Automatic clustering of similar VM to improve the scalability of monitoring and management in IaaS cloud infrastructures

C. Canali
R. Lancellotti

University of Modena and Reggio Emilia
Department of Engineering "Enzo Ferrari"

WEBLab

- **WEBLab: Web Engineering and Benchmarking Lab**
- **Contributing to**
  - DIEF - Department of Engineering "Enzo Ferrari" *(not only automotive)*
  - CRIS - Research center of Security
- **Research interests**
  - Distributed systems
  - Cloud computing
  - Performance / scalability issues
  - Monitoring in distributed systems
  - Security in networked / cloud systems
  - ...

Agenda

- **Background and motivation**
  - IaaS Cloud
  - Reference scenario
  - Traditional approach vs. clustering
  - Impact on monitoring and management
- **Clustering based on metric correlation**
  - Theoretical model(s)
  - Experimental evaluation
- **Clustering based on Bhattacharyya distance**
  - Theoretical model(s)
  - Experimental evaluation
- **Conclusion and future work**

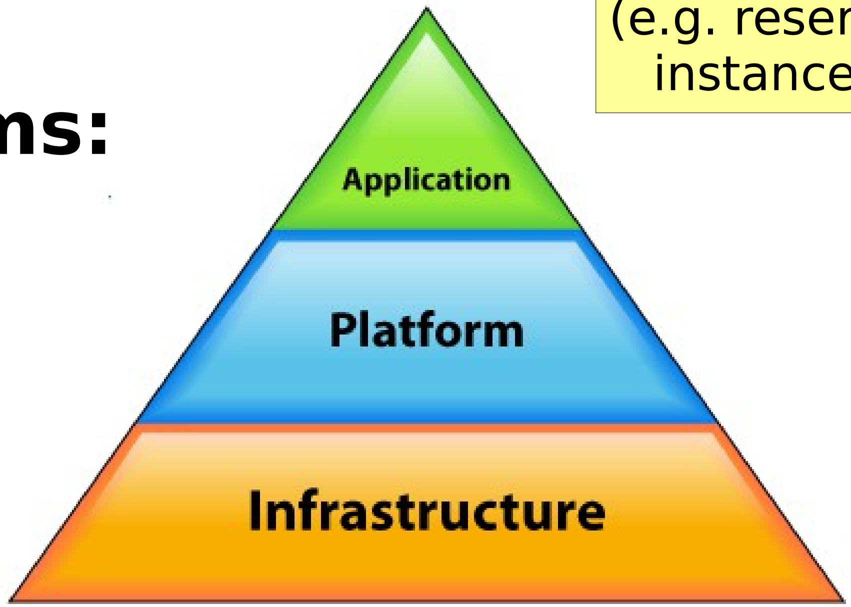
Cloud computing

Cloud computing

- **Cloud computing AKA Utility computing**
- **Access to resources and services:**
  - Multiple customers → same provider
  - Leveraging economies of scale
  - No initial cost (pay per use)
  - Exploit virtualization technologies

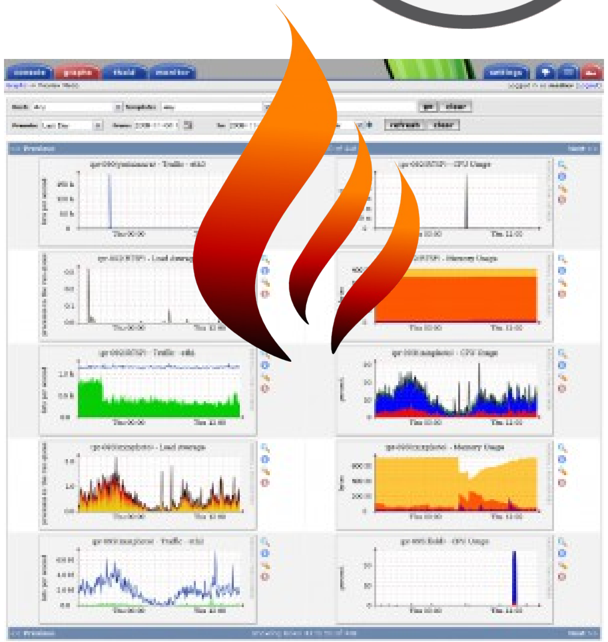
NOTE:
We may still have long-time commitments (e.g. reserved instances)

