# OPTIMUSCLOUD: Heterogeneous Configuration Optimization for Distributed Databases in the Cloud

**Ashraf Mahgoub**[1], Alexander Medoff[1] , Rakesh Kumar[2], Subrata Mitra[3], Ana Klimovic[4], Somali Chaterji[1], Saurabh Bagchi[1]

*1: Purdue University; 2: Microsoft 3: Adobe Research; 4: Google Research*

Supported by NIH R01 AI123037-01 (2016-21), WHIN center (2018-22)

# Agenda

- Introduction
- Challenges in Key-Value Stores Online Tuning
- Dynamic Workloads
- Prior work
- Proposed Approach
- Heterogeneous Configurations Benefits
- Use cases and Evaluation
- Conclusion

# Introduction

- OPTIMUSCLOUD's Goal: Achieving cost and performance efficiency for cloud-hosted distributed key-value store using online configuration tuning
- OPTIMUSCLOUD considers two set of configuration parameters:
  - Key-value store parameters:

Cache size,
# Reading\Writing threads,
Compaction method/throughput
etc.

Cloud VM parameters:

VM size/type which controls:
Number of cores
Memory Size
Network Bandwidth,
etc.

# Challenges in Online Tuning for Key-Value Stores

- Combining both sets of configuration parameters (Key-Value store + VM type/size) produces a large configuration space

cassandra

25+ Performance
Tuning Parameters

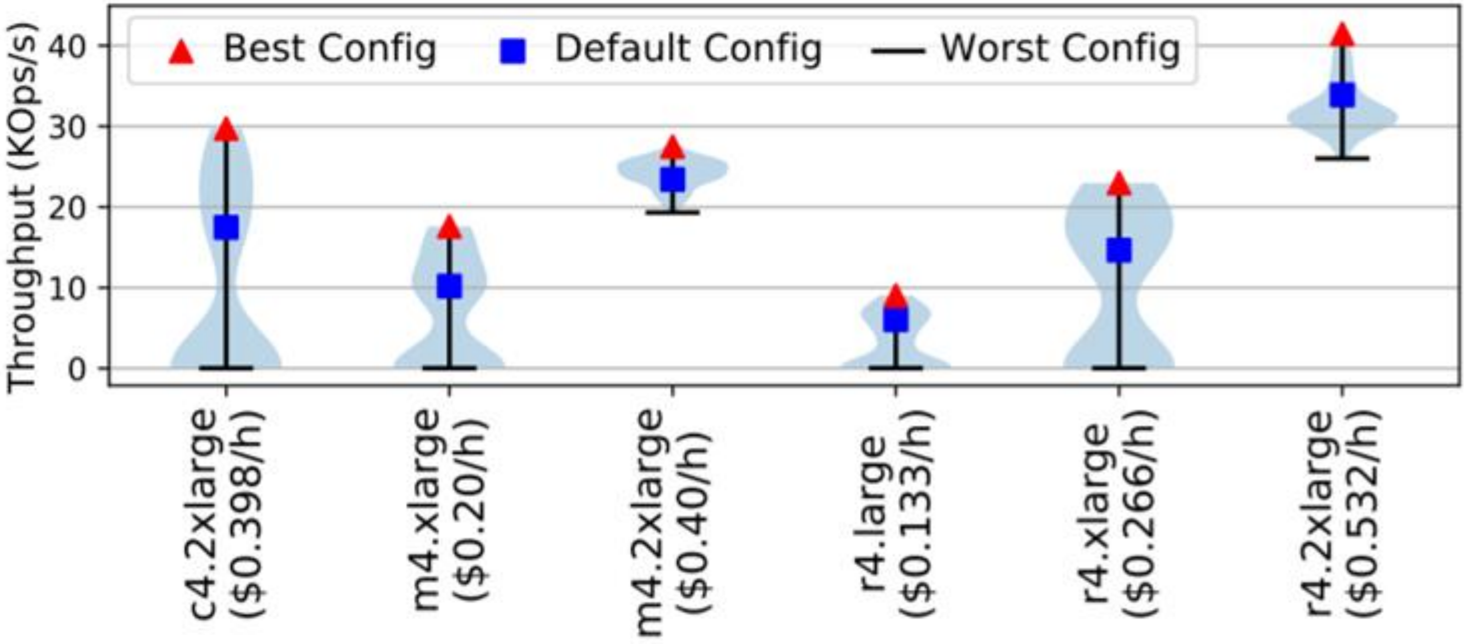
Amazon EC2

133 instance types/sizes
Prices vary by a factor of 5,000X

- Dependency between key-value store and VM configurations:
  - For example, the cache size of Cassandra is limited by the available RAM in the cloud VM

