# A Flexible System Integrating Monitoring and Analytics for Managing Large-Scale Data Centers

Chengwei Wang, Karsten Schwan, Vanish Talwar*, Greg Eisenhauer
Liting Hu, Matthew Wolf

CERCS Georgia Tech *HP Labs Palo Alto


Carnegie Mellon University Georgia Tech intel PRINCETON UNIVERSITY UC Berkeley

# Motivations

- More, In-Depth Analysis
- Online Problem Diagnosis/Agile Actuations
- Scalable in Cloud

# Research Challenges

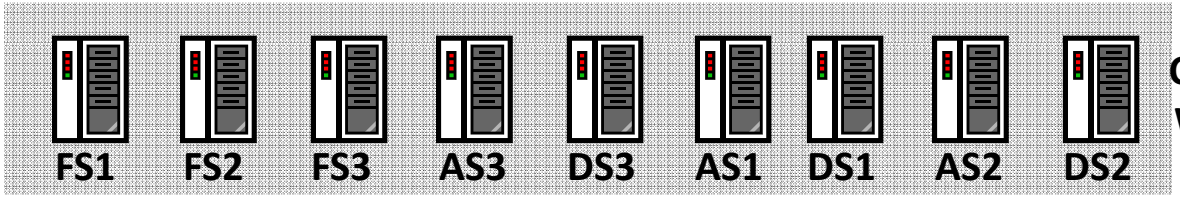
- Bridging Monitoring and Analytics
- Meeting the Varying Requirements in Time and Space
- Cost-Effectiveness Consideration
- Scalability

# Solution: Monalytics

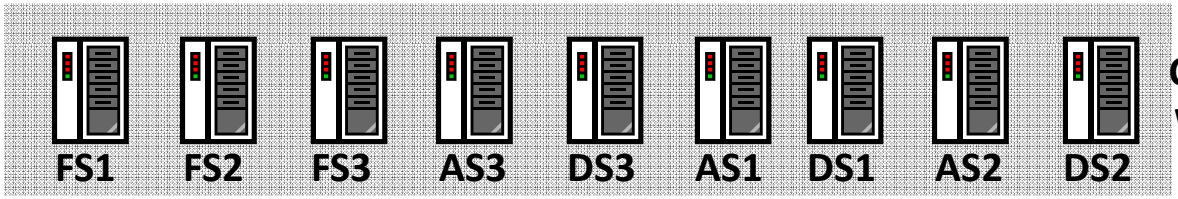
- Monalytics = Monitoring + Analytics
- In-situ Data Collection and Processing
- Design and Use Case Evaluation
- Analytical Model and Evaluation

# Use Case I: Zoom-In Analysis

# Use Case I: Zoom-In Analysis

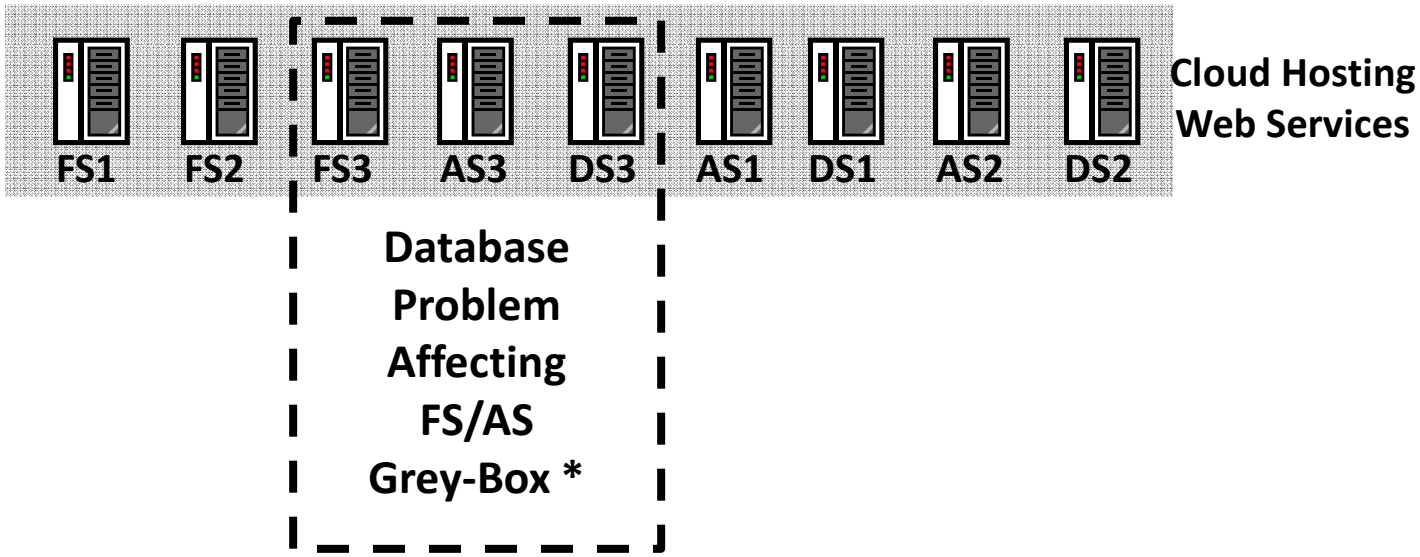


# Use Case I: Zoom-In Analysis

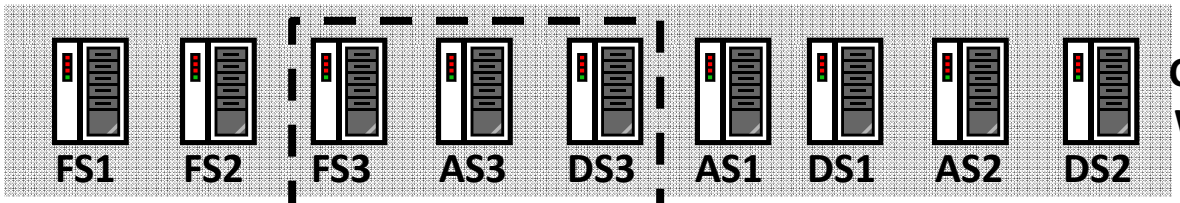



**FS: Front-End Server** **AS: Application Server** **DS: Database Server**

# Use Case I: Zoom-In Analysis



# Use Case I: Zoom-In Analysis

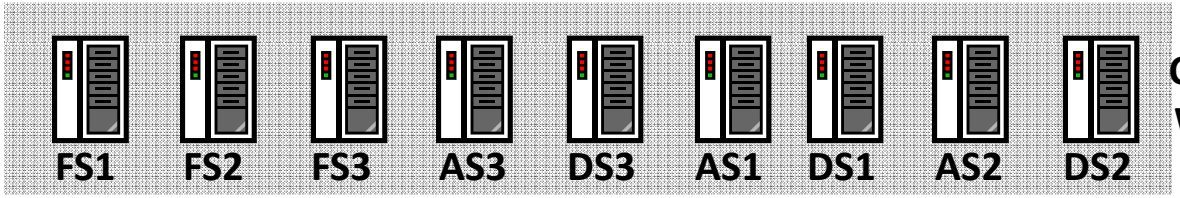



**Database Problem Affecting FS/AS Grey-Box ***

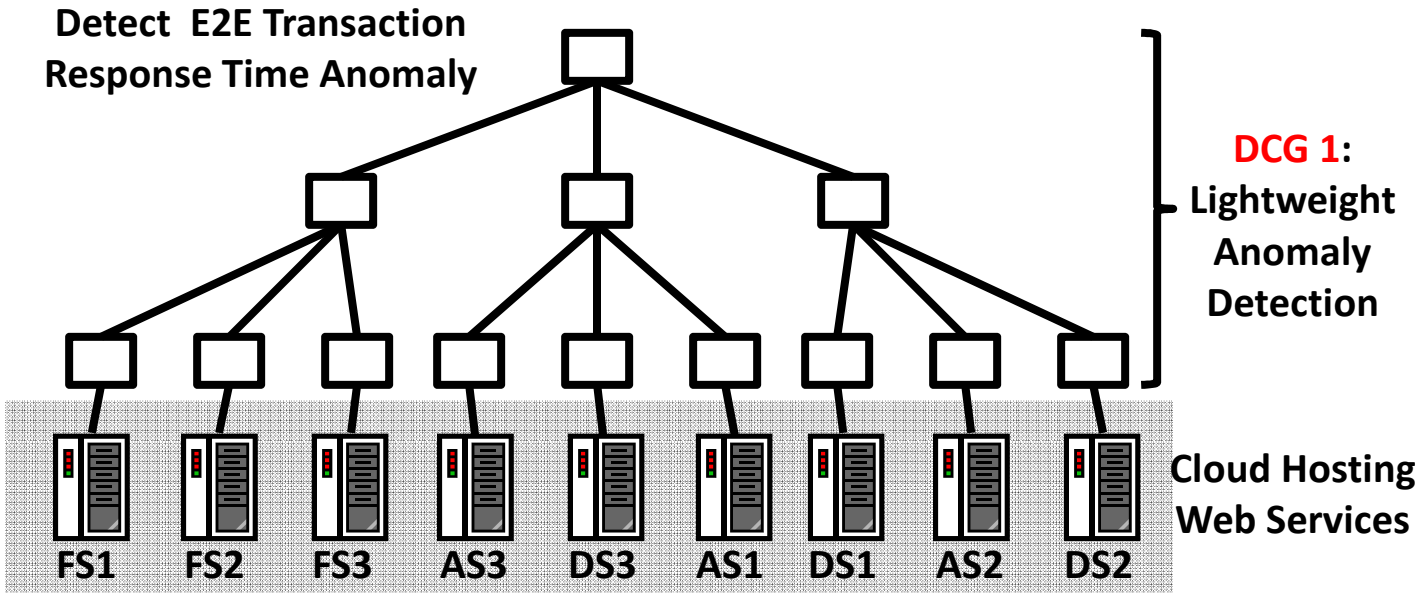
*Cloud operators may know FS3, AS3 and DS3 VMs belong to one cloud user but we don't know what's the relationship between them

# Use Case I: Zoom-In Analysis

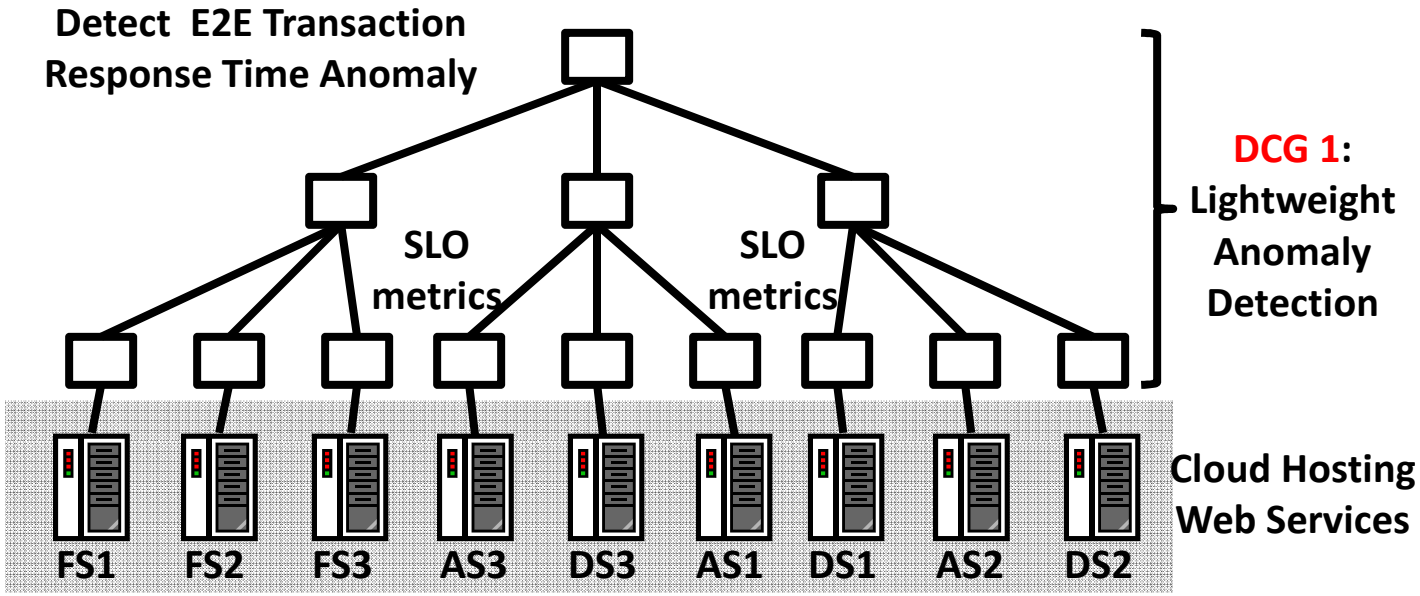
Utility Cloud can host thousands of web applications each of which has hundreds to thousands of components. How to quickly detect and find the root causes of the problem?



