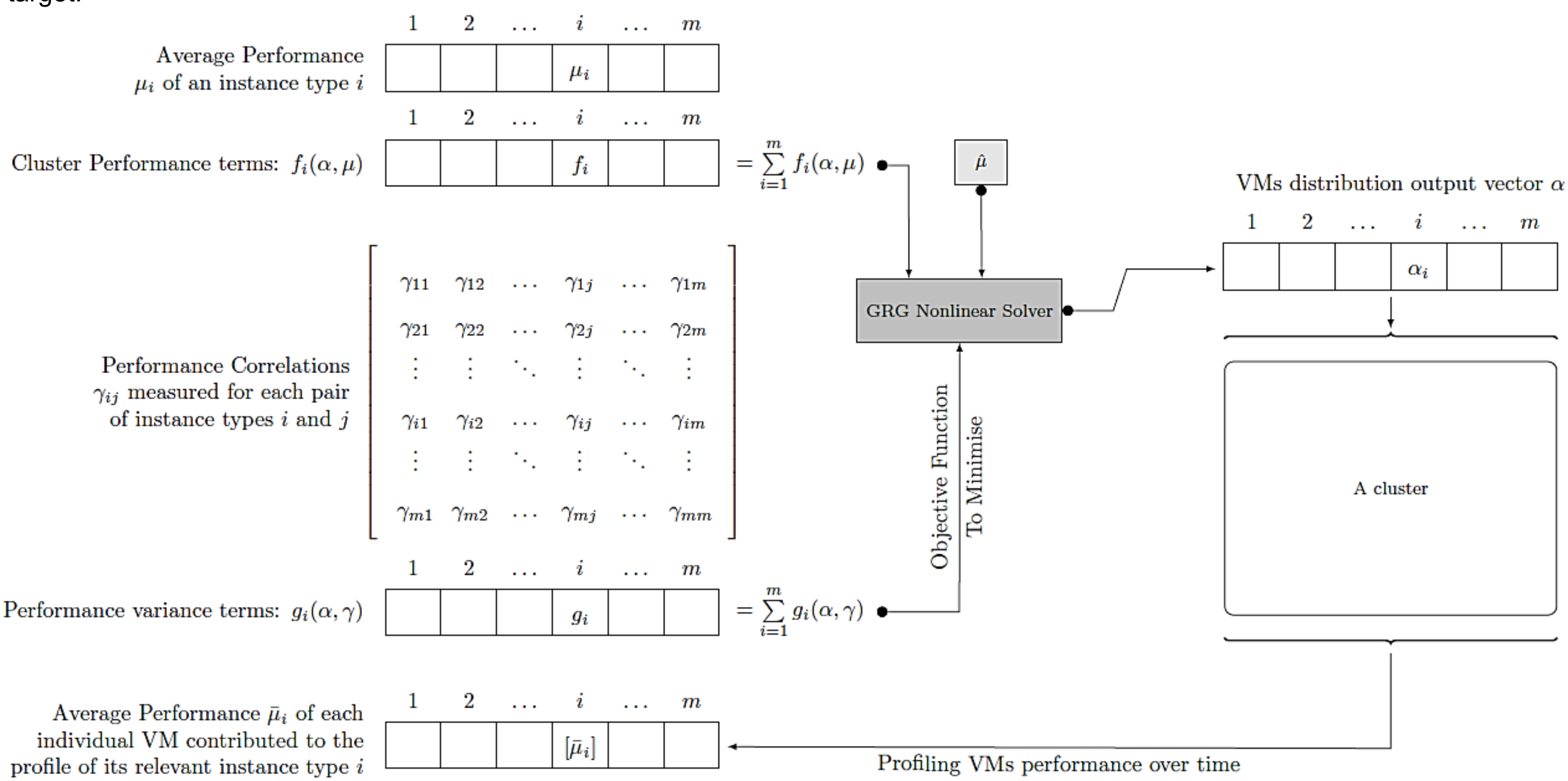
› The QUESTION is:

- ***How to translate a given target performance to a <u>resource allocation plan</u>, to construct a cluster from <u>heterogeneous resources</u> taking into account their <u>performance variabilities</u> ?***

Problem Formulation: Resource Selection Optimisation (MIP model)

› We devise a **Mixed-Integer Programming (MIP) model that** represents **resource selection optimisation problem.**

- **Exploiting the relationships** observed between **distributions of cloud instance types in a cluster**, **their individual average performances**, and the **overall average performance of the enclosing cluster**, we were able to formulate an ***optimisation problem model*** that can **search** for the ***optimal blend of instances from different types*** of different performance variabilities.
- By **utilising correlations** existing amongst distinct instance types' performances, the model searches the **solution space** to locate the **optimal combination of heterogeneous machines** in the cloud **to construct a compute cluster** that can achieve a given performance target.



THE UNIVERSITY OF SYDNEY

Evaluation

- Initial Performance Profiling and Demonstration of Measured and Predicted Cluster Performance.
- Extensive Simulations: Evaluating the Effectiveness of the Proposed Model.
- Experiments on Amazon EC2.
- Comparison Experiments with existing Solution.

Initial Performance Profiling, and Demonstration of Measured and Predicted Cluster Performance

	Instance Type	vCPU	RAM	Disk	#instances
Private Cloud	Type I	1	3.7GB	16GB	1
	Type II	2	7.5GB	16GB	3
	Type III	8	8GB	16GB	2

	Instance Type	vCPU	RAM	Disk	#instances
Public Cloud	m3.medium	1	3.7GB	100GB SSD	4
	m3.large	2	7.5GB	100GB SSD	4
	m3.xlarge	4	15GB	100GB SSD	4
	m3.2xlarge	8	30GB	100GB SSD	4

	Private Cloud
virtualisation	VMWare vSphere v5.5.0
OS	64-bit CentOS 5.6, kernel 2.6.32-431
VM Placement on Physical Servers	2 physical Servers with: 3.4GHz Quad Core Intel i7 CPU-hyperthreading enabled, 8GB RAM, and 460GB HDD.
Cluster Size(# instances)	6
Benchmark Application	PARSEC.streamcluster (in C++) online clustering application for input stream; (A representative for class of streaming data mining applications)
Programming Framework	C++ and associated Libs on native Linux OS
Data Size (GB)	36
Cluster Management and deployment	n/a