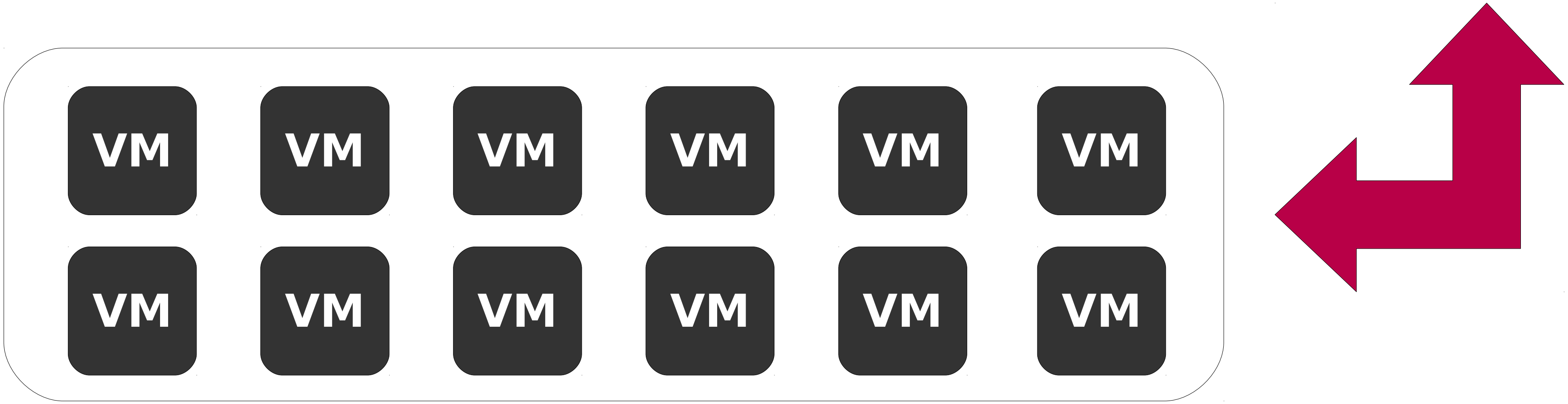
- **Multiple cloud paradigms:**
  - SaaS
  - PaaS
  - **IaaS**



Challenges: monitoring

- **Large data centers (> 10^5 VMs)**
  → huge amount of data
- **Multiple data centers**
  → geographic data exchange
- **VM can be anything**
  → treat VM as black boxes
- **→ Scalability issues**