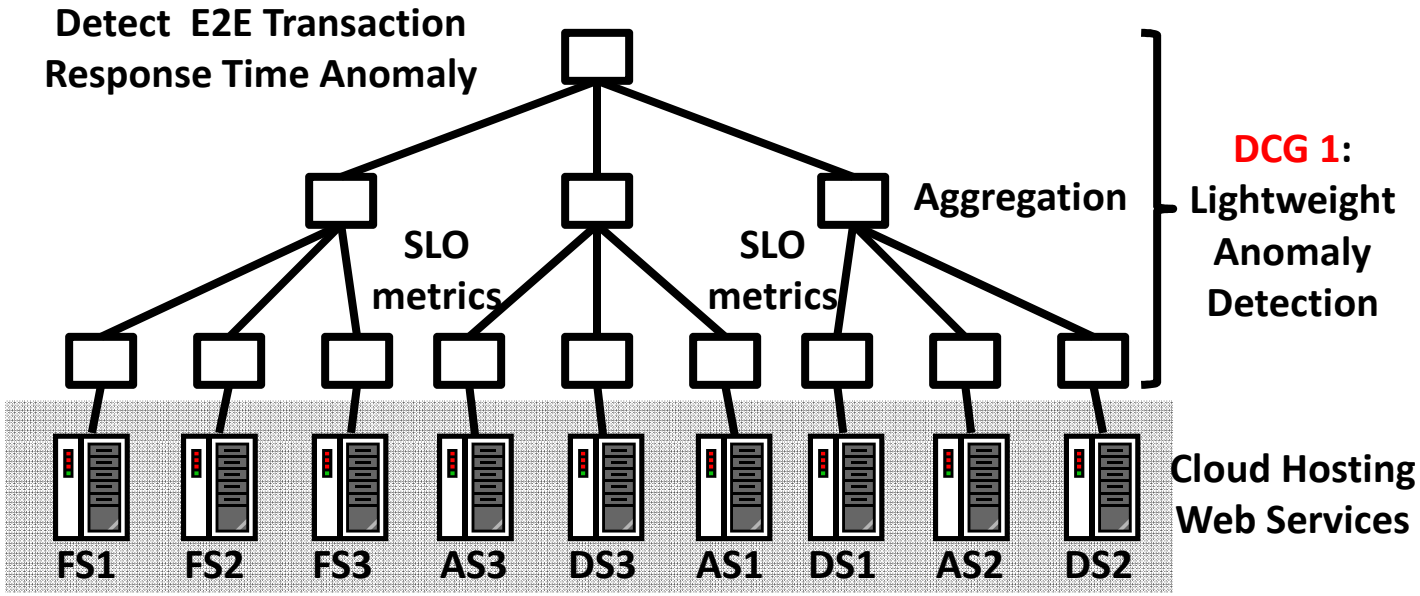
# Use Case I: Zoom-In Analysis



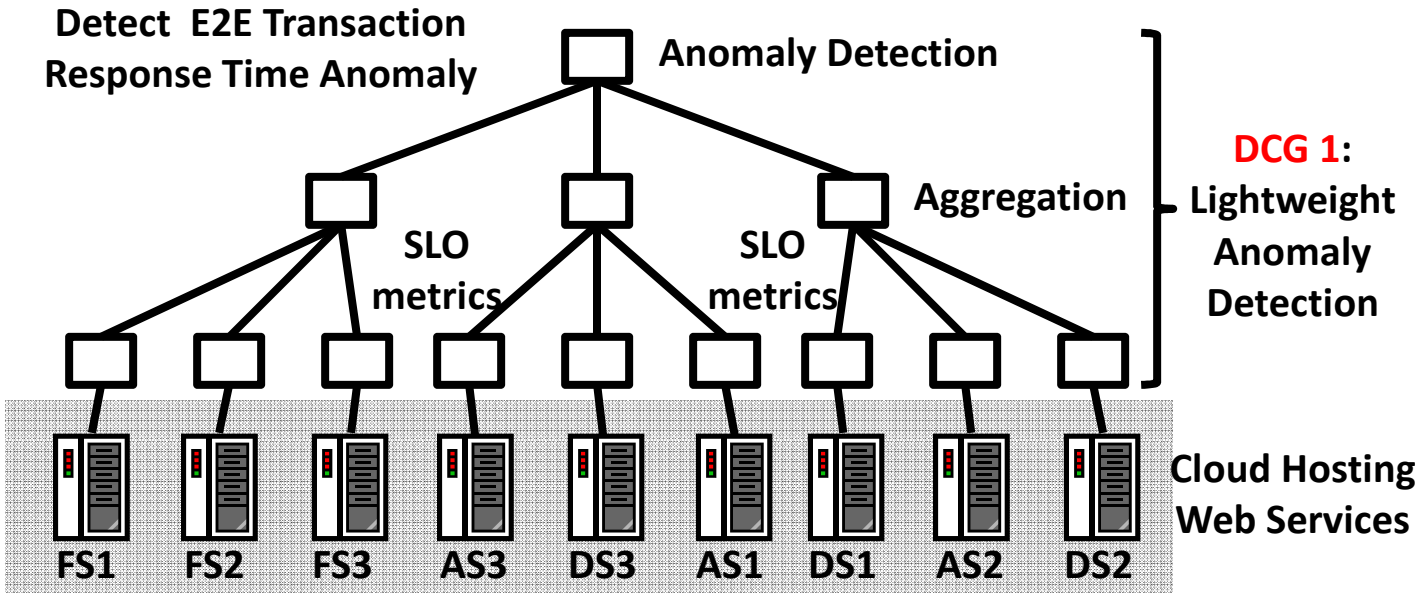
# Use Case I: Zoom-In Analysis



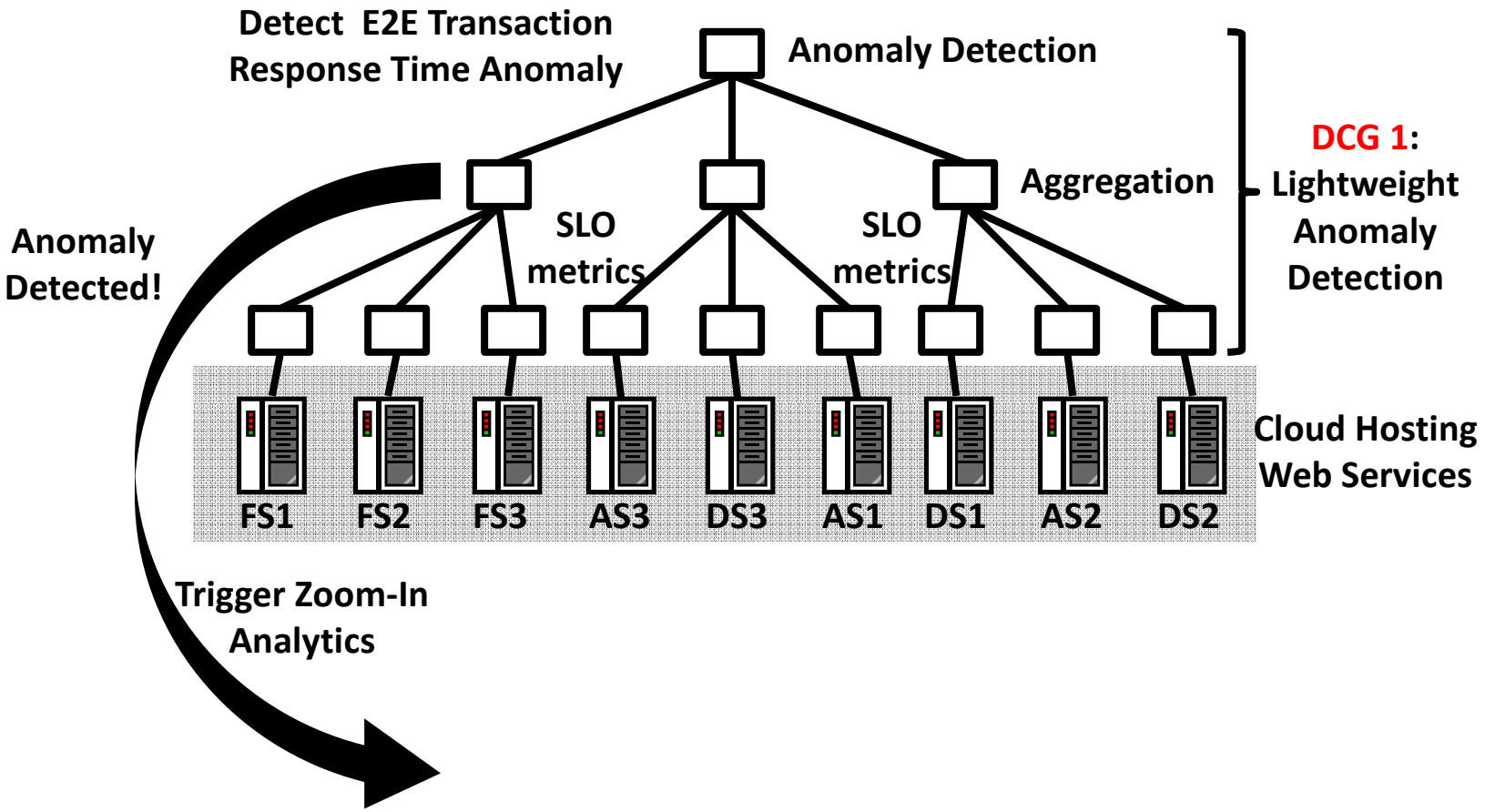
# Use Case I: Zoom-In Analysis



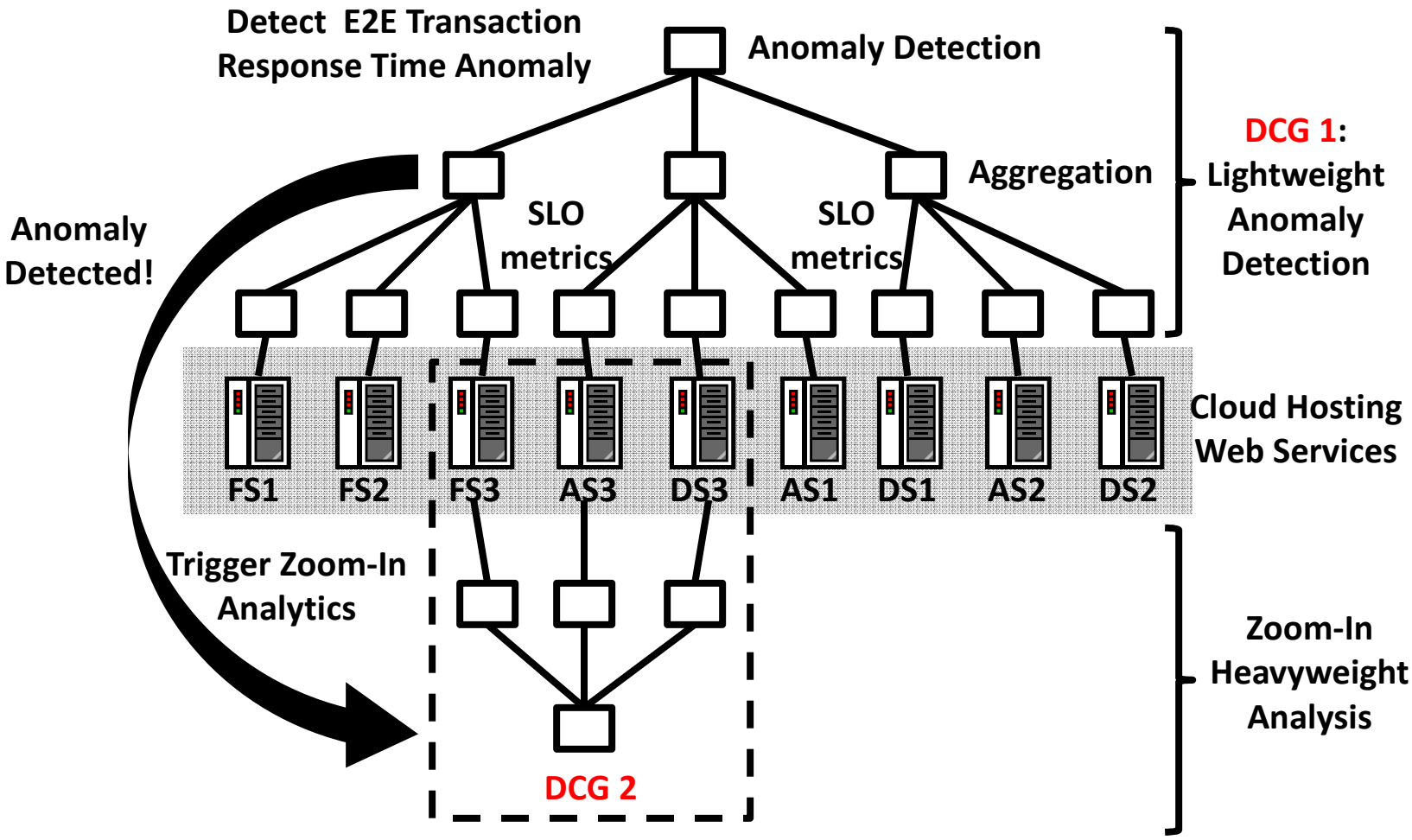
# Use Case I: Zoom-In Analysis



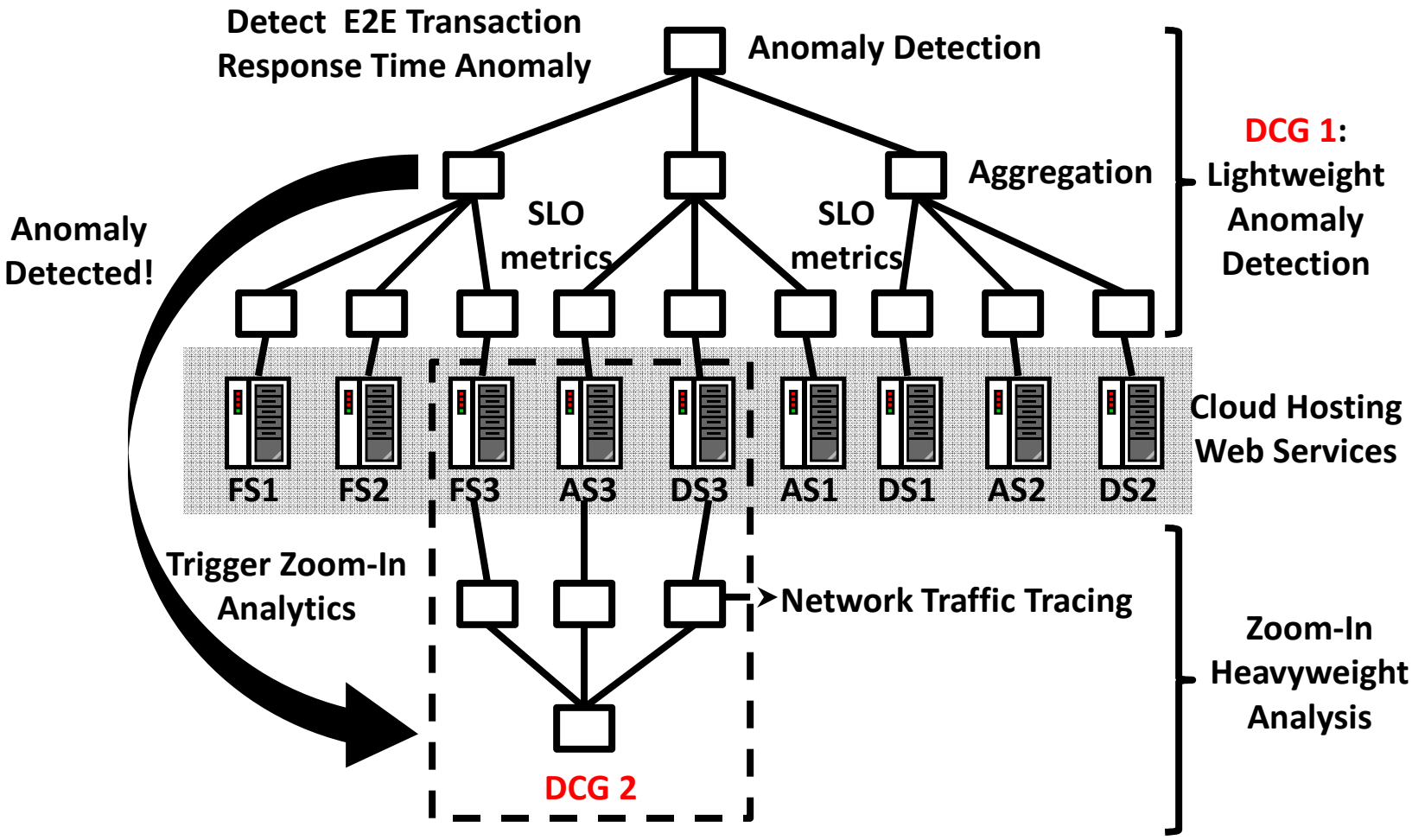
# Use Case I: Zoom-In Analysis



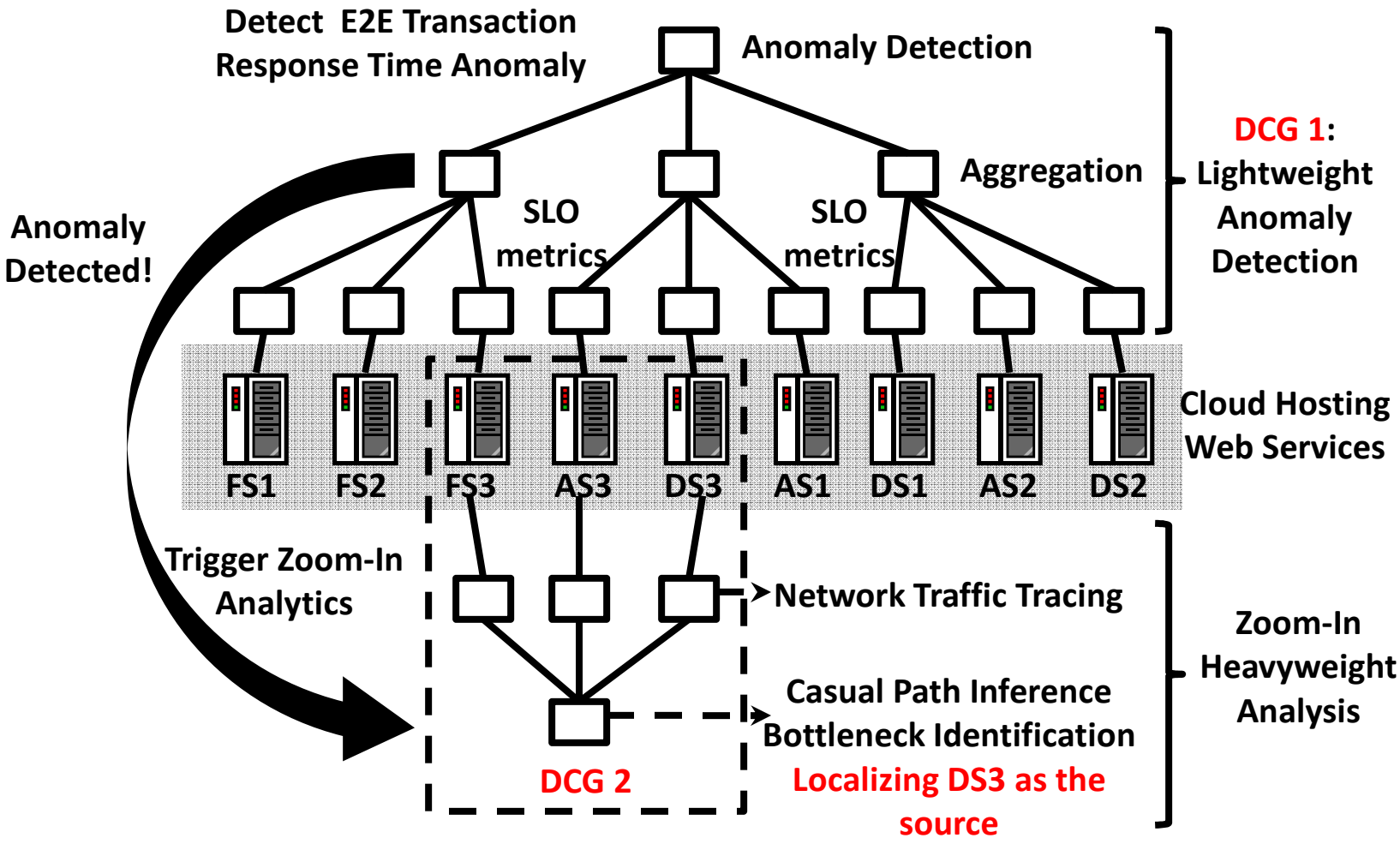
# Use Case I: Zoom-In Analysis



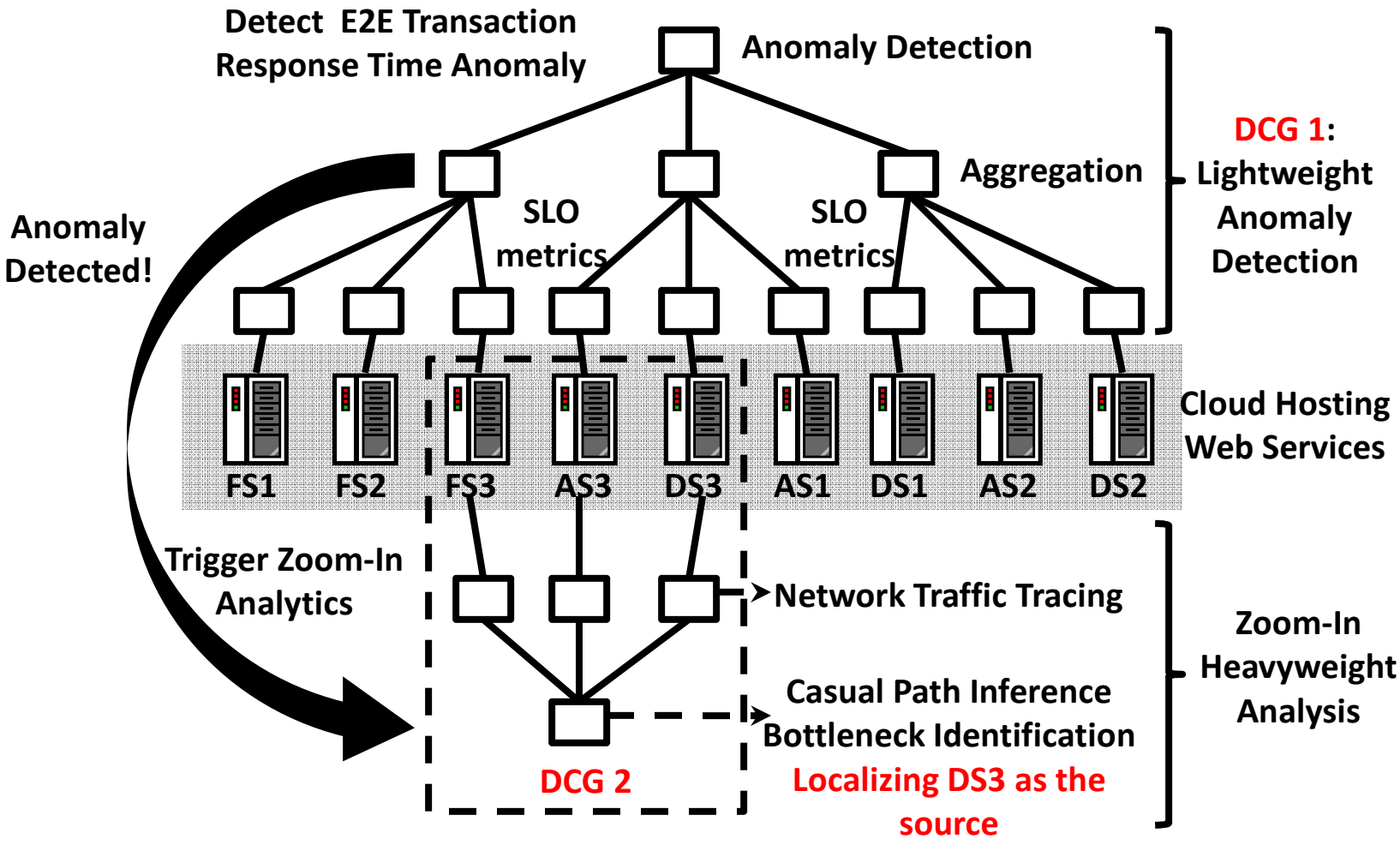