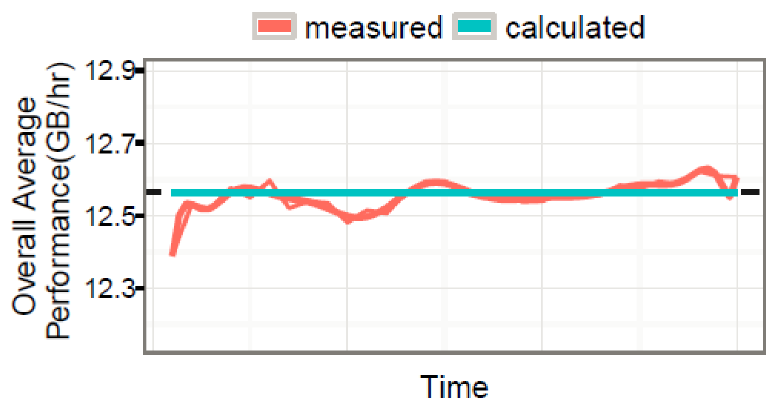
Public Cloud
AmazonAWS-specific
64-bit Ubuntu Server OS 14.04 LTS
Availability Zone us-east region
16 + 1 m3.xlarge (as master node with 160GB SSD)
MapReduce (WordCount) (batch processing)
Apache Hadoop v2.5.0 with HDFS and MapReduce (v1) framework With default FairScheduler
64 (uploaded and read from HDFS)
Cloudera Manager

Initial Performance Profiling, and Demonstration of Measured and Predicted Cluster Performance

› Average performance **measured** for each **instance type**:

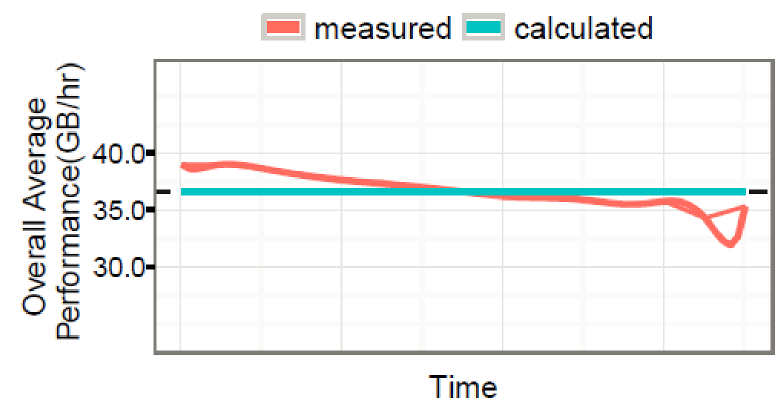
Machine Type	Type I	Type II	Type III
mean performance (GB/hr)	5.12	9.42	20.97

Machine Type	m3.medium	m3.large	m3.xlarge	m3.2xlarge
mean performance (GB/hr)	9.16	17.92	40.69	78.54



Proximity of predicted and measured performance of a cluster of 6 heterogeneous VMs. The measured data are smoothed for better presentation.

Calculated Overall Avg Perf = 12.56 GB/hr



Performance of a cluster of 16 heterogeneous EC2 instances, and is predicted at 36.58GB/hr. The measured data are smoothed for better presentation.

Calculated Overall Avg Perf = 36.58 GB/hr

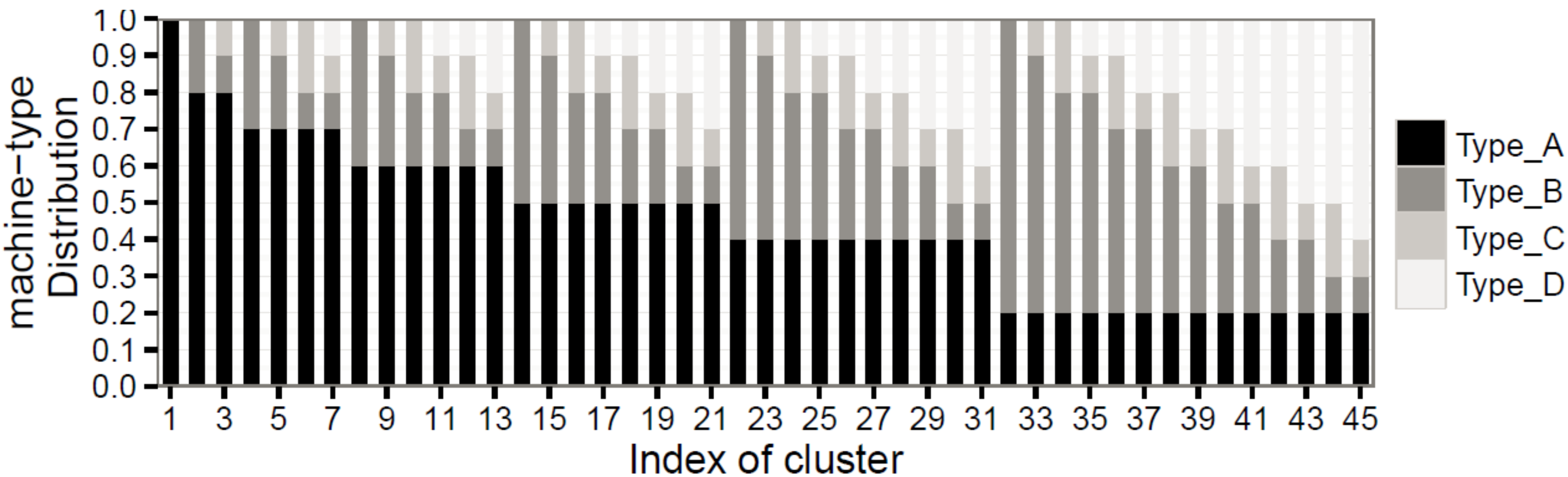
› Each **machine type *contributes*** its computing capacity (performance capacity) to the **cluster capacity** (that defines its **overall cluster performance**). The **contribution** of each instance type is **proportional** to its number of instances it has in the cluster.