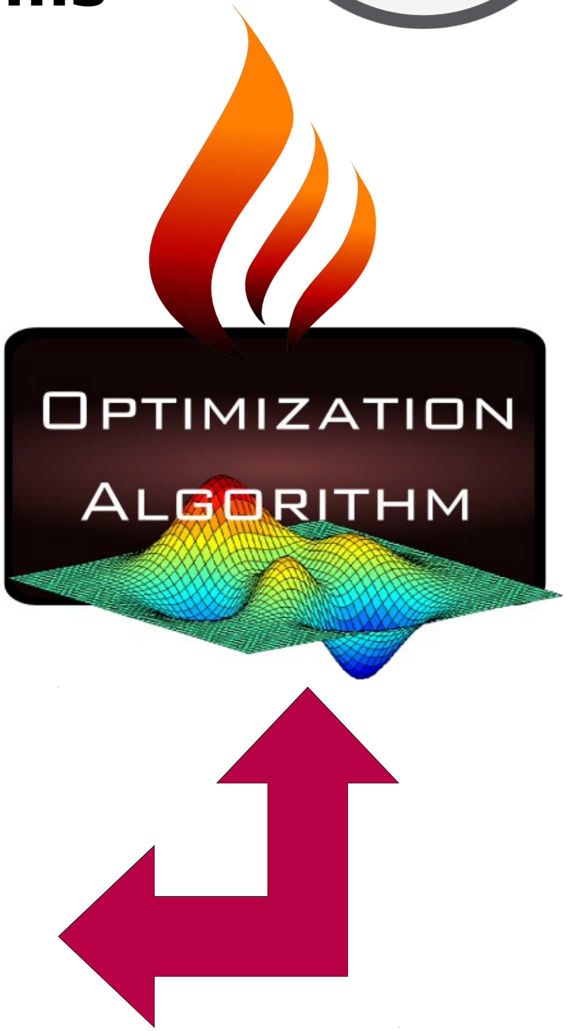
Challenges: monitoring

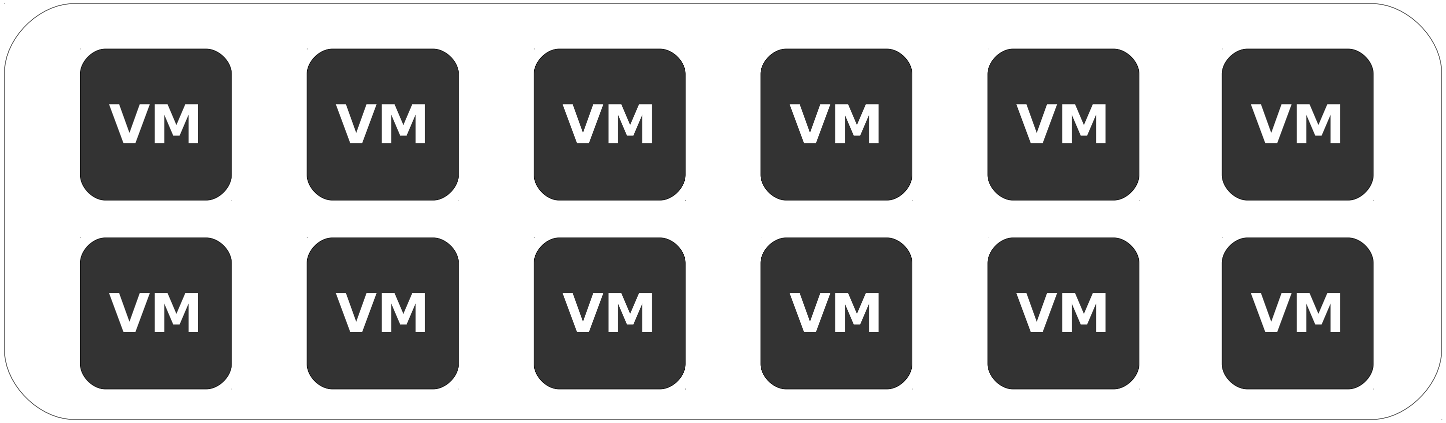
- **Current approach**
  → reduce amount of data in a uniform way:
  - Reduce sampling frequency
  - Reduce number of metrics considered

- **→ Reduced monitoring effectiveness**
  - Less information available to take management decision

Challenges: management

- **Large data centers → large opt. problems**
  - Too many variables
  - Too many bounds
  - Like a *huge* multi-dimensional tetris
- **VM can be anything**
  → treat VM as black boxes
  → difficult search for complementary workloads
- **→ Scalability issues**





Challenges: management

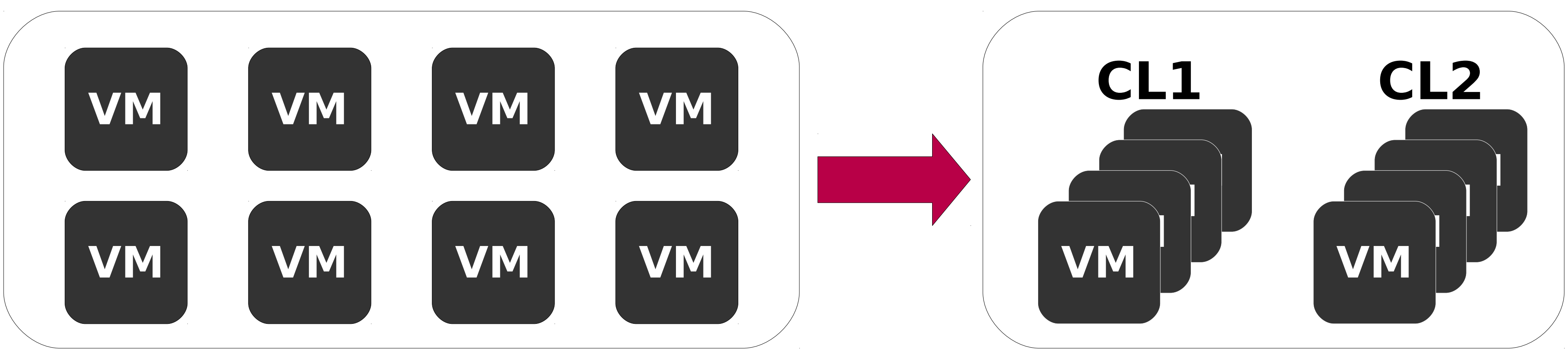
- **Current approach**
  → reduce amount of bounds:
    - Assume VM resource utilization constant over long periods (e.g. day/night)
    - Reduce number of metrics considered
    - Consider only nominal resource utilization

  → rely on hierarchical management

- **→ Reduced management effectiveness**
    - No support for fine grained management
    - Sub-optimal management decisions