# Use Case I: Zoom-In Analysis



# Use Case I: Zoom-In Analysis



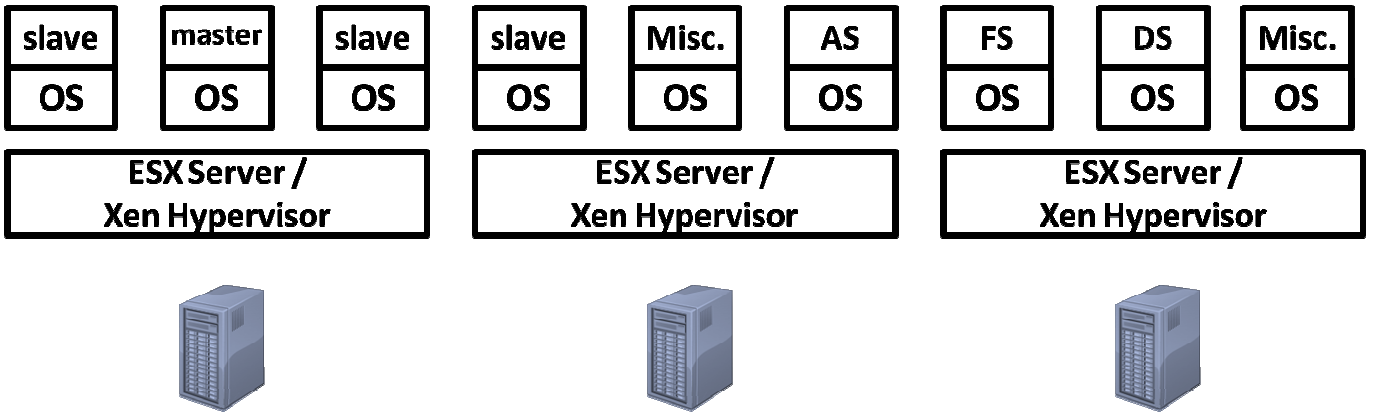
# Use Case I: Zoom-In Analysis



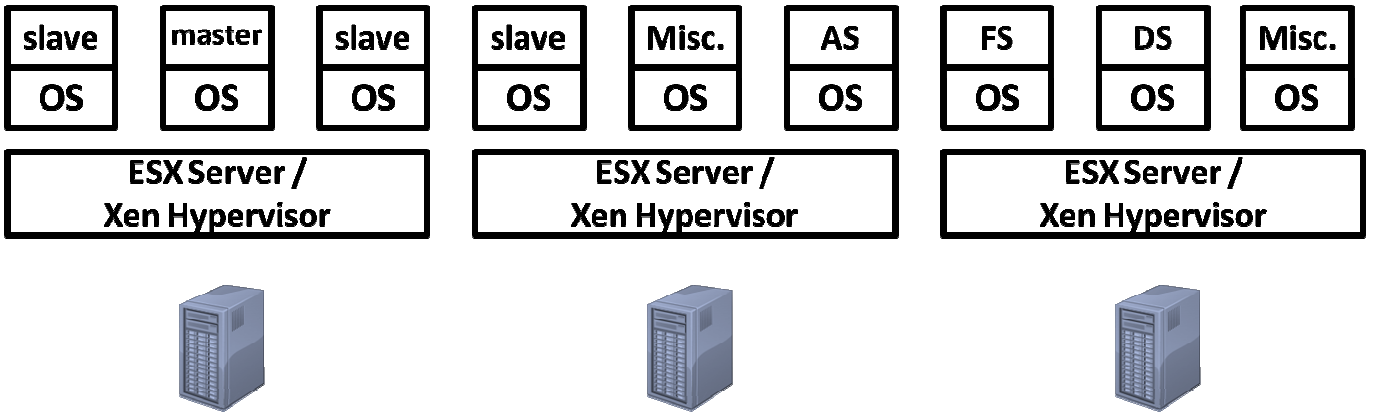

**Highly reduced data overhead and focus on problematic area**

# Use Case II: Virtual Machine Clustering

# Use Case II: Virtual Machine Clustering

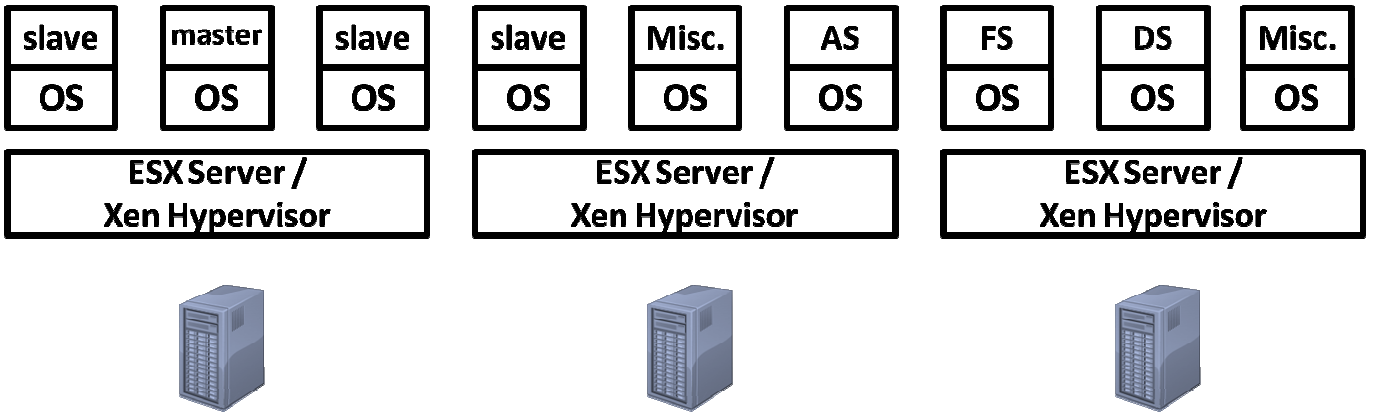


# Use Case II: Virtual Machine Clustering




**Master : MapReduce Master  Slave:  MapReduce Worker  Misc. : Other**

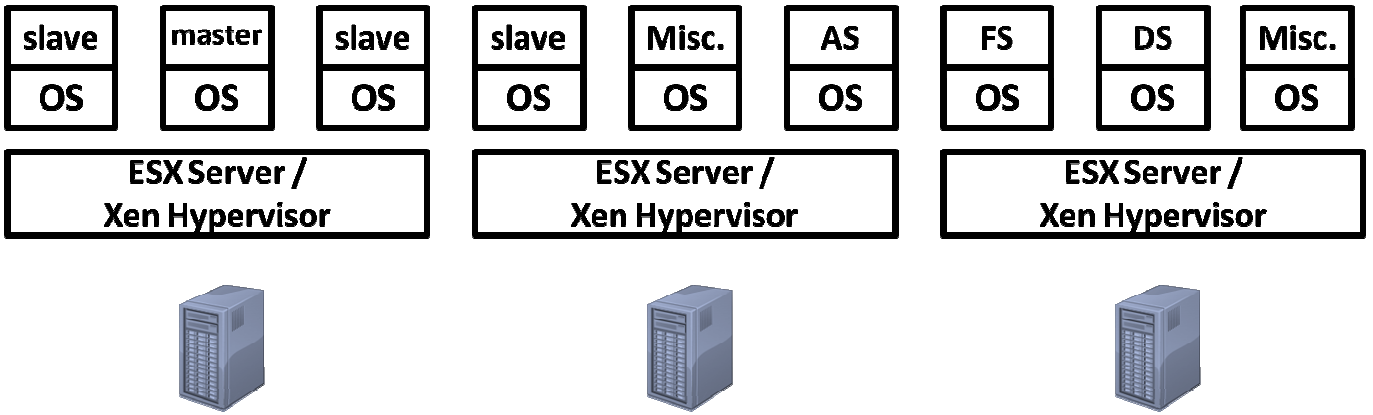