Experimental Simulations

› Simulation setup (A cloud with 4 different types of virtual instances):

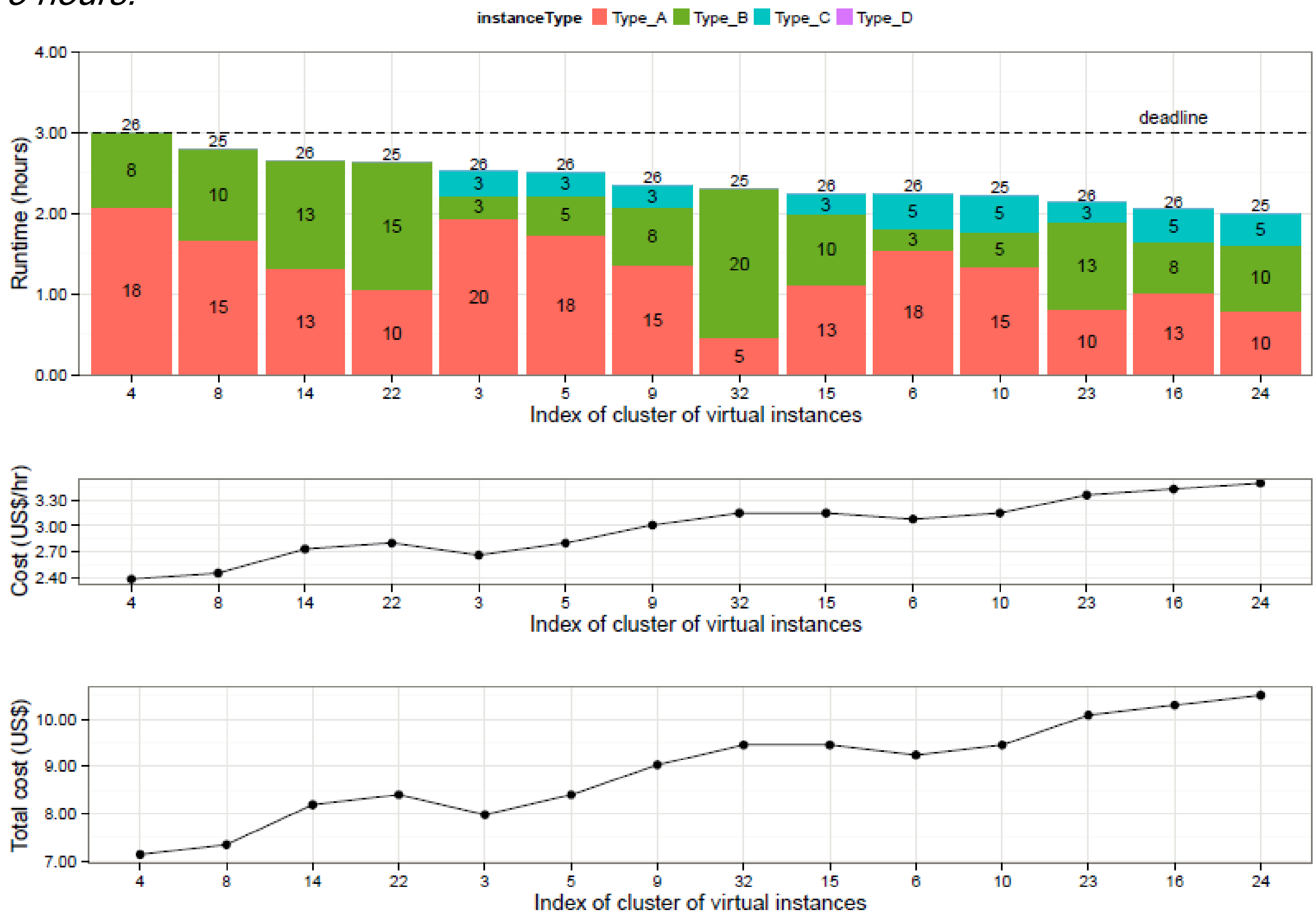
Machine Type	A	B	C	D
mean performance (GB/hr)	5.12	9.43	20.9	40.6
stdev performance (GB/hr)	1.2	0.89	0.5	0.31
cost unit (US$/hr)	0.07	0.14	0.28	0.56

› For simulation purposes: we predefine *45 resource allocation plans* to be used in our Simulator (implemented in R) :



Reckless Decisions Effects on Application Runtime

› **<u>Goal:</u>** *To run computations on a dataset of 500GB, over a 25-nodes cluster, and to complete execution in 3 hours.*



Searching For Optimal Allocation Plan and its Alternative Plans (1)

- **<u>Goal:</u>** *To run computations to consume 500GB of data and to complete in 3 hrs. We are to construct 25-nodes cluster such that the average Cluster performance is predicted to be around **6.667GB/hr**.*
  - Using the Generalized Reduced Gradient nonlinear optimisation engine (GRG solver) with all input data needed for the optimisation model.
    - **Output:** optimal plan is (64,36)% from type-A and type-B.
      → 16 typeA + 9 typeB

Measured average cluster performance (GB/hr) (on average)	
Optimal plan	6.674
alternative plans	6.941

- Under **Controlled experiments**, we repeat experiments seeking **alternative plans** with **similar performance** as the optimal (over-allocation margin is only 1-2 instances)

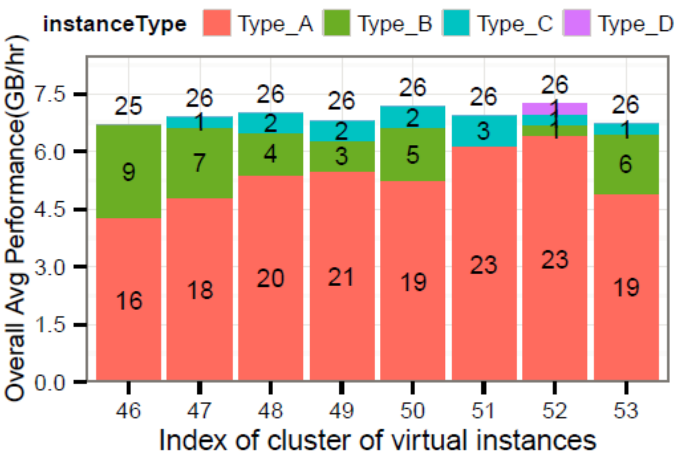


Fig. 10: Allocation plan 46 is the original optimised plan, and the others are alternative plans found by the model under controlled experiments.