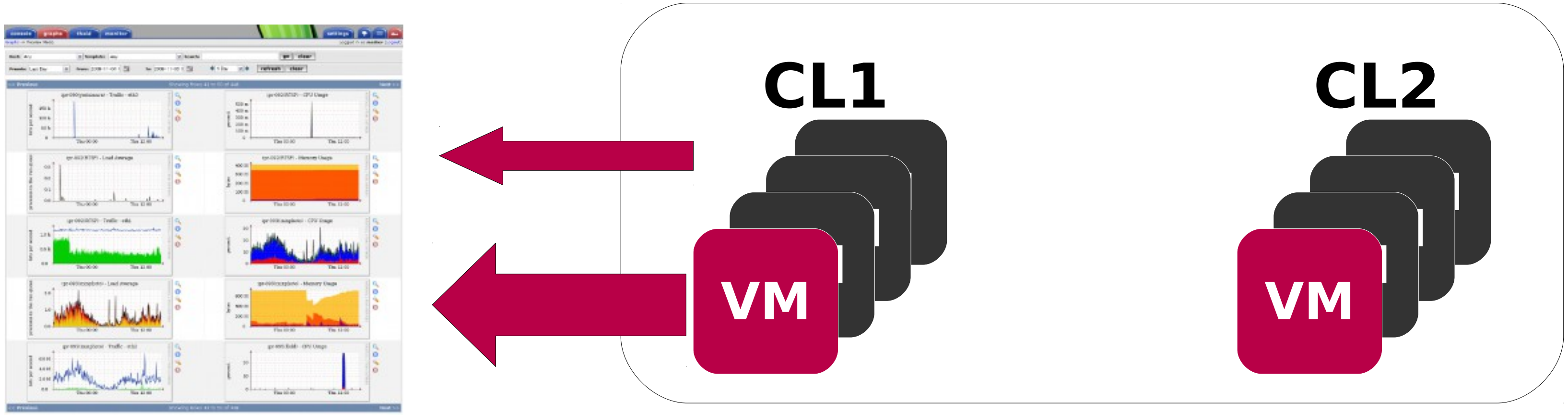
Exploiting VM similarity

- **No information on VM behavior is used to improve scalability**
- **Proposal: <u>automatically</u> cluster VMs with similar behavior**
- **Requirements:**
  - No human intervention
  - No models for VM classes
  - No crystal ball



Improving monitoring scalability

- **Group similar VMs together**
- **Elect a few (e.g., 3) cluster representatives**
  - Support for byzantine failures in representatives
- **Detailed monitoring of cluster representatives**
- **Reduced monitoring of other VMs**



Improving monitoring scalability

- **Numeric example**
- **Every VM as a black box:**

  - 1000 VMs, K metrics,
    1 sample/5 min
  - → 288 10^3 K sample/day
- **With clustering:**

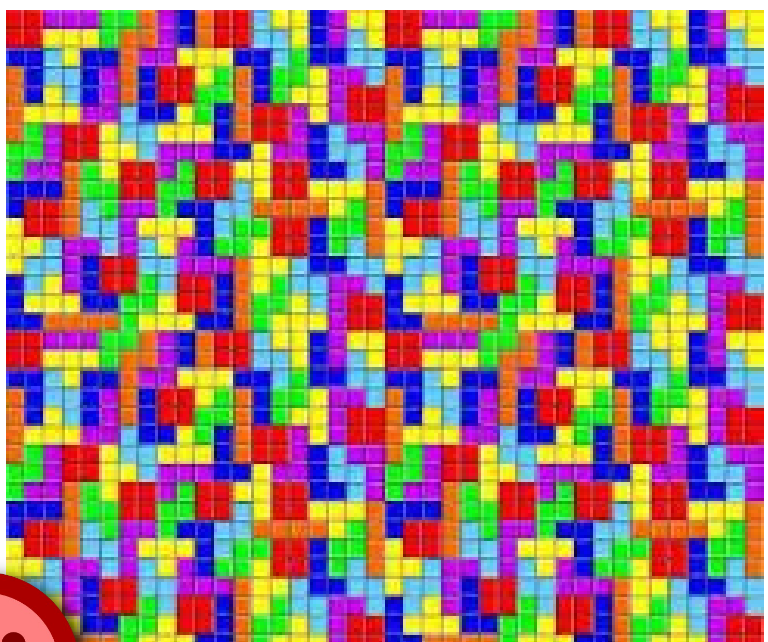
  - 15 clusters, 67 VMs per cluster
  - 3 representative per cluster
    → 45 VMs, K metrics, 1 sample/5 min

  - Non representatives
    → 955 VM, K metrics 1 sample/6 hour
  - → 16,8 10^3 K sample/day
- **Data collected reduced by 17:1**