# Use Case II: Virtual Machine Clustering



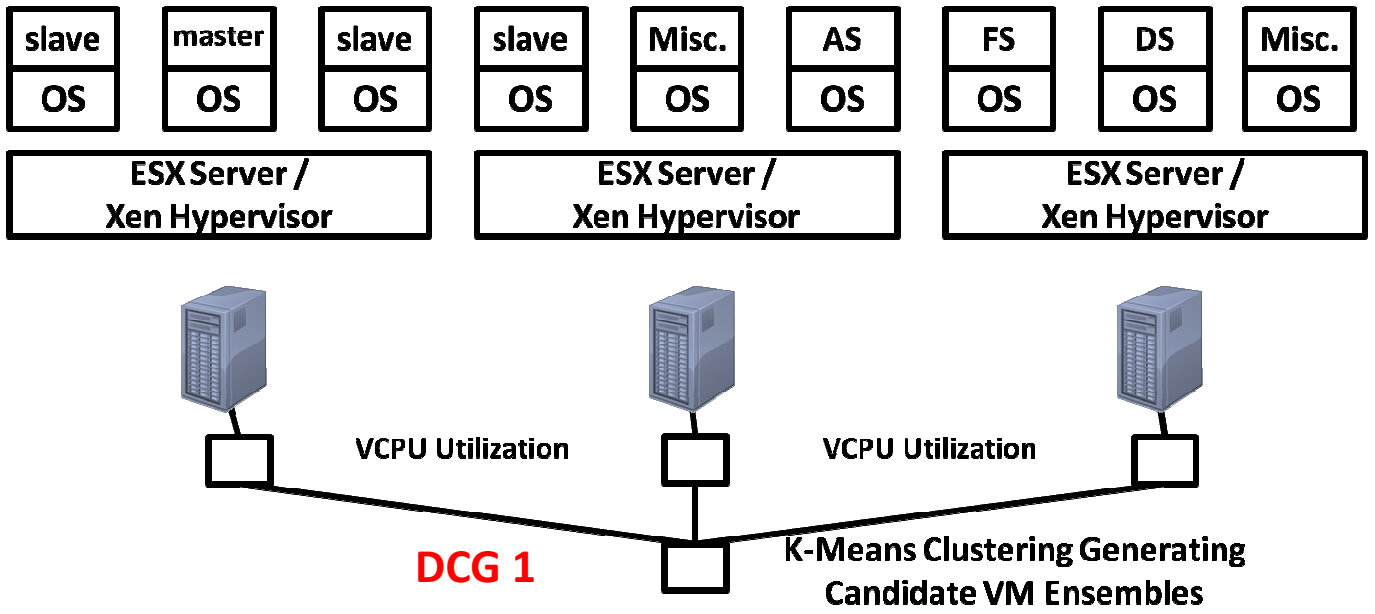

**Black-Box:** cloud operators don't know which VMs are logically grouped

# Use Case II: Virtual Machine Clustering

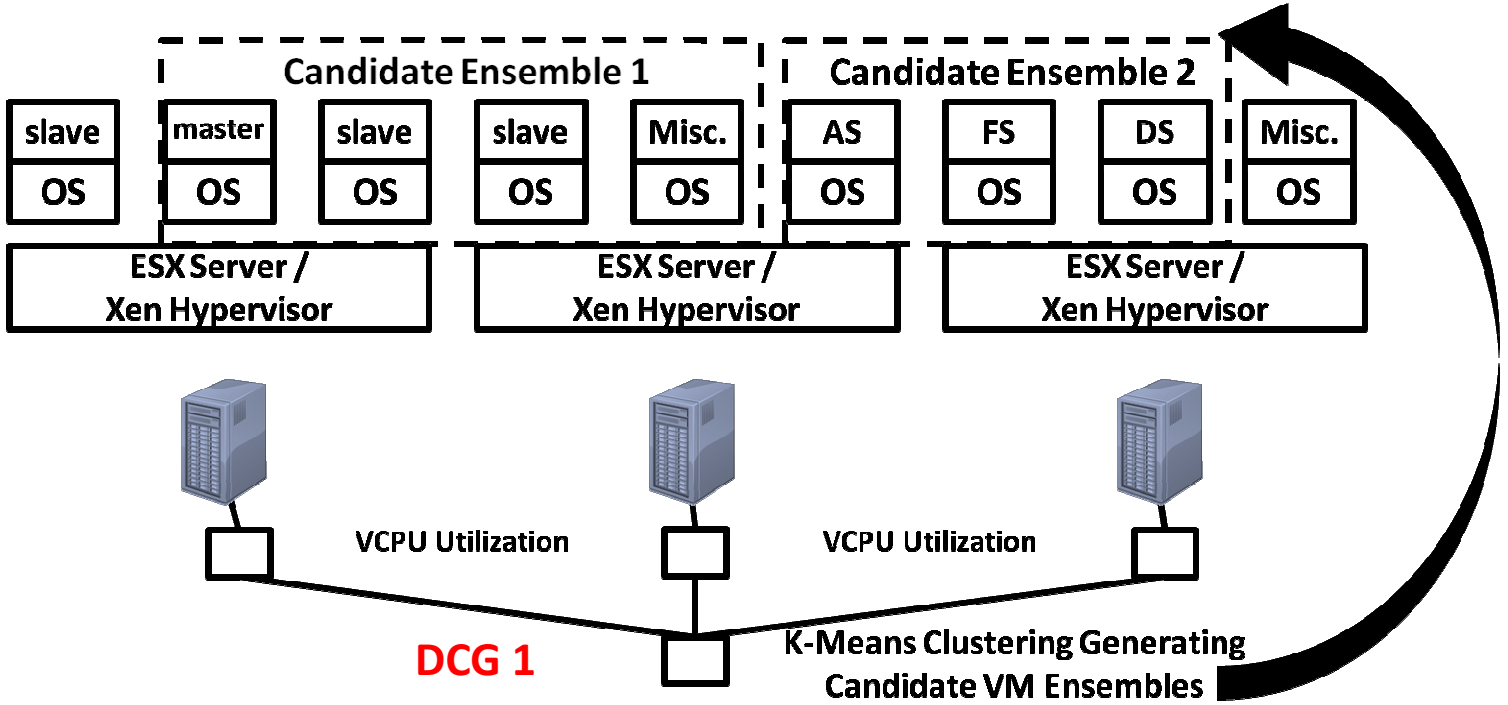
**How to dynamically `cluster' VMs for performance optimization?**
**e.g. Migrate cross-rack MapReduce VMs to the same rack**



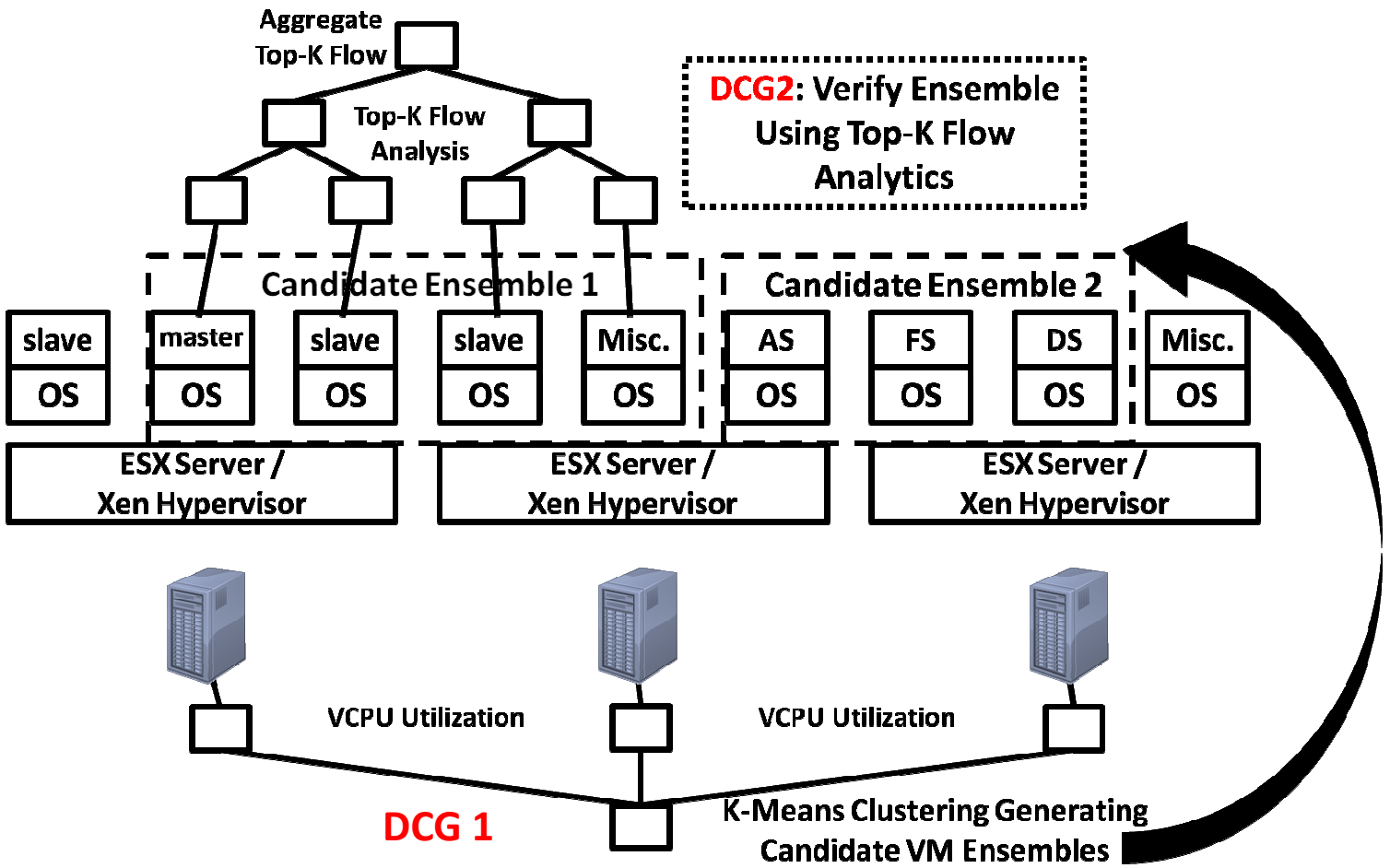
# Use Case II: Virtual Machine Clustering