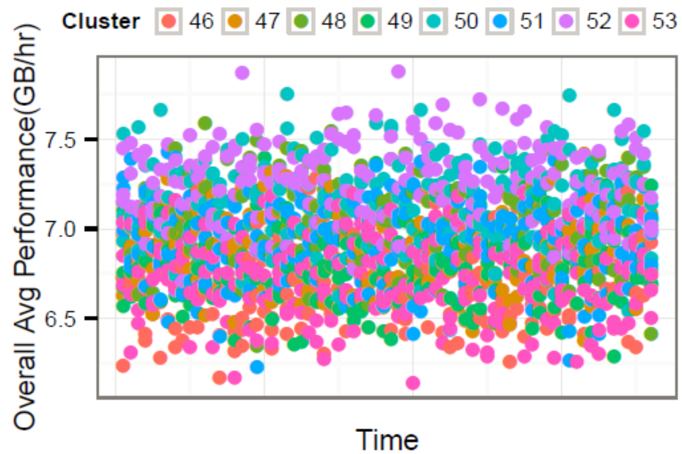


Fig. 11: Cluster of the allocation plan 46 and its alternatives are in execution. Similarity in performance is observed as predicted by the model.

Searching For Optimal Allocation Plan and its Alternative Plans (2)

› **<u>Runtime and Cost of the Optimal Plan and its alternatives:</u>**

- All allocation plans complete the job **just on time** and **close enough to the deadline of 3 hrs**.

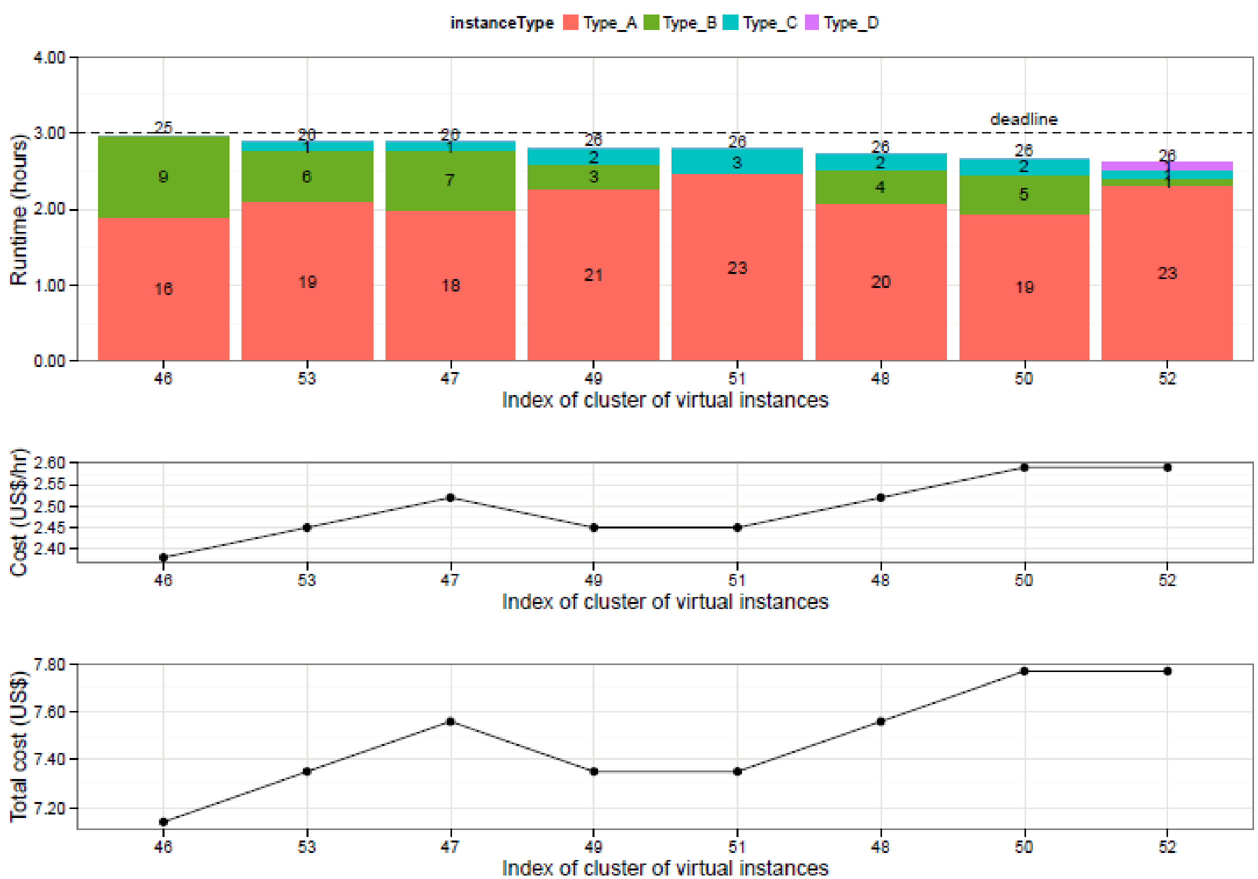


Fig. 12: Running computation job on 500GB data with deadline of 3 hrs on clusters defined by optimal allocation plan 46 and its alternative plans 47-53.

› Optimal plan 46 is <u>superior</u> because it allocates ***just the necessary resources*** to attain performance target and hence deadline is fulfilled, and that ***explains how close it is to the deadline.***

› Plan 46 has **lowest total cost of USD7.14** (optimised to allocate **cheaper** instances and **refrains** from allocating excess computational power).