- OPTIMUSCLOUD performs *joint* optimization while taking into account the dependencies between the two spaces to achieve globally optimized performance

# Cassandra's Performance on different VM types/sizes

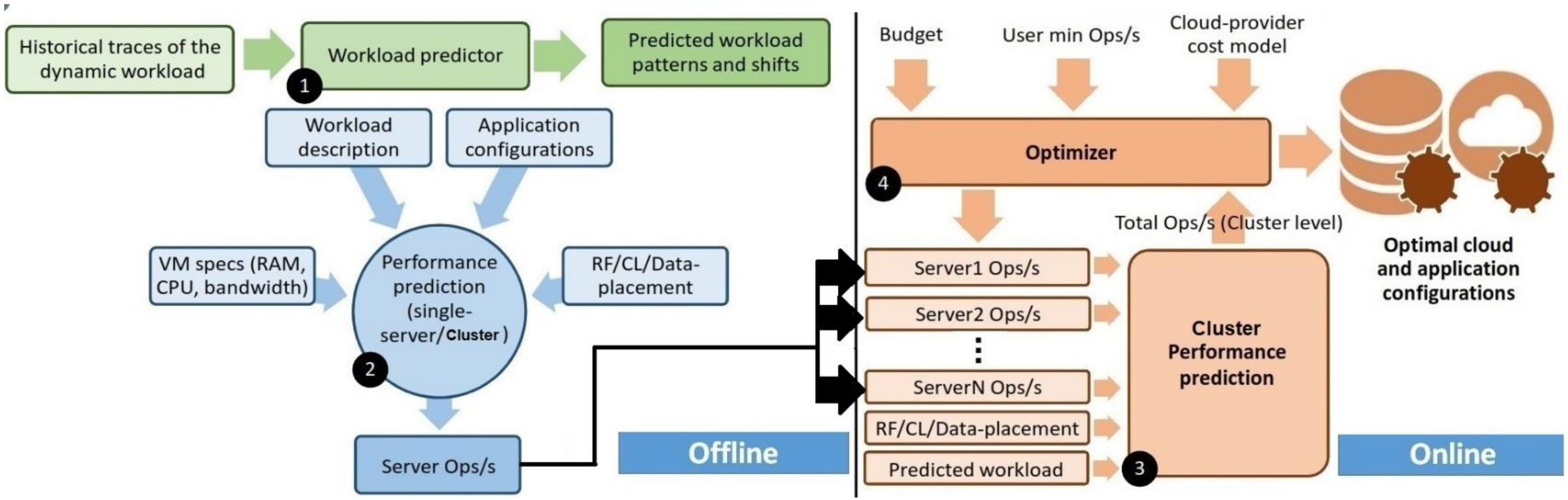



*Violin plot showing performance (throughput) of Best, Default, and Worst database configurations across different EC2 VM types.*

**Takeaways**:

- Best configurations vary across different VM types/sizes
- Therefore, jointly tuning key-value store and cloud VM parameters is crucial to achieve cost-optimal performance