# Use Case II: Virtual Machine Clustering

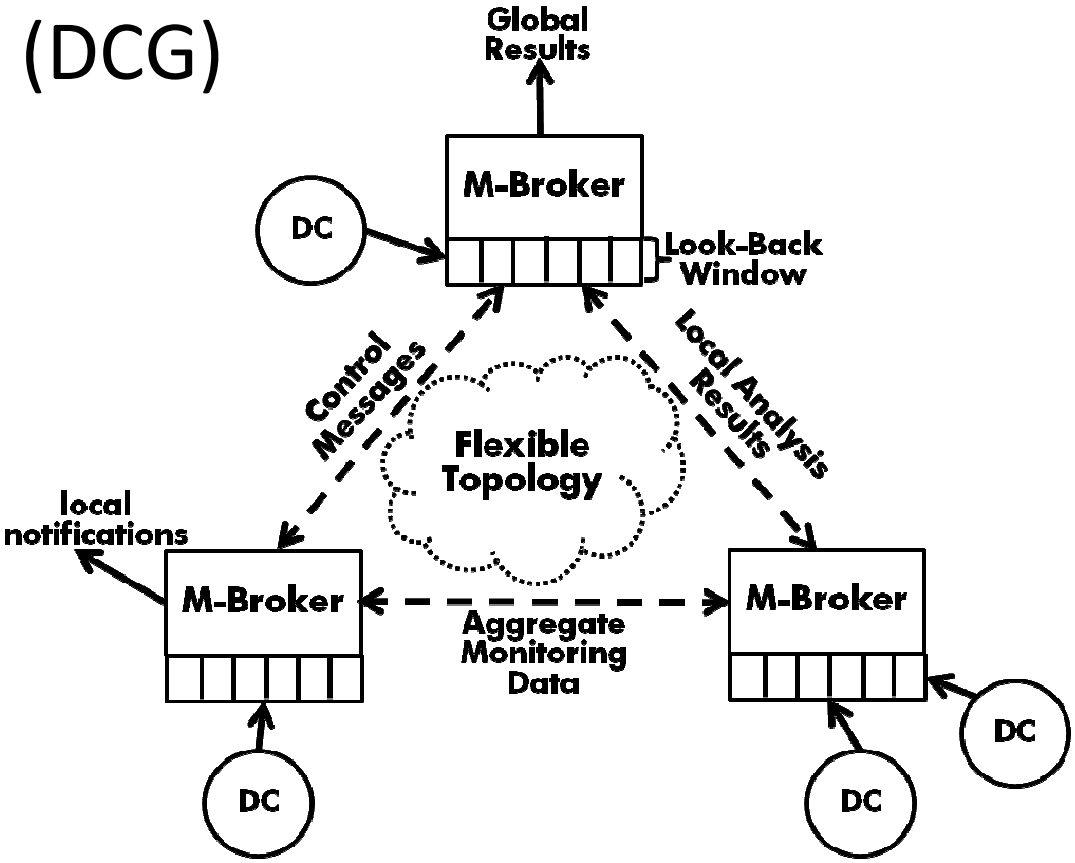


# Use Case II: Virtual Machine Clustering



# Monalytics Architecture

- Building Blocks: Distributed Computation Graph (DCG)

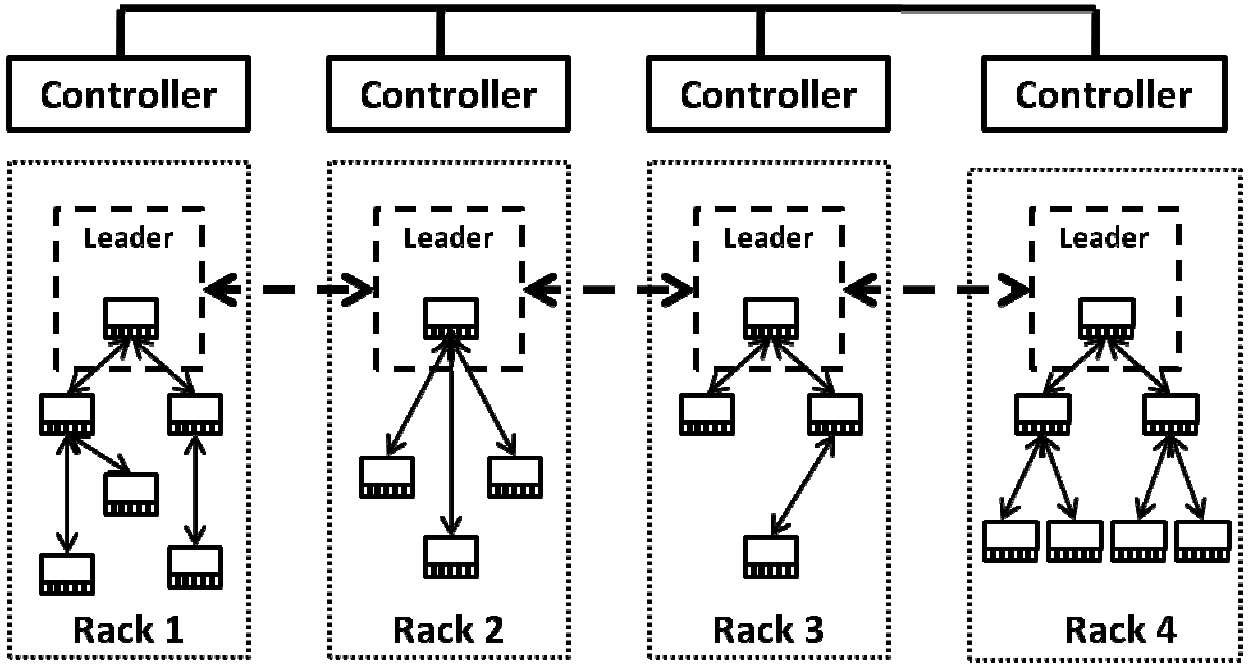


# Flexible Topologies of DCG

- Deployment On-Demand
- Traditional Topologies
  - > Centralized
  - > Hierarchical Tree
  - > Binomial Forests
- Hybrid Topologies
- Dedicated/Collocated Mode

# Monalytics Architecture

- Monalytics Assembly in Data Centers



# Parameters

| Parameter | Notation | Example value |
|---|---|---|
| Number of Nodes | $N$ | $100 - 10^6$[17] |
| Network Latency | $l$ | 0.25ms[17] |
| Bisection Bandwidth | $B$ | 1 Gbps[7] |
| Bandwidth Budget (in %) | $b$ | 0.1% - 1% |
| Size of Monitoring Data per Node | $s$ | 100KBytes |
| Number of Metrics | $n$ | 1890 |
| Processing Time per Metric | $a$ | $3.5 * 10^{-8}$seconds |
| Capital Cost per Server | $c$ | $1000[3] |
| Fraction of Mgmt Cost Per Server | $\alpha$ | 1/16[3] |
| Region Size | $N_r$ | 1000 |

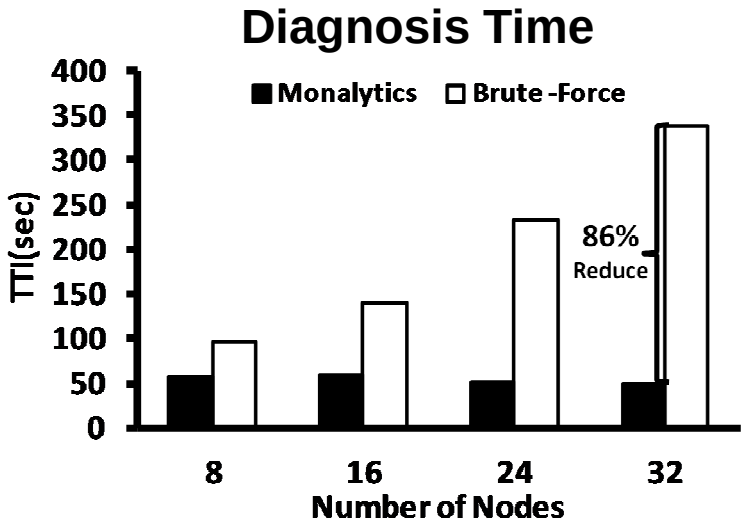
# Analytical Models

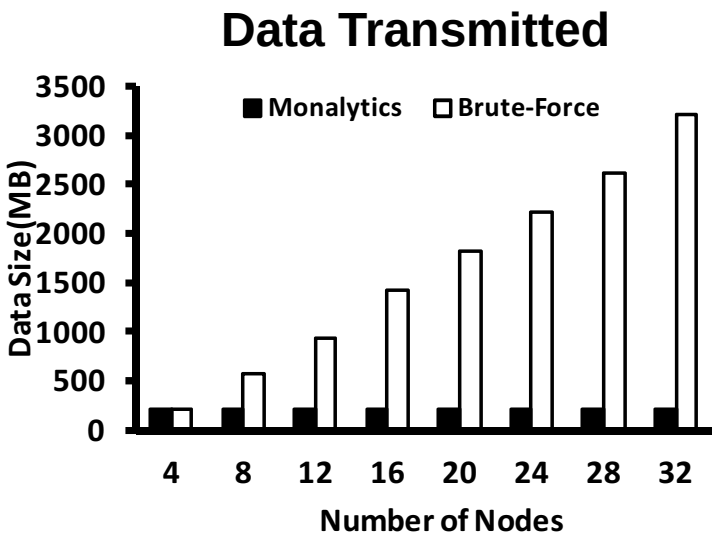
| | Time To Insight (TTI) | Capital Costs |
|---|---|---|
| **Centralized** | $T_C(N) = \frac{s}{B*b} + l + F(n,N), if B > B*b*N$<br>$or = \frac{s*N}{B} + l + F(n,N), if B \leq B*b*N$ | $C_C(N) = \lceil \frac{N}{1000} \rceil * c + N*c*\alpha$ |
| **Hierarchical Tree** | $T_{HT}(N) = (\lceil \log_d N \rceil - 1) * (\frac{d*s}{B} + l + F(n,d)) + \frac{\lceil \frac{N}{d^{\lceil \log_d N \rceil - 1}} \rceil * s}{B} + l + F(n, \lceil \frac{N}{d^{\lceil \log_d N \rceil - 1}} \rceil)$<br>$\bar{T}_{HT}(N) = (\lceil \log_d N \rceil - 1) * (\frac{d*s}{B*b} + l + F(n,d)) + \frac{\lceil \frac{N}{d^{\lceil \log_d N \rceil - 1}} \rceil * s}{B*b} + l + F(n, \lceil \frac{N}{d^{\lceil \log_d N \rceil - 1}} \rceil)$ | $C_{HT}(N) = \sum_{i=1}^{\lceil \log_d N \rceil} \lceil \frac{N}{d^i} \rceil * c + N*c*\alpha$<br>$\bar{C}_{HT}(N) = N*c*\alpha$ |
| **Binomial Forest** | $T_{BSF}(N) = \lceil \log_2 N \rceil * (\frac{s}{B*b} + l + F(n,2))$ | $C_{BSF}(N) = N*c*\alpha$ |
| **Hybrid Topologies** | $T_{HYBRID}(N) = \max_{i=1}^{m} T_{INTRA}(R_i) + T_{INTER}(m)$ | $C_{HYBRID}(N) = \sum_{i=1}^{m} C_{INTRA}(R_i) + C_{INTER}(m)$<br>$(N = \sum_{i=1}^{m} R_i)$ |

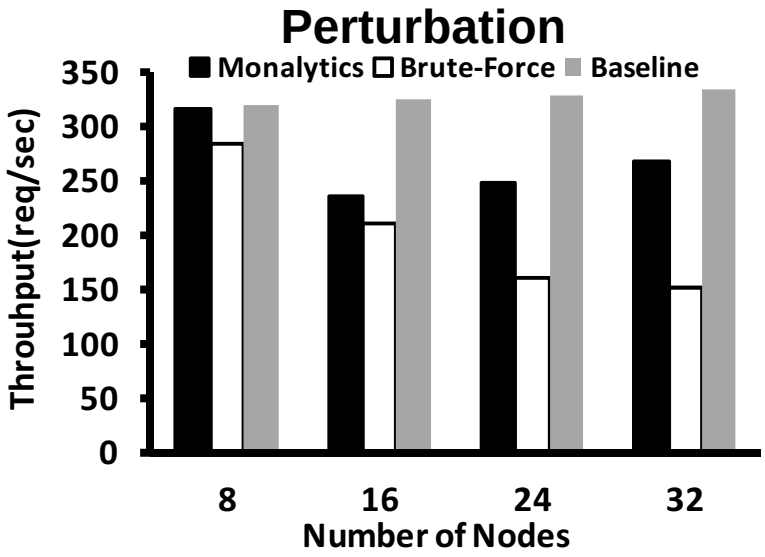
# Experimental Evaluations

- Test bed  Configuration
  > 36 Virtual Machines Hosted by 12 Blades
- Applications
  > RUBiS (Scale-Up: 8 VMs to 32 VMs)
  > Hadoop (Scale-Up: 8 VMs to 32 VMs)
- Compared with `Brute-Force' Approach in
  > Time To Insight (TTI)
  > Monitoring Data Size
  > Perturbation