Optimality of clusters: Provisioning necessary and sufficient Resources

› Clusters found perform within 95% of the optimal clusters' performance*.

- Our proposed mechanism always allocates the necessary computing nodes ***just enough to complete*** the job on time, given data size to process and the deadline to meet.

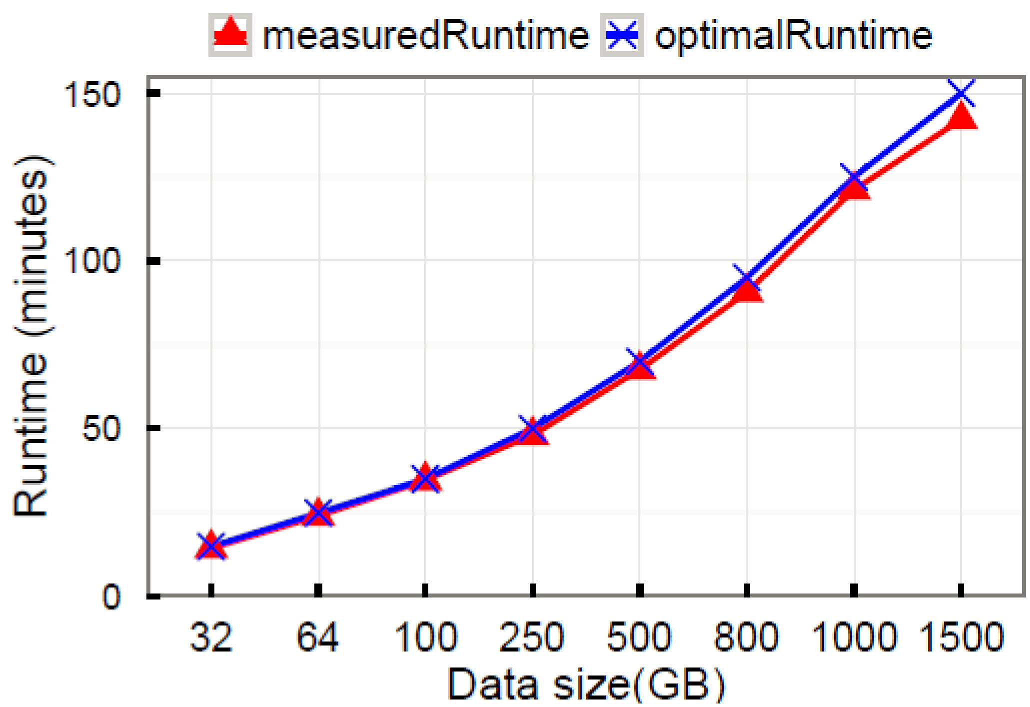


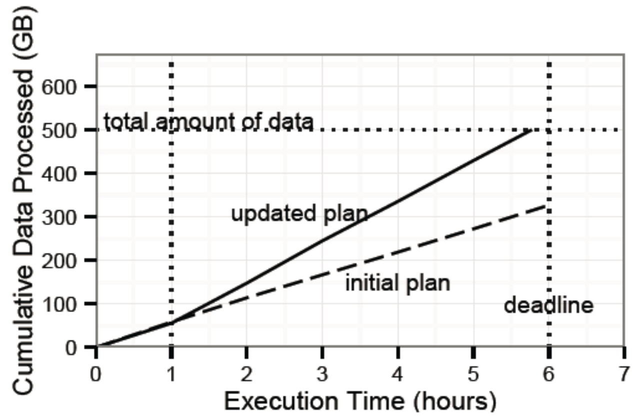
Fig. 13: Performance of eight 20-node clusters with optimised machine-type distributions.

*An optimal cluster is that one that can process given data and **completes exactly on** the given deadline.

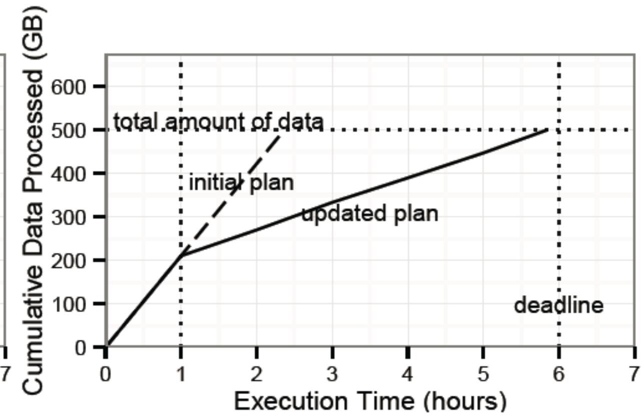
Adapting to Performance Variations and Mispredictions

- We consider cases where users **under-** or **over-estimate** performance of computing resources to request from cloud:
- Two sets of experiments to process 500GB of data in 6 hrs, on a cluster of 8-nodes with two plans of different instance-type distributions.

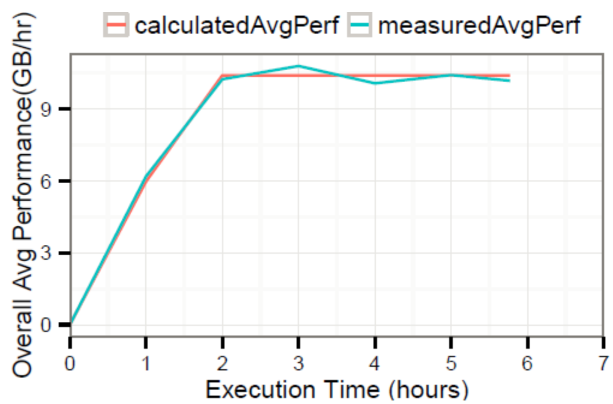
		Measured average cluster performance (GB/hr)	runtime (hrs)
a	**default plan** 7A+2B=9	6.03	9.26
	Optimised plan 3A+4B+2C=9	10.4	**5.771**
b	**default plan** 2A+2B+1C+4D=9	25.3	2.367
	Optimised plan 5A+4B=9	6.8	**5.849**