# OPTIMUSCLOUD'S OVERVIEW

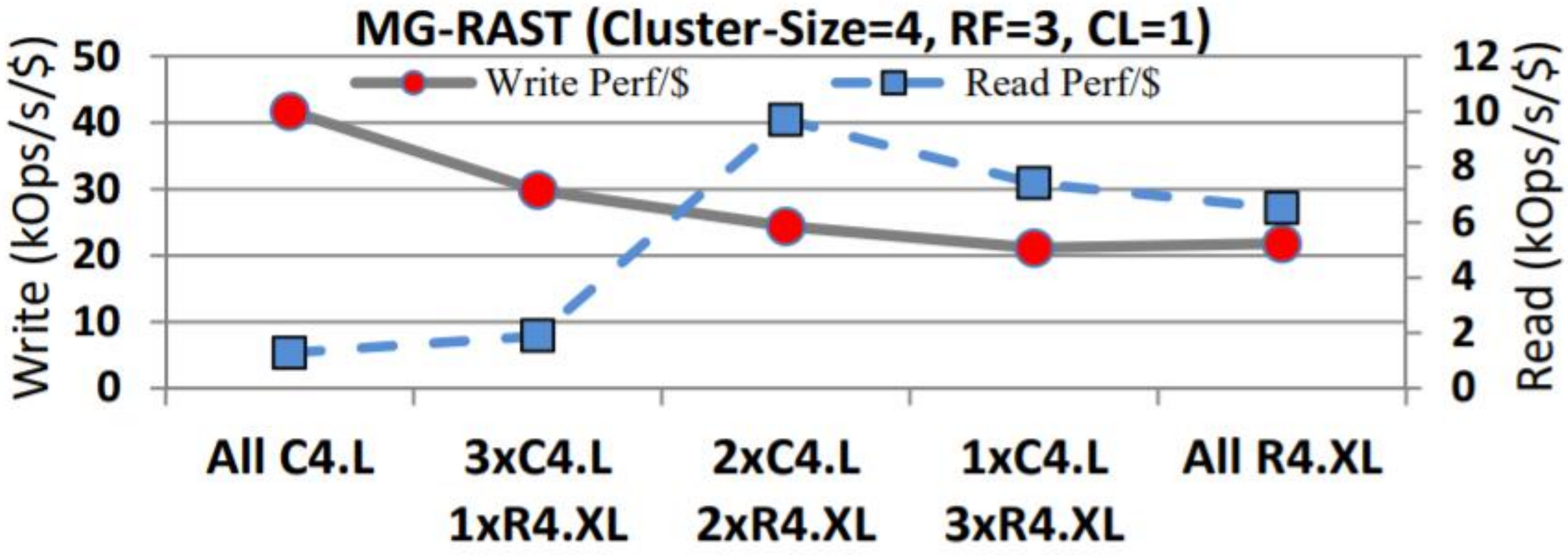


# Dynamic workloads and online reconfiguration

- Dynamic workloads:
  - Workload characteristics (e.g. Read-to-Write ratio, Request-rate, etc.) change over time, sometimes unpredictably
  - New characteristics causes current configurations to perform sub-optimally, necessitating reconfigurations
- Impact of online reconfiguration :
  - Changing configurations at runtime usually requires a server-restart, causing a downtime and a degradation in performance
  - For fast changing workloads, frequent reconfiguration of the overall cluster could severely degrade performance

- Q: Can we reconfigure only a subset of the nodes in the cluster? Which subset?
  - This will lead to *heterogenous* configuration

# Why heterogeneous configurations is beneficial?




*Change in Perf/$ for the write (solid) and read throughput (dotted) as we reconfigure the nodes from C4.large to R4.xlarge.*

Best Configurations To optimize Perf/$:
Write-Heavy -> All C4.L
Read-Heavy -> 2 C4.L & 2 R4.XL



PURDUE UNIVERSITY

# OptimusCloud's Solution

- Heterogeneous configurations: Reduce reconfiguration downtime & avoids overprovisioning
- However, heterogeneity increases the configuration space size
  - Consider a cluster of N=20 nodes and I=15 configurations
  - Homogeneous: We have I=15 possible configurations
  - Heterogeneous: We have $\binom{N+I-1}{I-1} = 1.3 \times 10^9$ possible configurations

- OptimusCloud uses the concept of **Complete-Sets** to reduce the size of the search space
  - **Complete-Set:** the minimum subset of nodes for which the union of their data records covers all the records in the database at least once

# Complete-Sets

- This concept of Complete-Set relies on selecting the fastest replica for a given request
  - Dynamic Snitch (Cassandra) or Adaptive Replica Selection (Elasticsearch)
- Consistency-Level (CL) defines how many replicas need to reply to a request before it is satisfied
  - Therefore, the slow replica will dominate the response latency
  - The servers within a Complete-Set must be upgraded to the faster configuration upon a workload change for the cluster performance to improve
- OPTIMUSCLOUD keeps the configurations homogeneous within the same Complete-Set, while allowing different Complete-Sets to have different configurations