# Experimental Results (ZIA)



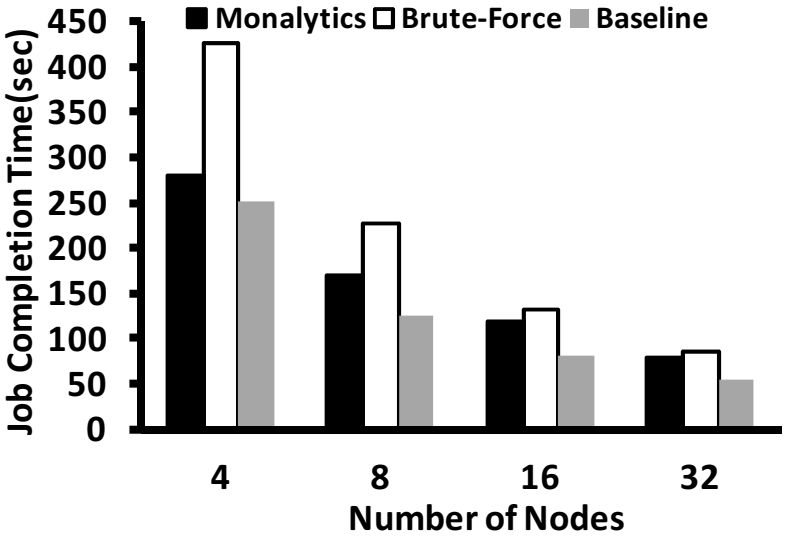



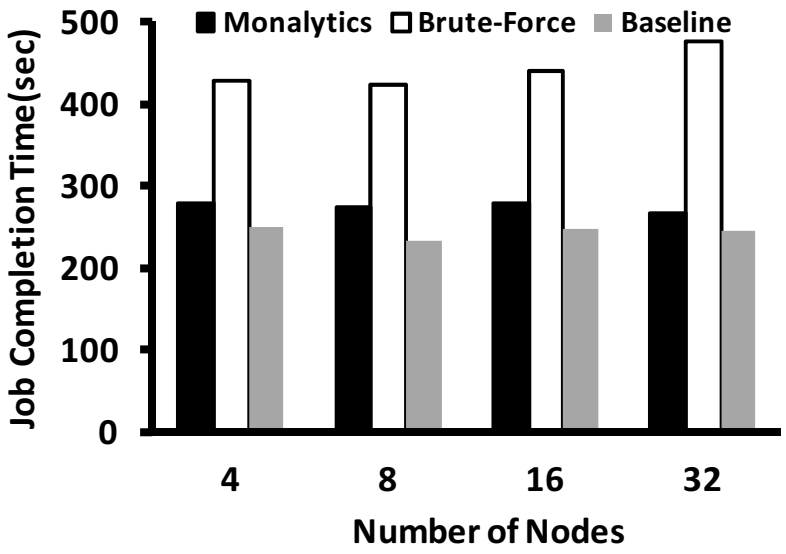

**The flexible DCG deployment saves network bandwidth.**

Intel Science & Technology Center for Cloud Computing

Mellon University Georgia Tech intel PRINCETON UNIVERSITY UC Berkeley

# Experimental Results (VMC)



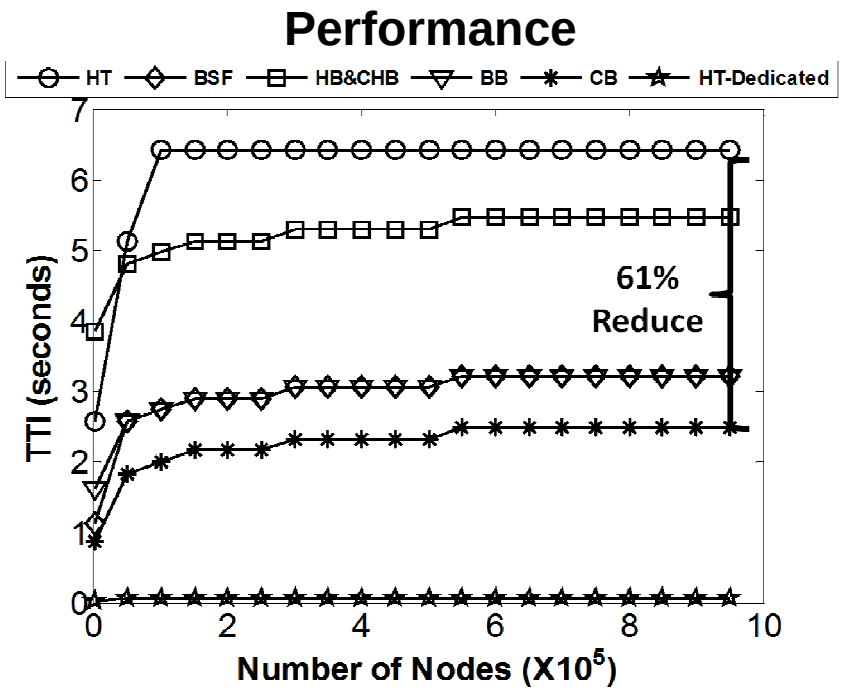

**The flexible DCG deployment saves CPU Cycles**

# Analytical Evaluations

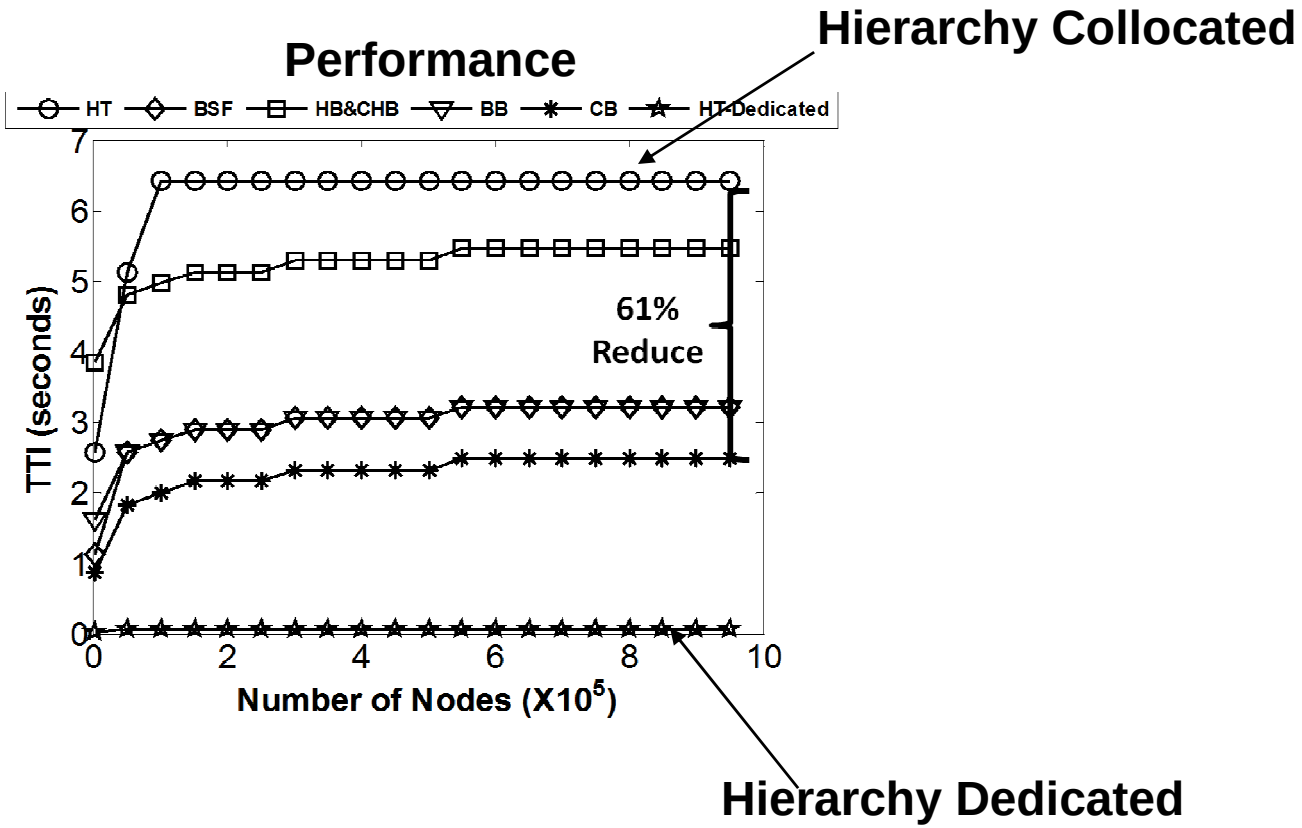
- Purpose
  - > DCG in Large Scale (1 million nodes)
  - > Viability of Hybrid DCGs
  - > Needs for Flexible Topologies
- Compared with Traditional Topologies
  - > Accommodating Different Analytic Characteristics
  - > Time To Insight (TTI)
  - > Capital Cost

# Hybrid DCGs

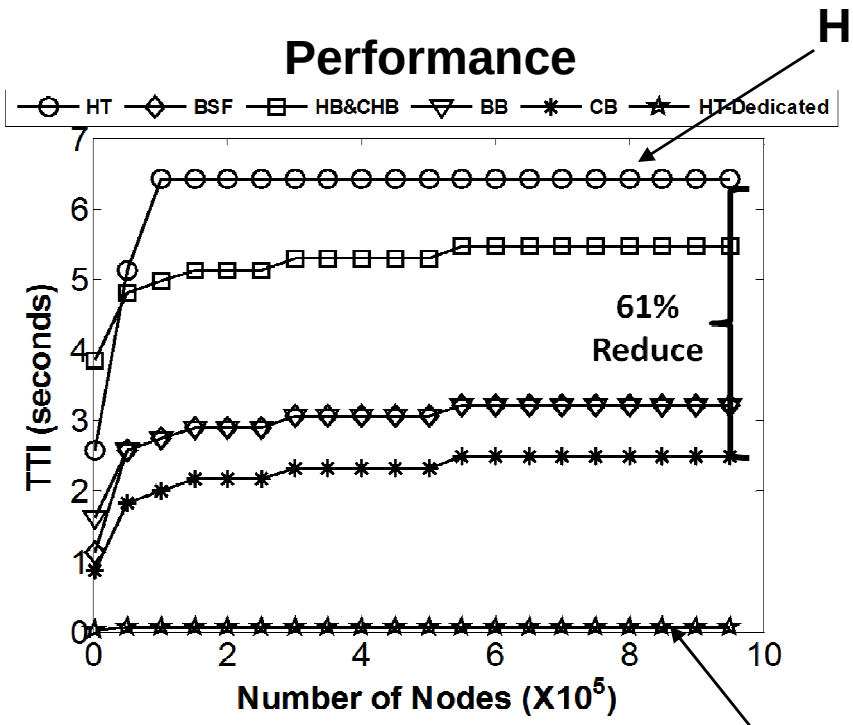
# Hybrid DCGs

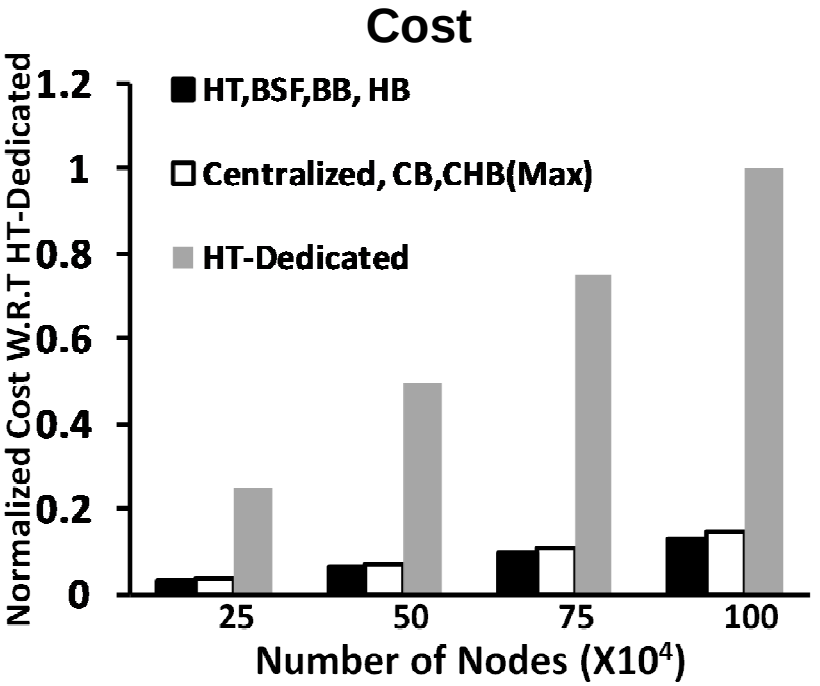

Performance
HT
BSF
HB&CHB
BB
CB
HT-Dedicated
TTI (seconds)
Number of Nodes (X10^5)
61%
Reduce
0
1
2
3
4
5
6
7
0
2
4
6
8
10

# Hybrid DCGs

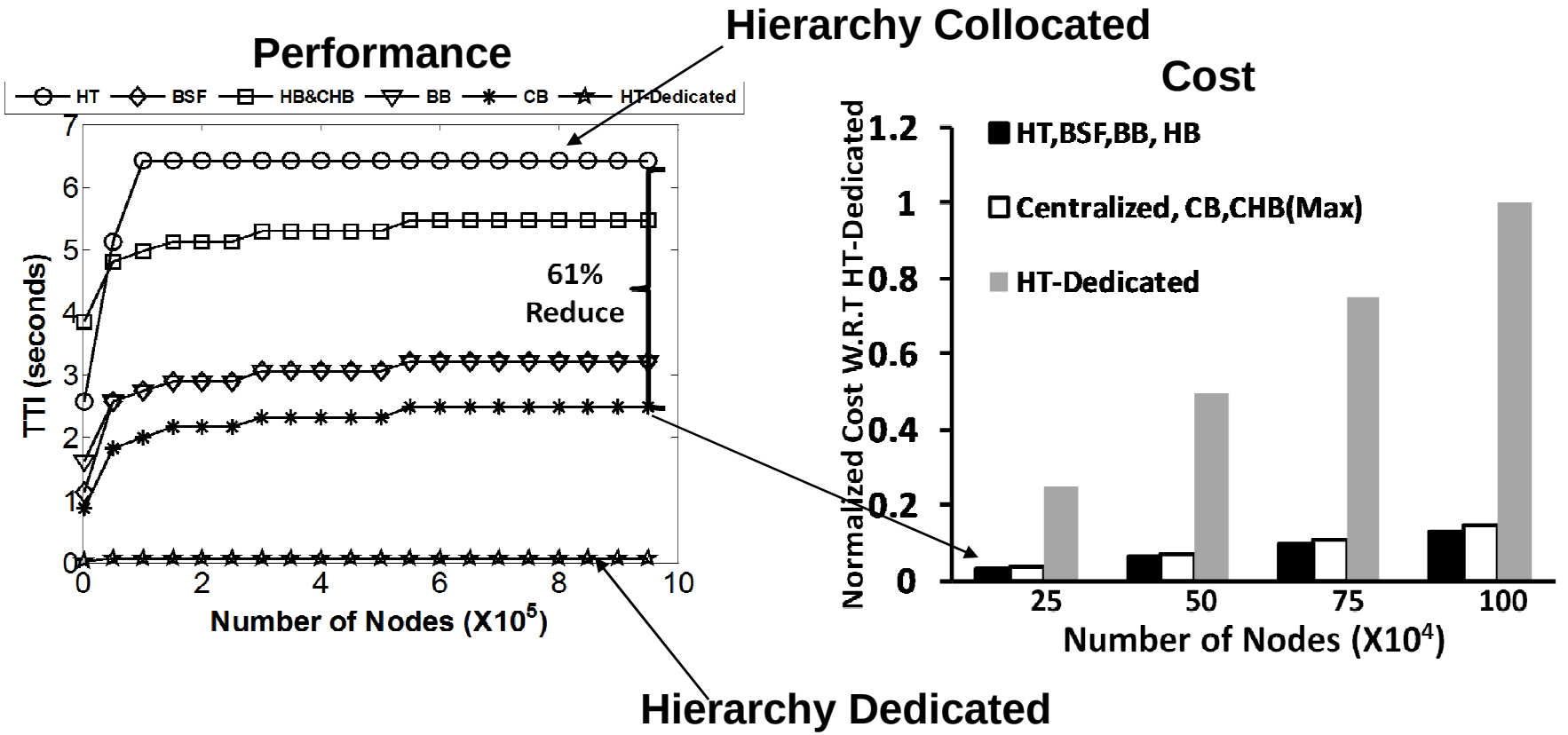

Performance
HT
BSF
HB&CHB
BB
CB
HT-Dedicated
TTI (seconds)
Number of Nodes (X10^5)
61%
Reduce
Hierarchy Collocated
Hierarchy Dedicated