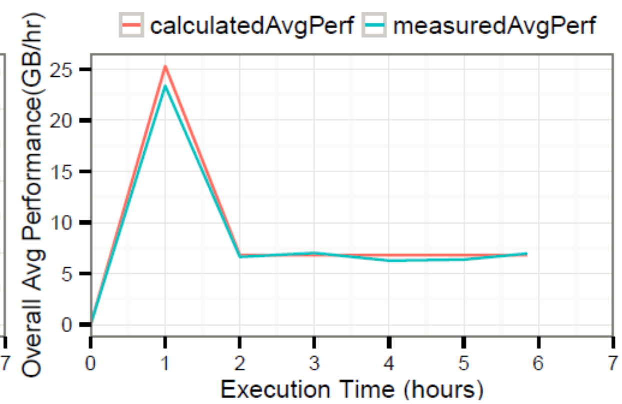
(a) Initial default plan 2: (80,20)% for types (A,B). Updated optimised plan: (33.1,45.8,21.0)% for types (A,B,C).



(b) Initial default plan 43: (20,20,10,50)% for types (A,B,C,D). Updated optimised plan: (61,39)% for types (A,B).



(a) Initial plan 2 has overall performance of 5.982GB/hr, and it is adjusted to 10.402GB/hr as the optimised updated plan predicts.



(b) Initial plan 43 has overall performance of 25.3GB/hr, and it is adjusted to 6.8GB/hr as the optimised updated plan predicts.

Constructing Compute Clusters on Amazon EC2 (1)

› **Goal:** *We are to find and run cluster(s) of **10 instances** that is able to process an input data size of **90GB** and completes in just **10 minutes***.

- We set up our MIP model with the GRG Nonlinear solver to search the solution space seeking allocation plans for optimal clusters with appropriate distributions of machine types such that an overall average performance of 54GB/hr is predicted to be achieved.

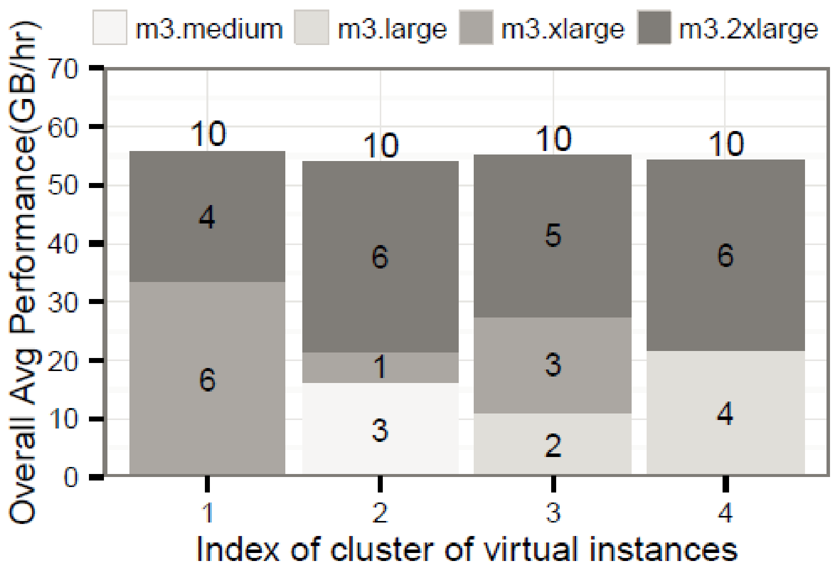


Fig. 15: 10-node optimal clusters on Amazon EC2 of different instance-type distributions but of similar overall mean performance predicted at 54GB/hr.

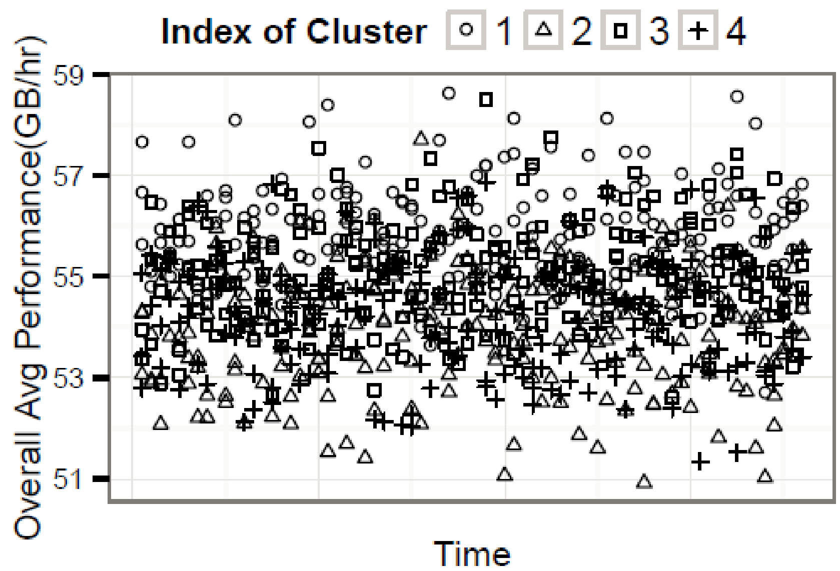


Fig. 16: Similar performance is speculated around 54GB/hr. Each point in chart is an average of all nodes' performances.

Constructing Compute Clusters on Amazon EC2 (2)

- Practically run the **optimised clusters** (of indices **3** and **4**), and compare their performance to a **homogeneous cluster** of 10-node from type **m3.large**.
  - We construct: a 10-node cluster (as specified by optimised allocation plan 3) + 1 m3.xlarge master node
  - Experiment is repeated 3 times and averages are taken.
  - To meet deadline, the **homogeneous cluster** must be **scaled-out** with **additional 20** m3.large **instances**.