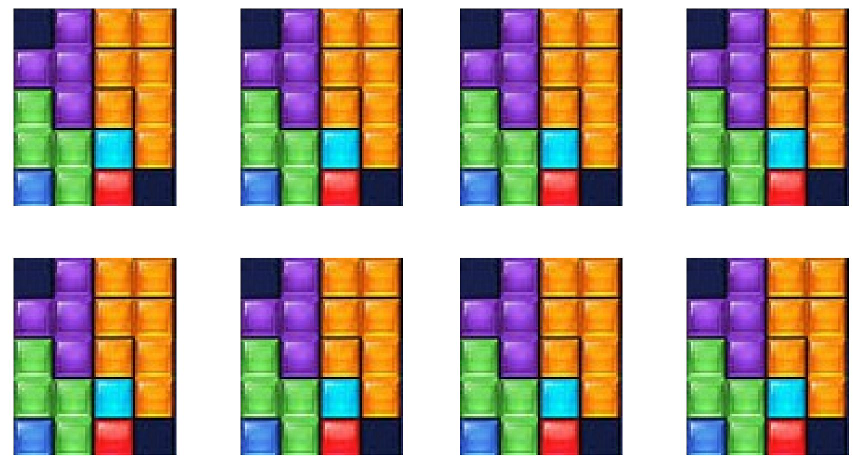
Improving management scalability

- **Server placement and consolidation**
- **Build a small consolidation solution**
- **Replicate solution as a building block**