# Hybrid DCGs

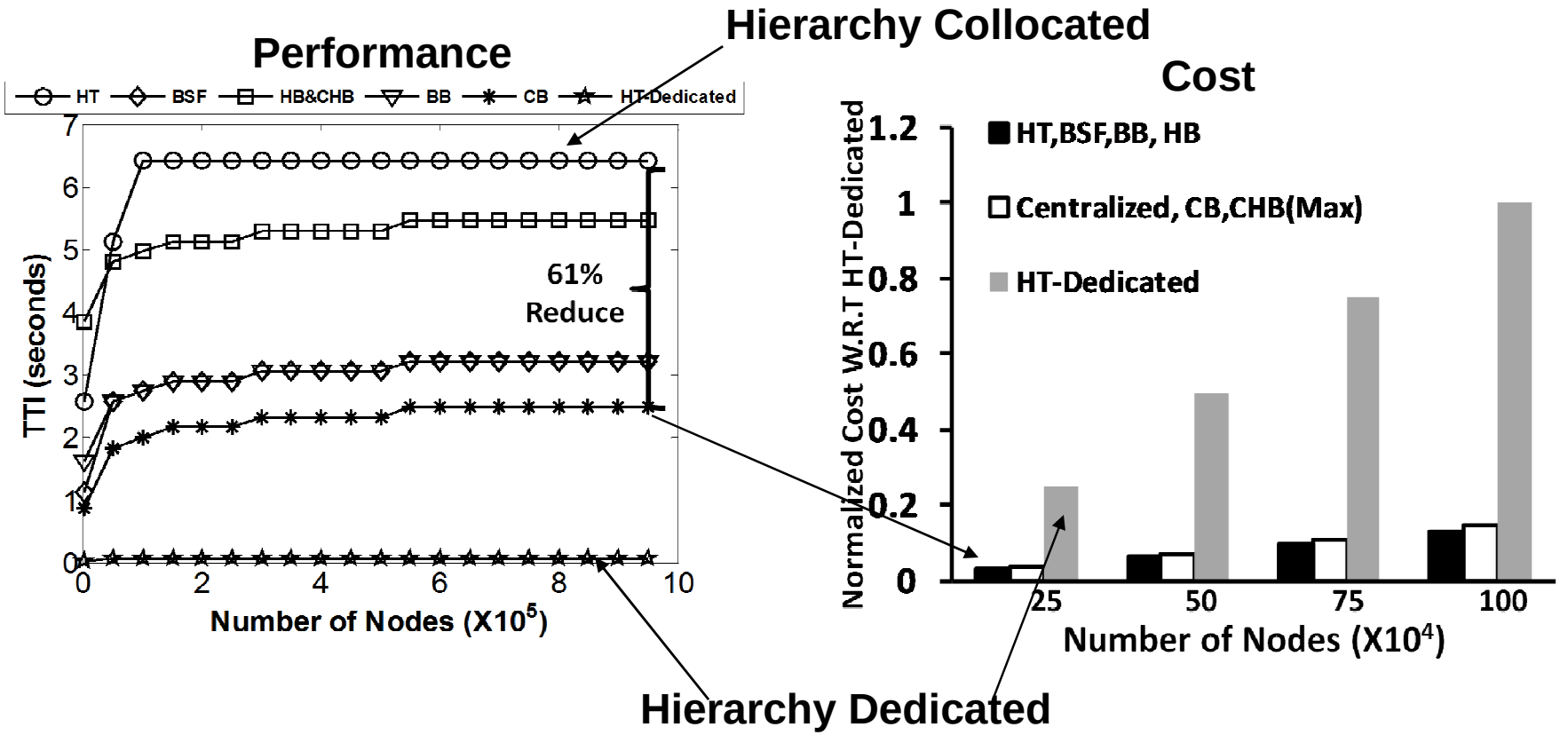

Performance
HT
BSF
HB&CHB
BB
CB
HT-Dedicated
TTI (seconds)
Number of Nodes (X10^5)
61%
Reduce
Hierarchy Collocated
Hierarchy Dedicated


Cost
Normalized Cost W.R.T HT-Dedicated
HT,BSF,BB, HB
Centralized, CB,CHB(Max)
HT-Dedicated
Number of Nodes (X10^4)

# Hybrid DCGs

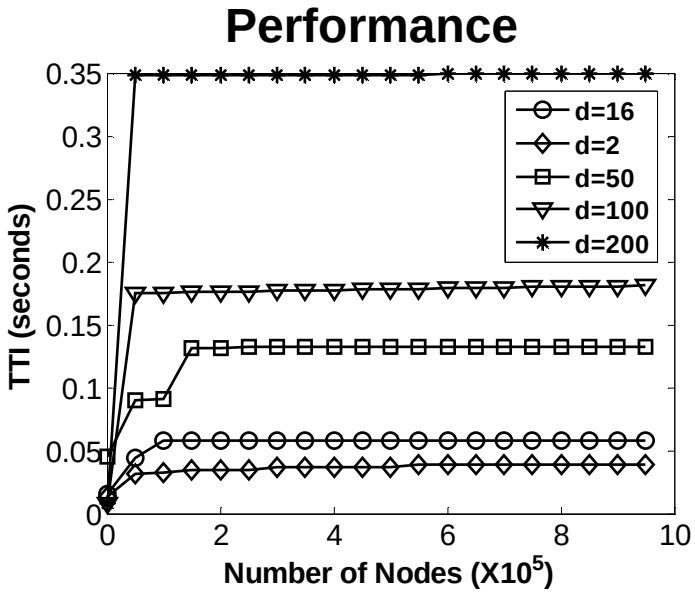

Hierarchy Collocated
Performance
HT
BSF
HB&CHB
BB
CB
HT-Dedicated
TTI (seconds)
Number of Nodes (X10^5)
61%
Reduce
Cost
Normalized Cost W.R.T HT-Dedicated
HT,BSF,BB, HB
Centralized, CB,CHB(Max)
HT-Dedicated
Number of Nodes (X10^4)
Hierarchy Dedicated

# Hybrid DCGs

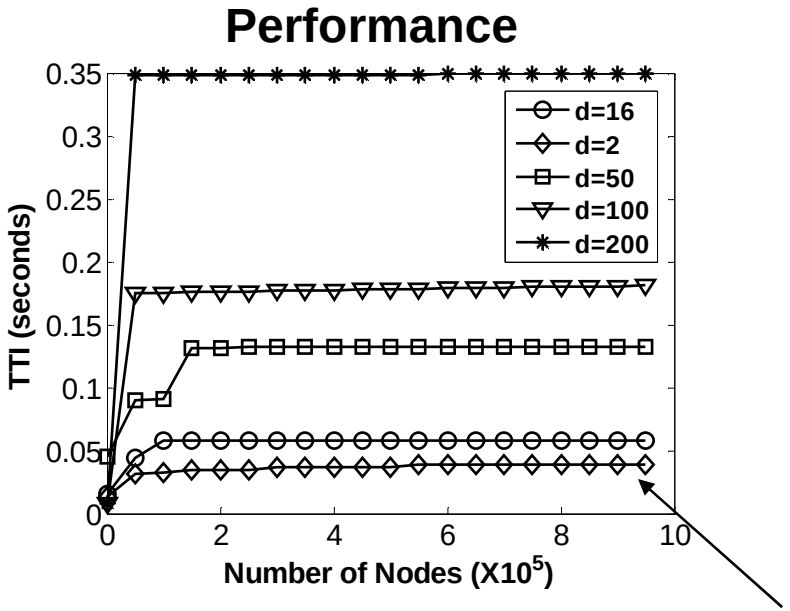

Hierarchy Collocated
Performance
HT
BSF
HB&CHB
BB
CB
HT-Dedicated
TTI (seconds)
61%
Reduce
Number of Nodes (X10^5)
Cost
Normalized Cost W.R.T HT-Dedicated
HT,BSF,BB, HB
Centralized, CB,CHB(Max)
HT-Dedicated
Number of Nodes (X10^4)
Hierarchy Dedicated