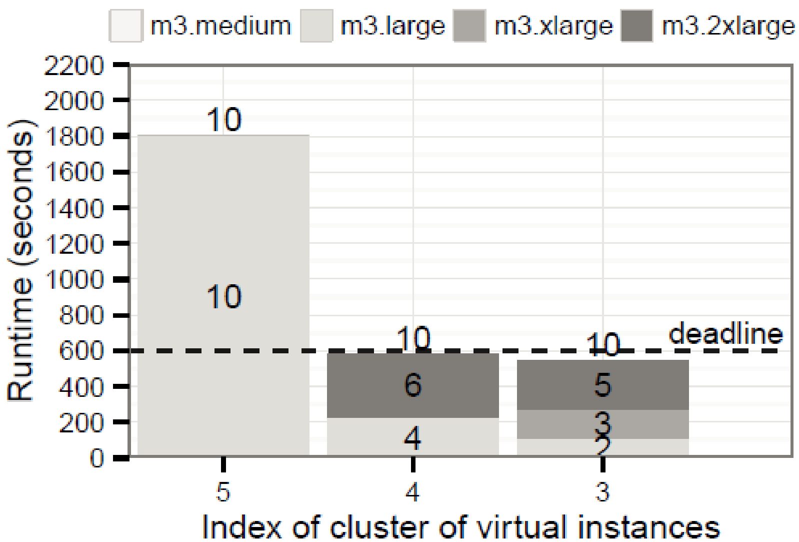


Fig. 17: Optimised clusters 3 and 4 took 9.1 and 9.69 min, respectively, to process 90GB data and meet 10-min deadline on EC2. Default cluster 5 took ∽30min.

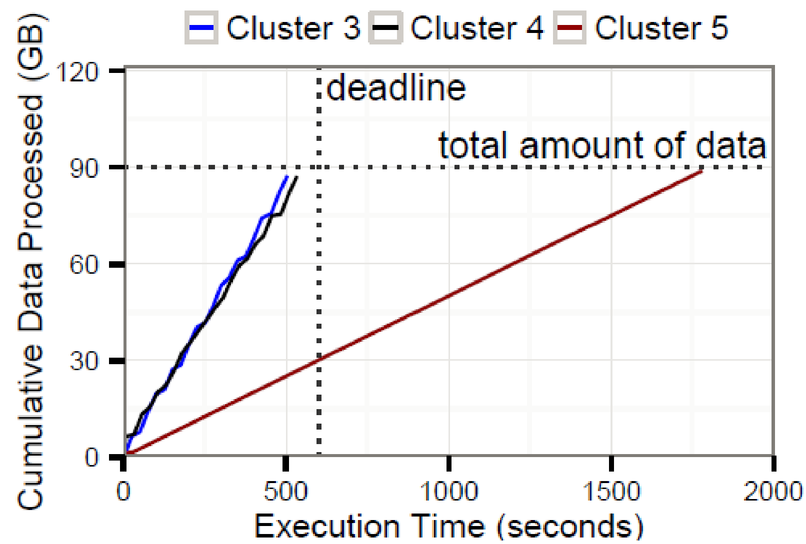


Fig. 18: Optimised clusters 3 and 4 perform as predicted and meet deadline. Default cluster 5 performs at rate of 18.07GB/hr and took 3x deadline.

Comparison with Existing Allocation Approach

› **LP-Conductor**: appeared in "*Orchestrating the Deployment of Computations in the Cloud with Conductor*", USENIX Symposium on NSDI 2012.

- It addresses the problem of allocating resources for computations.
- It proposes a linear programming model with an **linear** objective function representing total execution cost.

› We implement the LP-Conductor, and use **Simplex LP solver** to solve it.

- An assumption is made:
  - Since we can only process input data in the cloud that has already been uploaded, we assume that the entire input data are already uploaded onto the cloud.

› **Goal:** *Using the same cloud, we are to construct 10-node clusters to process 500GB of data with a deadline of 8 hours.*

- The idea is that **each model** tries to allocate instances for the cluster from different types to process an amount of data evenly distributed across a maximum of 8 time intervals of 1-hr length each.

MIP-model *vs* LP-Conductor (1)
Under different levels of Performance Variations

› We induce performance variations of all instance types by systematically increasing their standard deviations (σ) all at once by 0-90%, i.e. σ(1+ξ).

- Our MIP model exhibits high resilience to performance variability. It consistently readjusts cluster composition by allocating proper mixture of instance types that can hand performance variation.
- LP model is inconsistent in its allocation decisions when dealing with performance variations in cluster VMs to maintain reliable steady processing rate.
- As for MIP model, over than 98% of clusters were able to process data within the deadline, and those very few clusters that had missed it, did complete just a few minutes beyond the deadline.
- Execution time of some clusters allocated by the LP-Conductor completes *3+ hours beyond* the deadline.

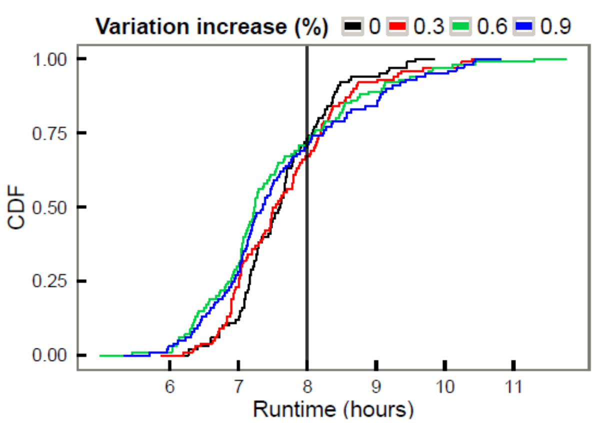


Fig. 19: CDF of runtime of 10-node clusters allocated by the Conductor LP-model to process 500GB in 8 hrs.

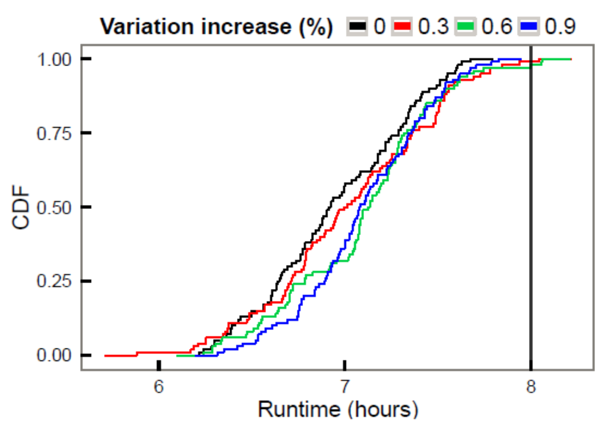


Fig. 20: CDF of runtime of 10-node clusters allocated by our MIP-model to process 500GB in 8 hrs.