# How partitioning the cluster into Complete-Sets reduces the search space?

- First, we show that we have at most **#Complete-Sets = Replication-Factor** for any cluster (proof is given in the paper)
  - RF is practically low (3 or 5)

- Second, reconfiguring **#Complete-Sets = Consistency-Level (CL<=RF)**, all requests are served from nodes with optimized configurations

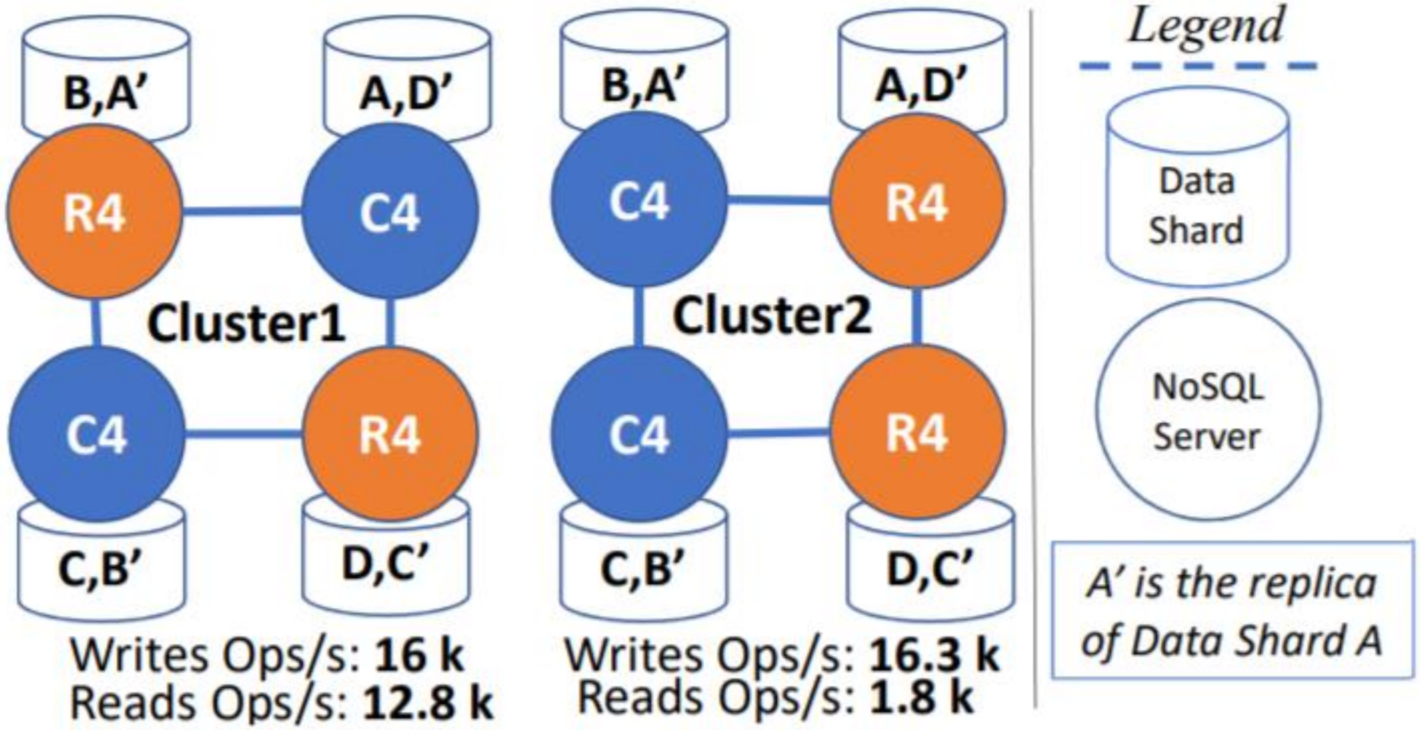
- With S Complete-Sets, the size space is reduces to $\binom{S+I-1}{I-1} = 680$ possible configurations for a cluster with RF=3

(Compared to $1.3\times10^{9}$)



PURDUE UNIVERSITY

# Using data-placement info to identify Complete-Sets




***Cluster1 achieves 7× read Ops/s over Cluster2***

# Applications

1. MG-RAST:
   - Real workload traces from the largest metagenomics analysis portal
   - Its workload does not have any discernible daily or weekly pattern, as the requests come from all across the globe
   - Workload can change drastically over a few minutes (accurately predictable for 5min)
2. Bus-Tracking:
   - Real workload traces from a bus-tracking mobile application
   - Traces show a daily pattern of workload switches.
   - Workload is accurately predictable for longer look-ahead periods (e.g. 2 hours)
3. HPC:
   - Simulated workload traces from data analytics jobs submitted to a shared HPC queue.
   - Using profiling techniques, job execution times can be predicted with high accuracy and for long look-ahead periods.