# Flexibility In Need

# Flexibility In Need

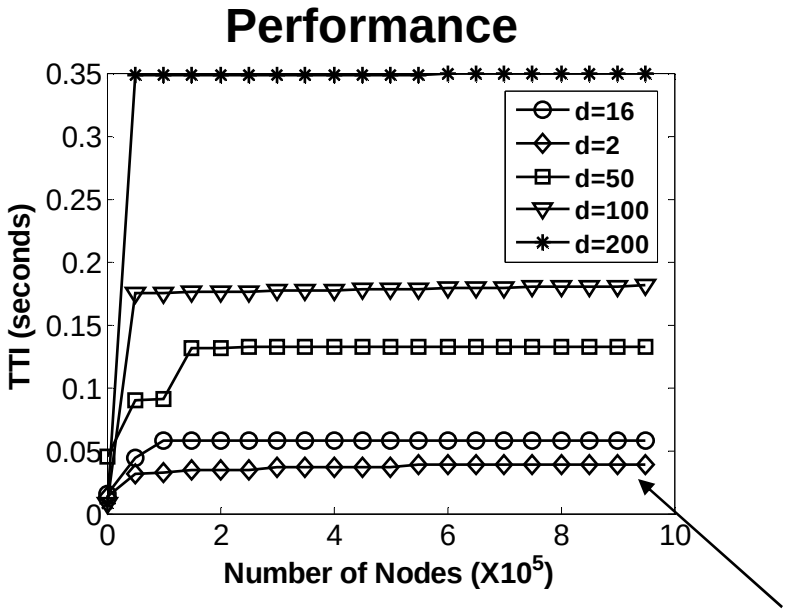


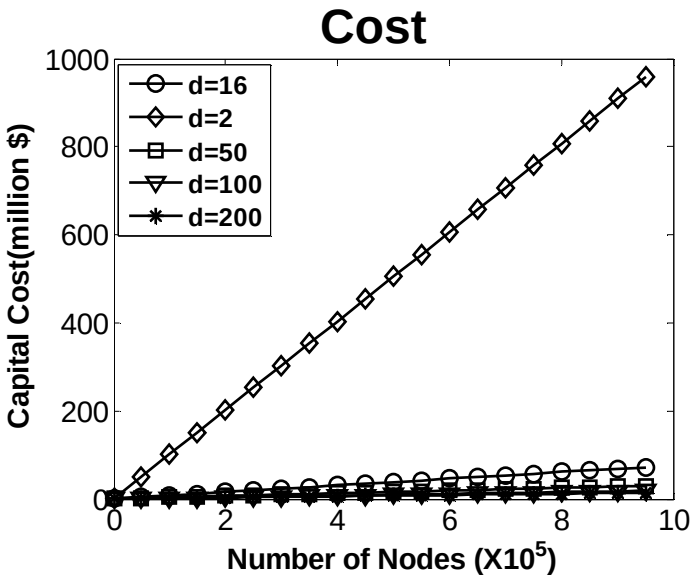
# Flexibility In Need




**Best Performance**

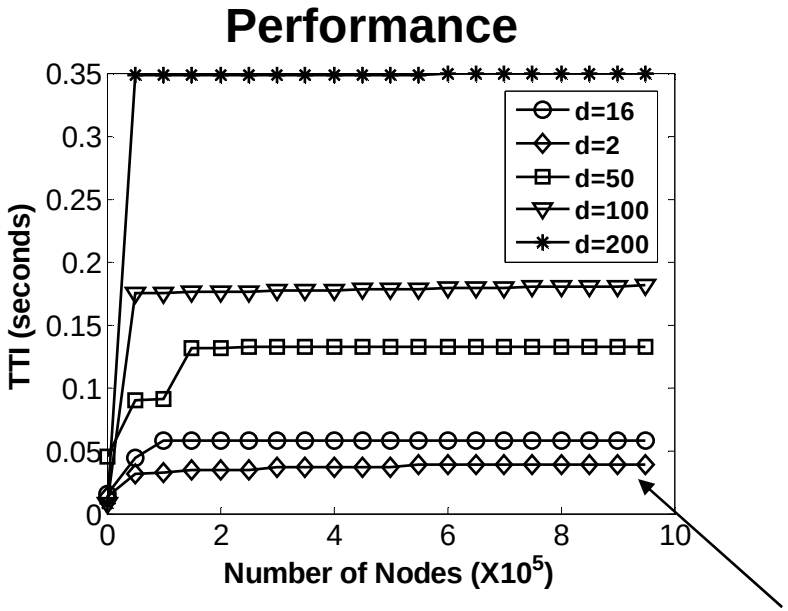
# Flexibility In Need

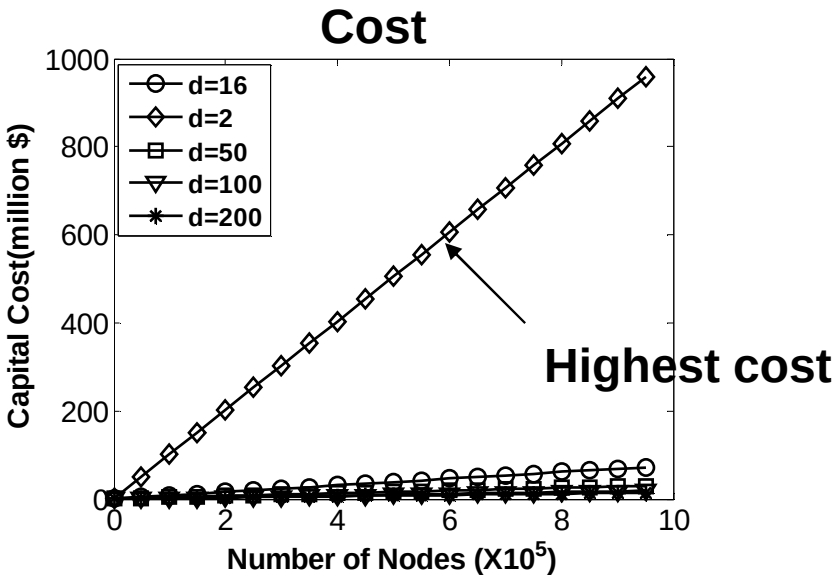



**Best Performance**

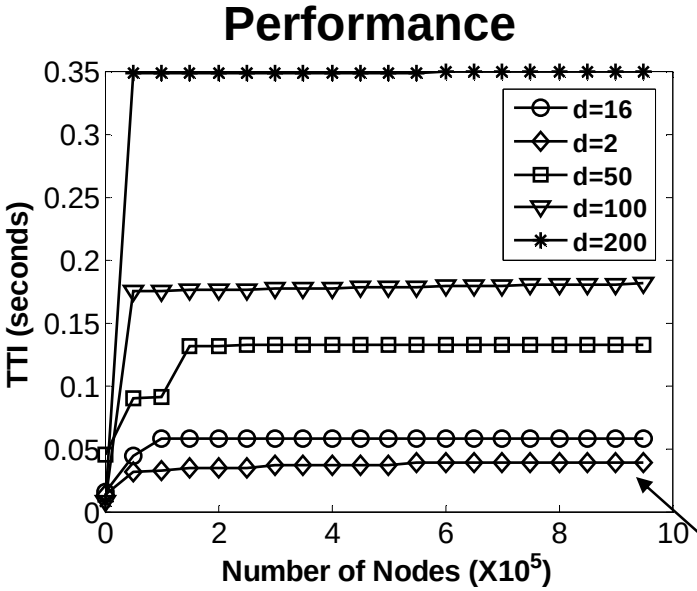
# Flexibility In Need

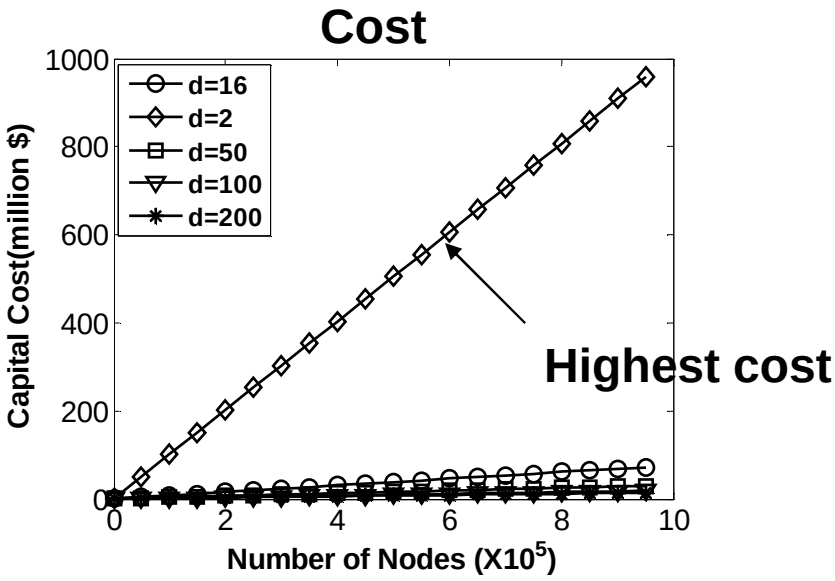


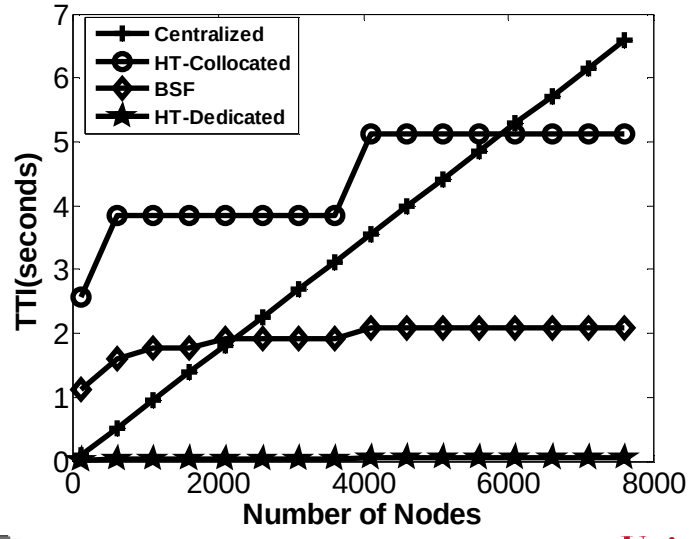

Best Performance

# Flexibility In Need

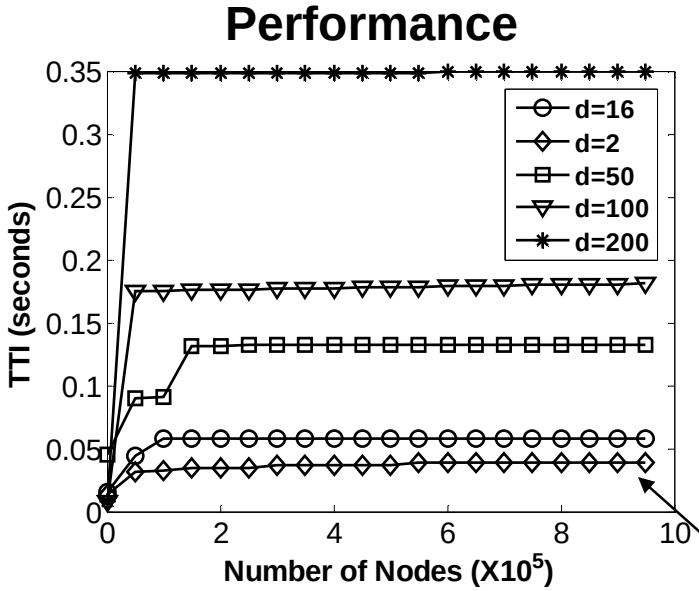


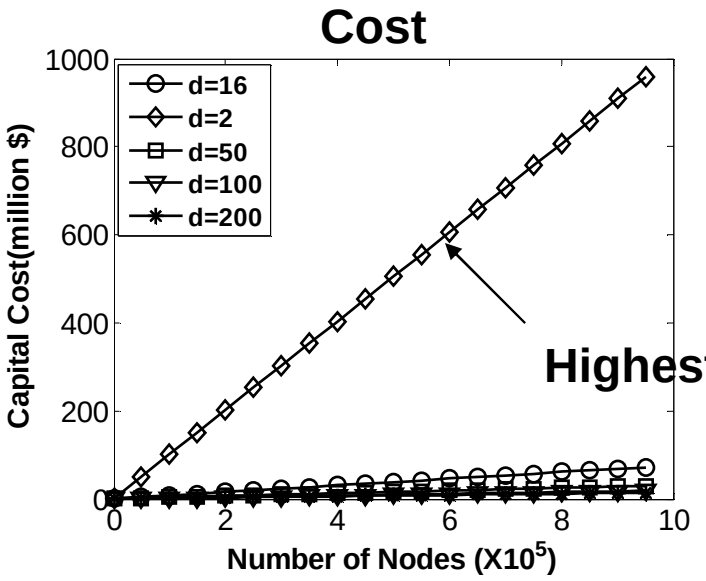




Best Performance

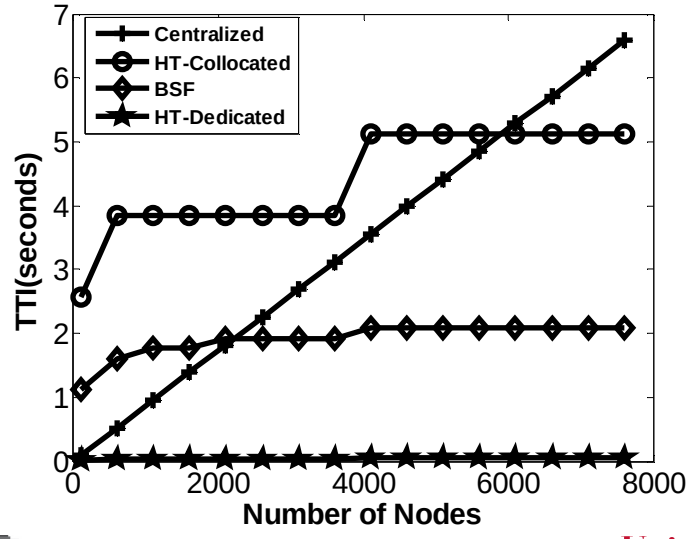


# Flexibility In Need




Best Performance




**No " one size fits all"**

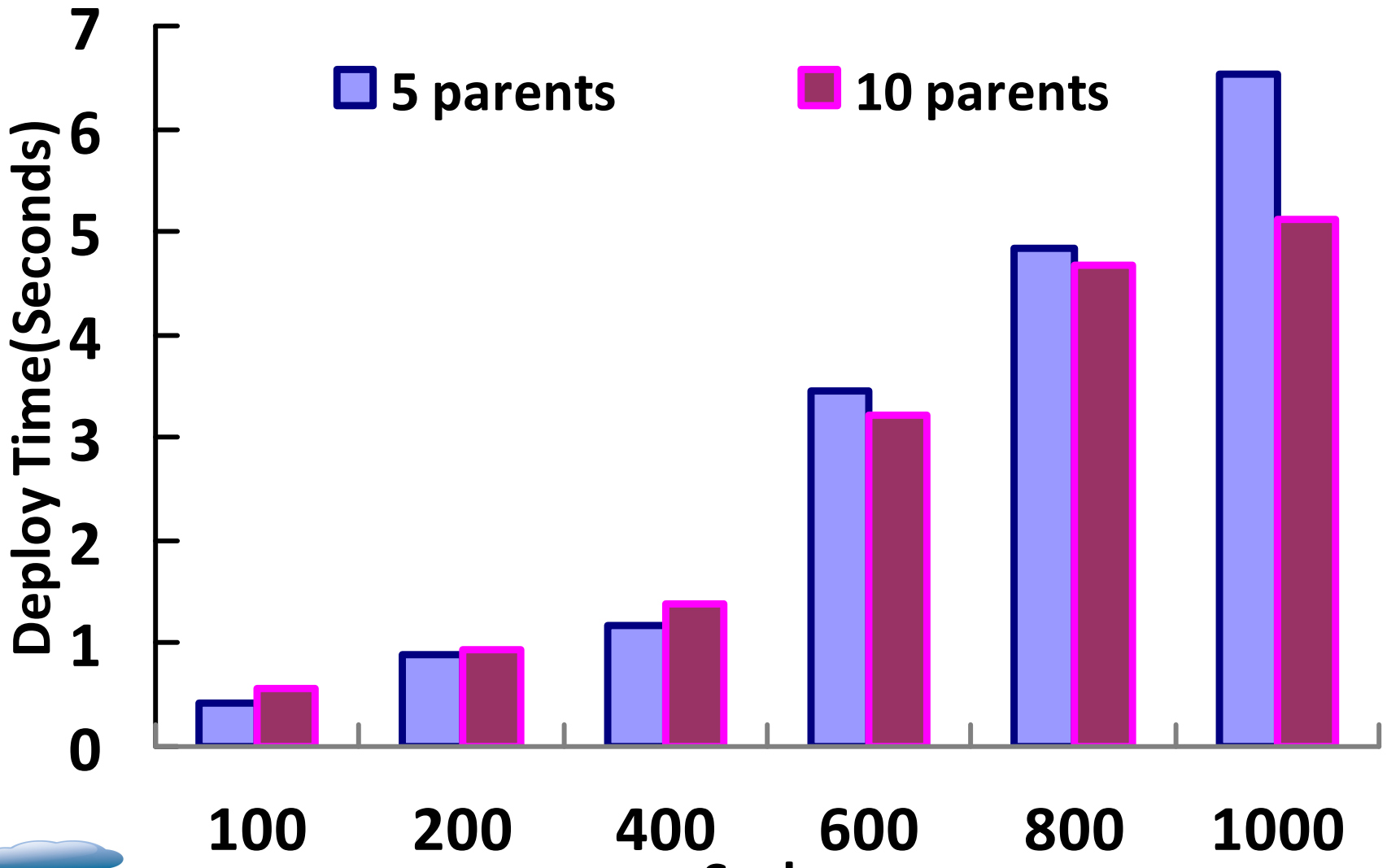
# Summary

- Proposed flexible architecture integrating large-scale monitoring and online analytics
- In-Situ DCG yields better analysis performance and less perturbation.
- Analytical models validating flexibility of DCG in large scale.

# Work in Progress

- VersaScope: Design and Implementation of a Monalytics-based system
- Test VersaScope on 1000+ VMs
- Mechanism for “Smart” DCG Deployment
- Stream Processing Usecases

# VersaScope Preliminary Results




Intel Science & Technology Center for Cloud Computing

Carnegie Mellon University · Georgia Tech · intel · PRINCETON UNIVERSITY · UC Berkeley

# Thanks! Questions?