Global problem



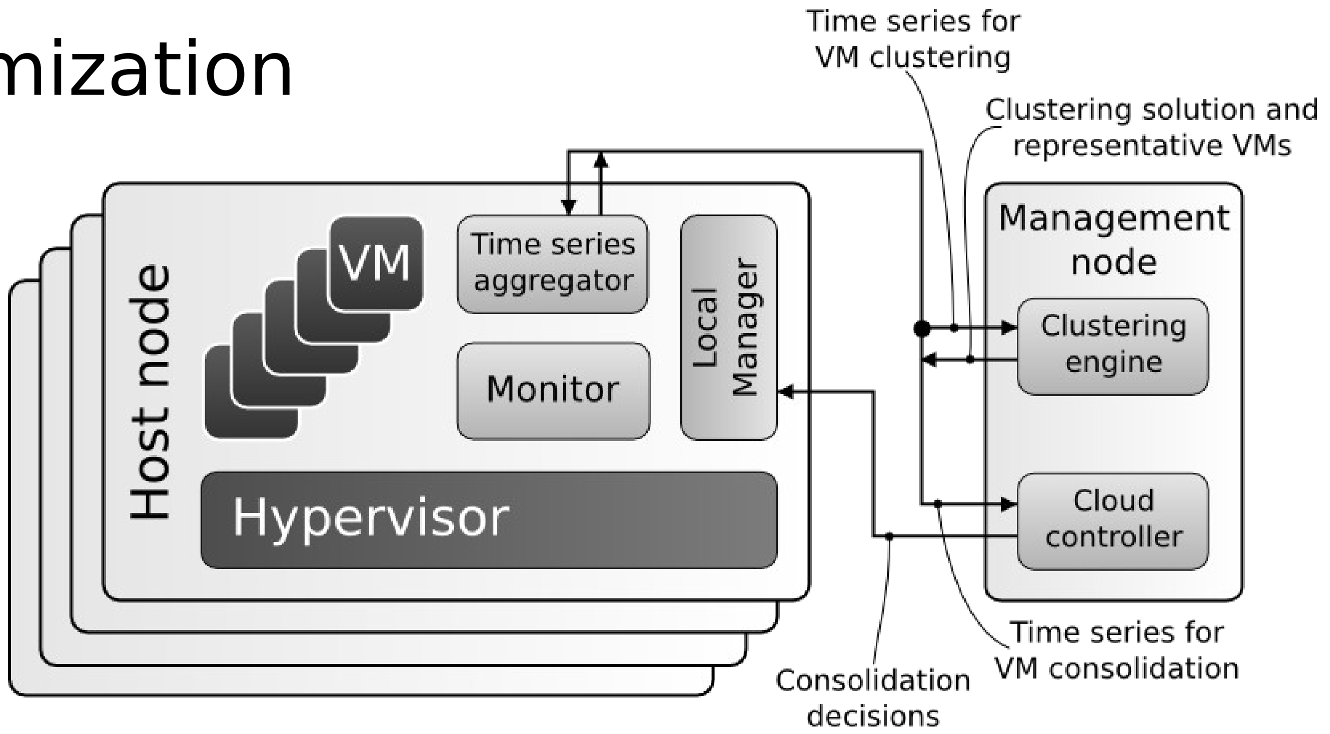
Building block solution



Residual problem solution

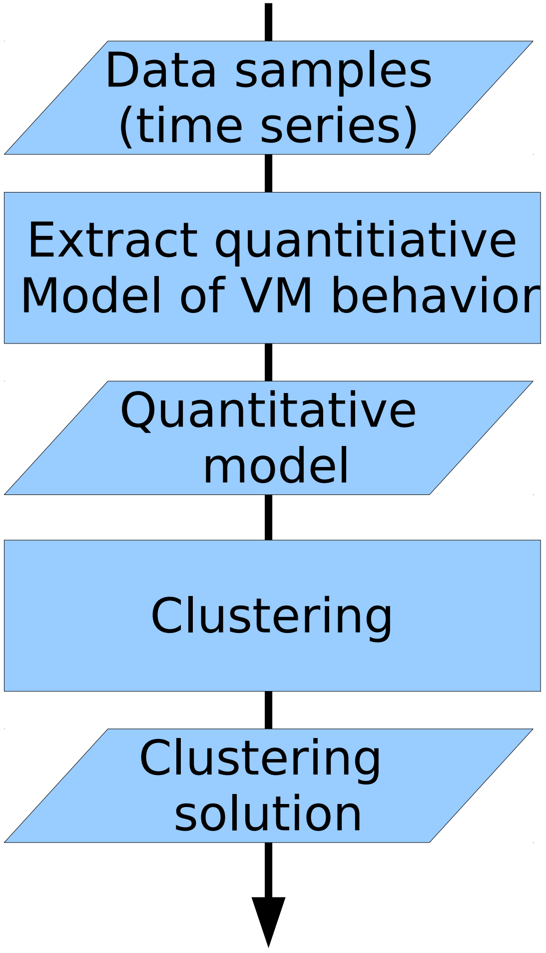
Reference scenario

- **IaaS with long term commitment**
  - Amazon Reserved instances, private cloud
- **Reactive VM relocation**
  - Local manager
- **Periodic global consolidation**
  - Global optimization



Proposed methodology

- **Methodology:**
  - Define quantitative model for VM behavior
  - Cluster similar VM together
- **Elect a few (e.g., 3) cluster representatives**
- **Fine-grained monitoring of cluster representatives**
- **Reduced monitoring applied to other VMs**
  - Reduced number of metrics
  - Lower sampling frequency



Design choices

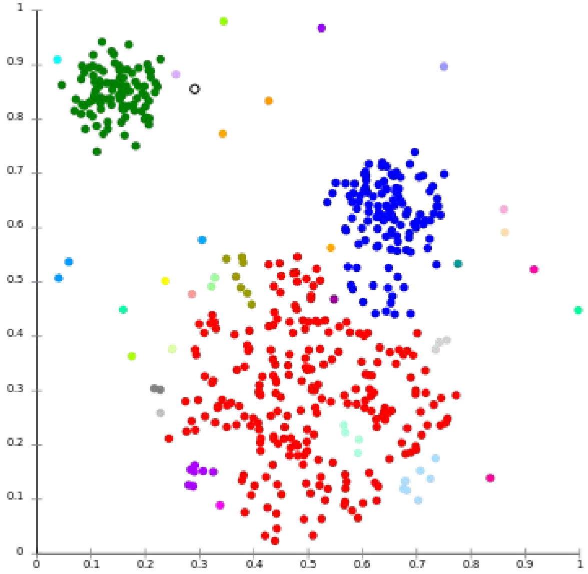
- **How to represent VM behavior?**
- **Use correlation between metrics**
  - Possible enhancement: use PCA
- **Use probability distribution of metrics**
  - Use histograms & Bhattacharyya distance
  - May need to select which information are "useful"
  - Must merge heterogeneous information from multiple metrics
  - May exploit ensemble techniques to provide robust performance
  - Possible enhancement: use histogram smoothing

Design choices

- **How to perform clustering?**

- **Use K-Means**
  - When VM behavior is represented as a feature vector

- **Use spectral clustering**
  - When VM behavior can be used to compute distance/ similarity between VMs