# Performance Prediction Accuracy

| Workload | MG-RAST | | BUS | | HPC | |
|---|---|---|---|---|---|---|
| Metric | $R^2$ | $RMSE$ | $R^2$ | $RMSE$ | $R^2$ | $RMSE$ |
| N-Solitary -Models | 0.2 | 3401.4 | 0.127 | 109.9 | 0.04 | 2778 |
| Selecta [ATC-18] | -0.14 | 4149.3 | 0.66 | 110.6 | 0.932 | 2451 |
| OPTIMUS -Categorical | 0.41 | 1334.2 | 0.986 | 21.87 | 0.983 | 1172.9 |
| OPTIMUS -Numerical | **0.89** | **1260.9** | **0.988** | **19.77** | **0.986** | **1076.2** |

***Comparison of different performance prediction techniques.*** OPTIMUS-CLOUD ***achieves better performance in terms of*** $R^2$ ***and*** *RMSE* ***over all baselines.***

# Baselines

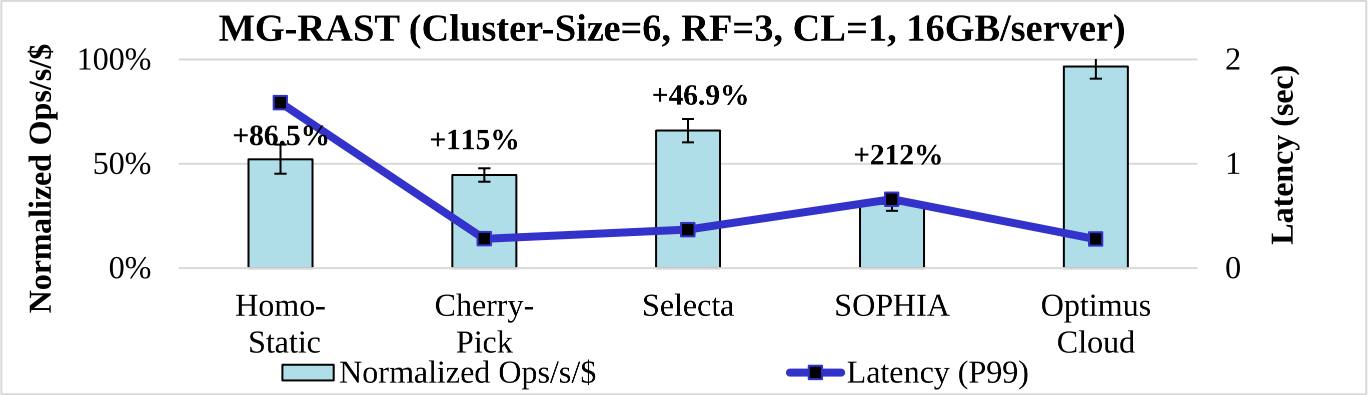
1. Homogeneous-Static: the single best configuration to use for the entire duration of the predicted workload. Impractical because assumes perfect knowledge of future workload