ξ (%)	Deadline (8hrs)							
	Meet		Miss					
	%		%		average (hrs)		max (hrs)	
	MIP	LP	**MIP**	LP	**MIP**	LP	**MIP**	LP
0	100	73	0	27	-	8.52	-	9.57
30	99	67	1	33	8.04	8.66	8.04	10.39
60	98	72	2	28	8.05	8.93	8.06	11.30
90	100	70	0	30	-	8.99	-	10.44

TABLE 3: Detailed information of clusters performance for both LP-Conductor and MIP models.

MIP-model *vs* LP-Conductor (2)
Under different Data volume sizes to Process

- **LP-Conductor:**
- Overall, clusters allocated by LP model exhibit much longer runtime that led to very high deadline-miss rate.
  - For ξ=0, as data size increases, LP-driven clusters had experienced lower miss rate (It allocates more than **75% of instances from type-D** which has the **least** performance variation).
  - For ξ=1, miss rate **rises again as soon as** instances **suffer higher variation** in performance with **doubled** performance variations
- **MIP-model:**
  - Our MIP model maintains **robust behaviour** in **consistent** manner **regardless** of the level of performance variations imposed.
    - That is because its core allocation mechanism considers **performance correlations** amongst all instance types involved and re-adjusts clusters accordingly to maintain **steady** processing rates to meet deadlines.

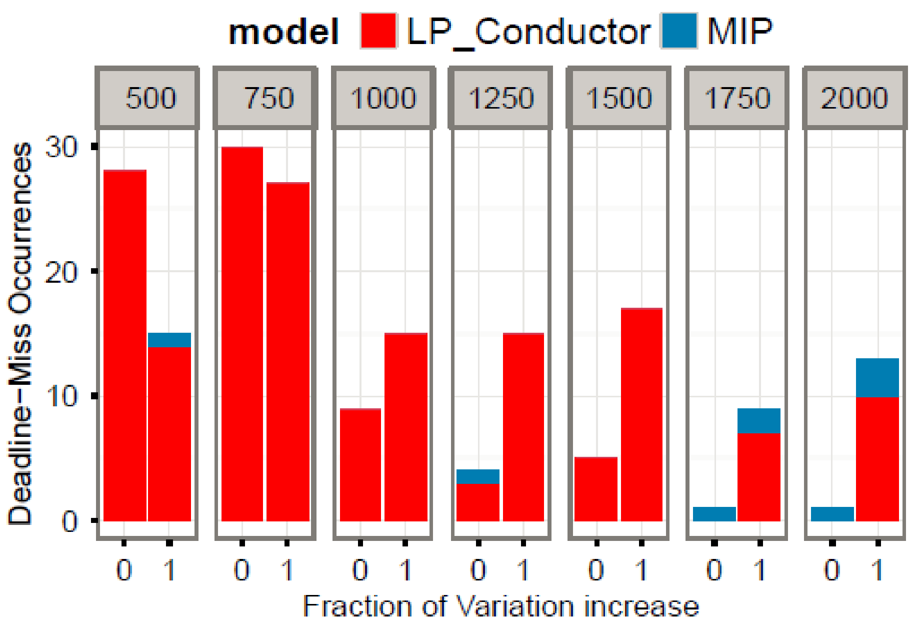


Fig. 21: Deadline miss w.r.t. perf. variations of 0% and 100% (0 and 1).

Conclusion and Future Work

- **Performance predictability** is **a major concern** in current clouds where **heterogeneous resources** are to be allocated.
- We propose a **resource allocation mechanism** that jointly addresses the phenomena of **performance variations** and **promotes performance predictability** of **computing clusters**.
- We define a **realistic performance metric** at the application level to collectively and **truly represent performance variances occurred** at all system-level components.
- The **optimisation model** incorporated in our mechanism considers **correlations of performance variabilities** occurred across and within all instance types offered in the cloud.
- The mechanism computes optimal **resource allocation plans** of **compute clusters for data processing** by allocating optimal blend of machines of different types that can assure performance predictability.
- We plan to extend the **resource allocation optimisation model proposed** in this paper incorporating a **trade-off frontier** between cluster performance and associated costs. It can be used to make prudent allocation decisions.
- Adam, O., Lee, Y.C., Zomaya, A.Y., “Constructing Performance-Predictable Clusters with Performance-Varying Resources of Clouds,” *IEEE Transactions on Computers* (to appear).

Other recent work

- Qingye Jiang, Young Choon Lee, Albert Y. Zomaya: Price Elasticity in the Enterprise Computing Resource Market. **IEEE Cloud Computing** 3(1): 24-31 (2016)
- Huawei Huang, Song Guo, Peng Li, Weifa Liang, Albert Y. Zomaya: Cost Minimization for Rule Caching in Software Defined Networking. **IEEE Trans. Parallel Distrib. Syst.** 27(4): 1007-1016 (2016)
- Rajiv Ranjan, Joanna Kolodziej, Lizhe Wang, Albert Y. Zomaya: Cross-Layer Cloud Resource Configuration Selection in the Big Data Era. **IEEE Cloud Computing** 2(3): 16-22 (2015)
- Rajiv Ranjan, Lizhe Wang, Albert Y. Zomaya, Dimitrios Georgakopoulos, Xian-He Sun, Guojun Wang: Recent Advances in Autonomic Provisioning of Big Data Applications on Clouds. **IEEE Trans. Cloud Computing** 3(2): 101-104 (2015)
- Lizhe Wang, Yan Ma, Albert Y. Zomaya, Rajiv Ranjan, Dan Chen: A Parallel File System with Application-Aware Data Layout Policies for Massive Remote Sensing Image Processing in Digital Earth. **IEEE Trans. Parallel Distrib. Syst.** 26(6): 1497-1508 (2015)

Thank you