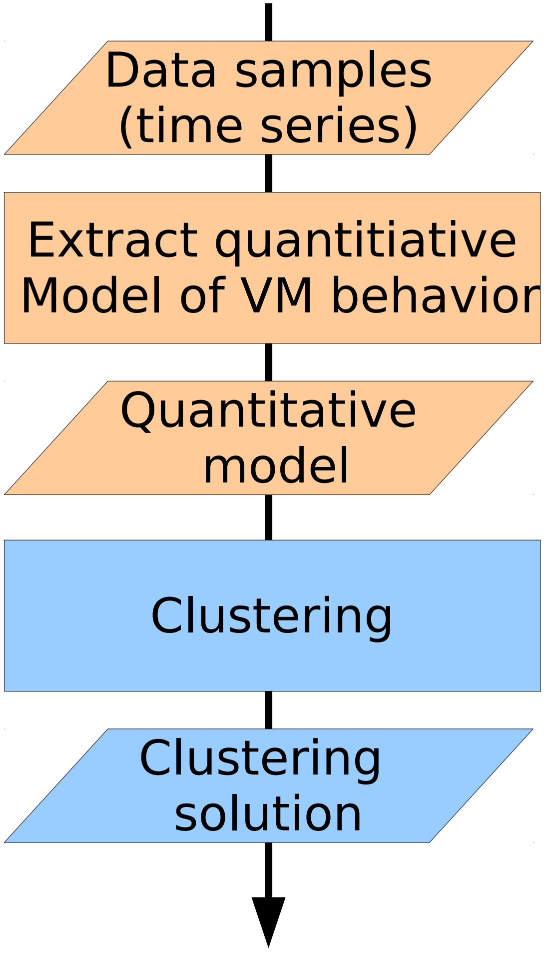


Agenda

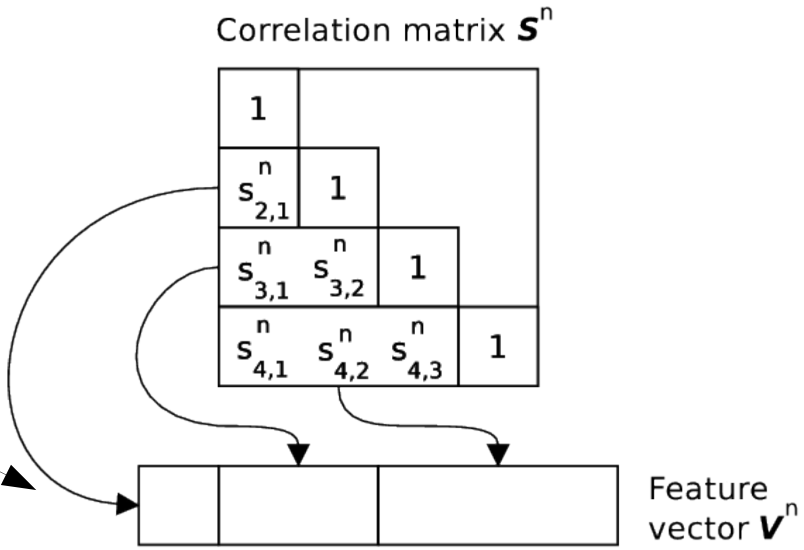
- **Background and motivation**
  - IaaS Cloud
  - Reference scenario
  - Traditional approach vs. clustering
  - Impact on monitoring and management
- **Clustering based on metric correlation**
  - Theoretical model(s)
  - Experimental evaluation
- **Clustering based on Bhattacharyya distance**
  - Theoretical model(s)
  - Experimental evaluation
- **Conclusion and future work**

Theoretical model

- **Extraction of a quantitative model of VM behavior**
  - *Input:* time series of metrics describing VM n behavior (X1, ... ,Xm)
  - Compute correlation matrix Sn for each VM n
  - *Output:* feature vectors Vn

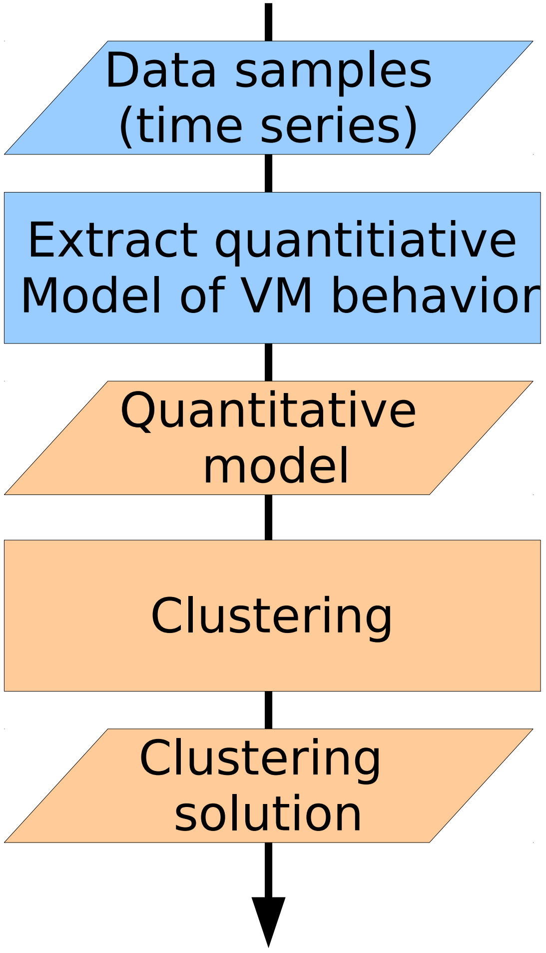


NOTE:
We exploit simmetry in matrix Sn to remove redundant information



Theoretical model

- **Clustering of VMs**
  - *Input:* feature vector Vi
  - Clustering based on k-means algorithm
  - *Output:* clustering solution