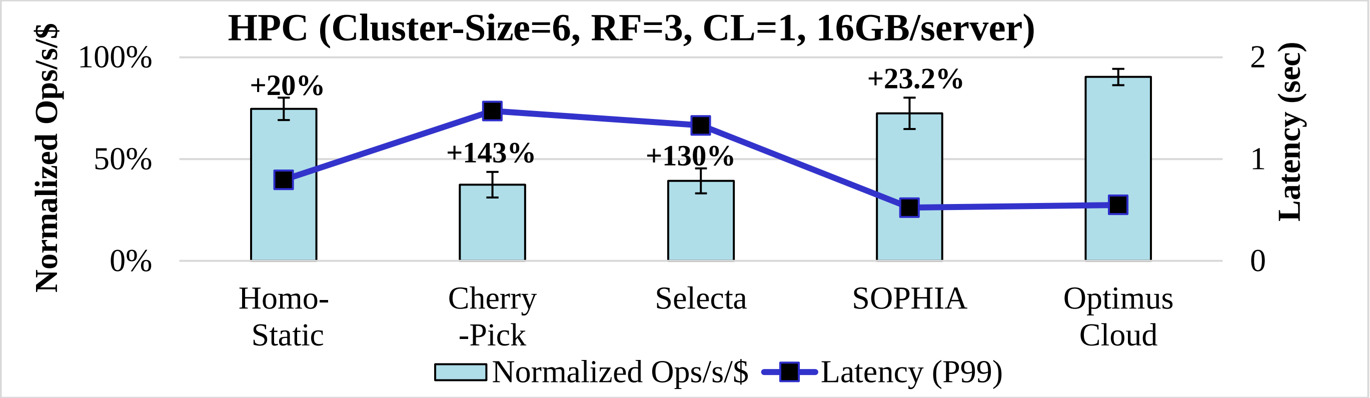
2. CherryPick [*NSDI-17]:* Uses Bayesian Optimization to find a heterogeneous cloud configuration for a representative job/phase of the workload

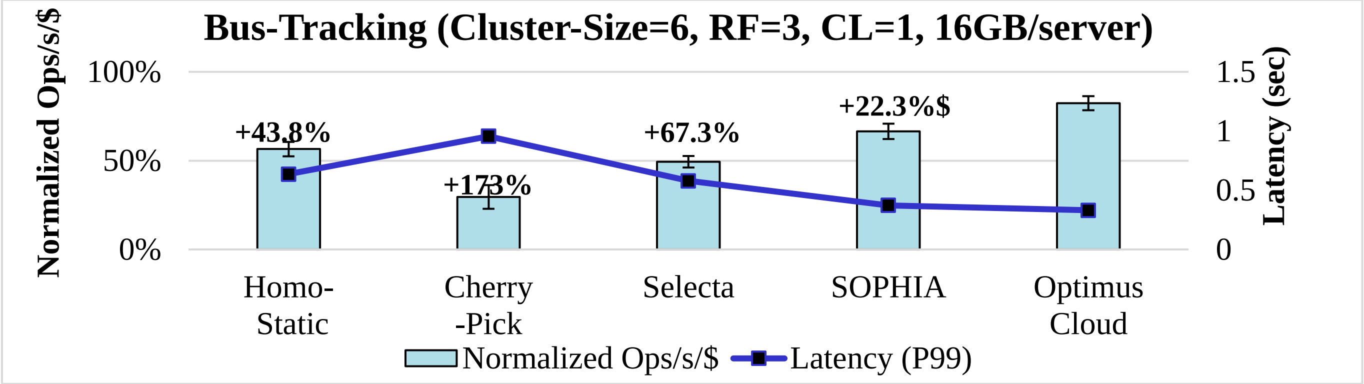
3. Selecta [*ATC-18]*: uses SVD techniques to select the optimized homogeneous cloud configuration for different jobs/phases of the workload

4. SOPHIA [*ATC-19]*: uses Genetic-Algorithms and performance modeling to find optimized homogeneous configurations for Key-Value store parameters

# Evaluation: Cassandra

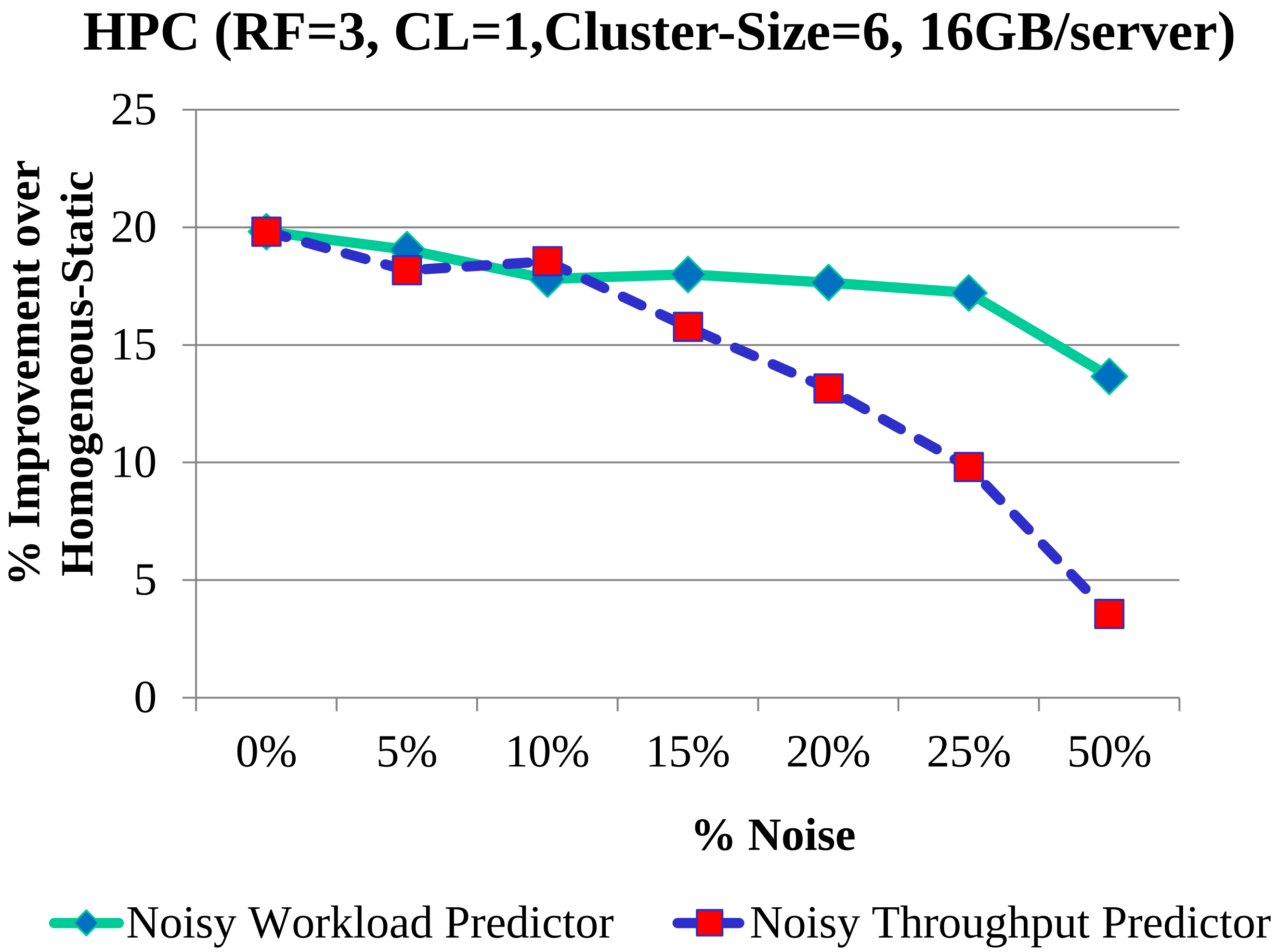



Compared to SOPHIA, OPTIMUSCLOUD achieves up to 212% better Perf/$ as Sophia considers only homogeneous configurations for key-value store parameters without considering online reconfiguration for the cloud VM type/size.

# Tolerance to Prediction Errors




OPTIMUSCLOUD's improvement over Homogeneous-Static decreases with increasing levels of noise, as the selected configurations deviate from the best configurations.

OPTIMUSCLOUD's is more sensitive to errors in the throughput predictor compared to errors in the workload predictor, which is demonstrated in the steeper downward slope in the noisy throughput predictor curve.

# Conclusion

- For cost-optimal performance of a distributed Key-Value store in the cloud, it is critical to ***jointly*** tune Key-Value store and cloud configurations.

- OPTIMUSCLOUD provides the insight that it is optimal to create ***heterogeneous*** configurations and for this, it determines at runtime the *minimum number of servers* to reconfigure.

- Using a novel concept of **Complete-Sets**, OPTIMUSCLOUD provides a technique to reduce the large search space that is brought out by heterogeneity

- Configurations found by OPTIMUSCLOUD outperform those by prior works, CherryPick, Selecta, and SOPHIA, in both **Perf/$** and **Tail Latency (P99)**


Thank you