Case study

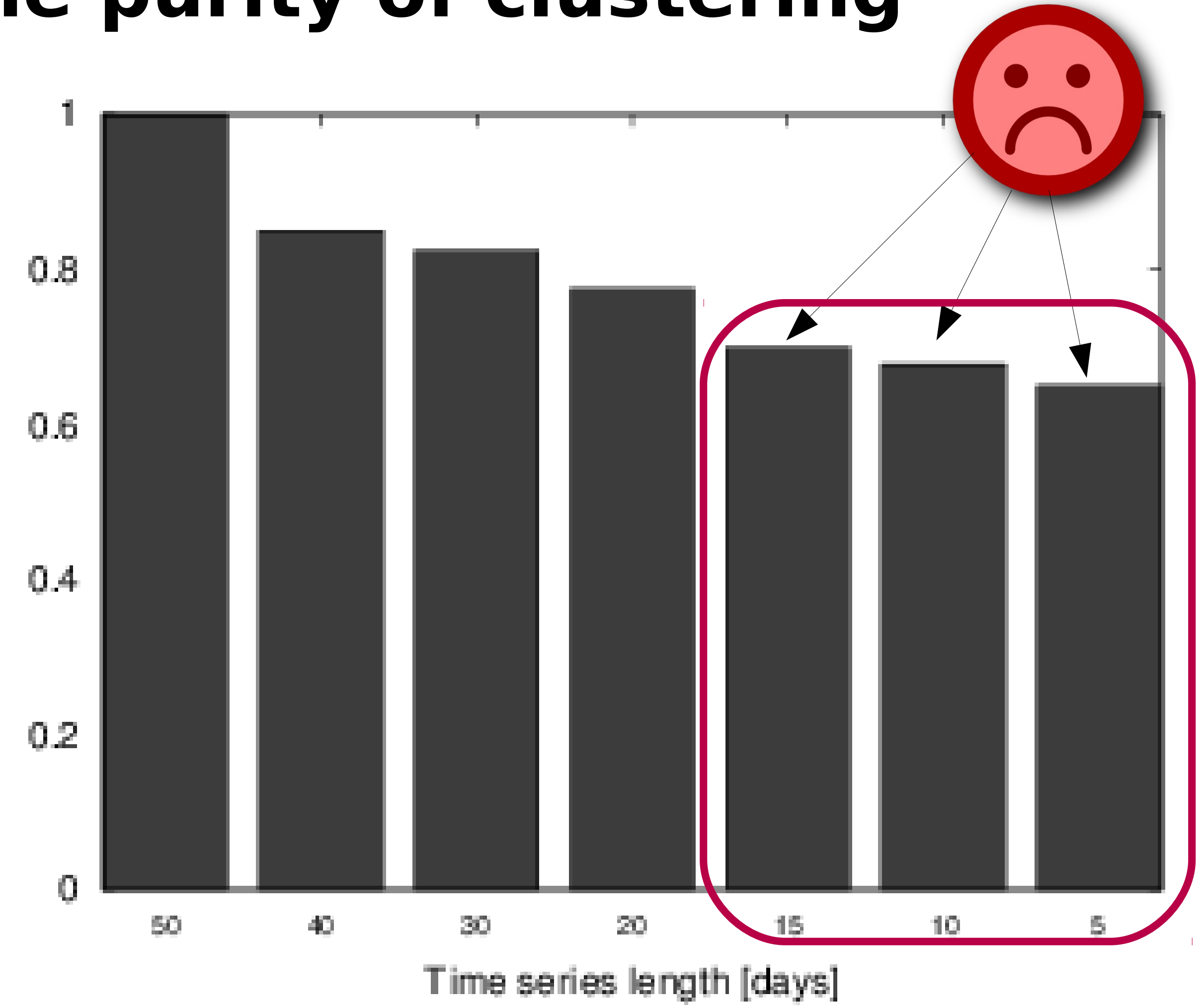
- **Datacenter supporting a e-health Web application**
  - Web server and DBMS
  - 110 VMs
  - 11 metrics for each VM,
  - Sampling frequency: 5 min

- **Goal: separate Web servers and DBMS**
  - Main metric: Purity of clustering
- **Three types of analyses**
  - Impact of time series length
  - Impact of filtering techniques
  - Impact of number of nodes

Impact of time series length

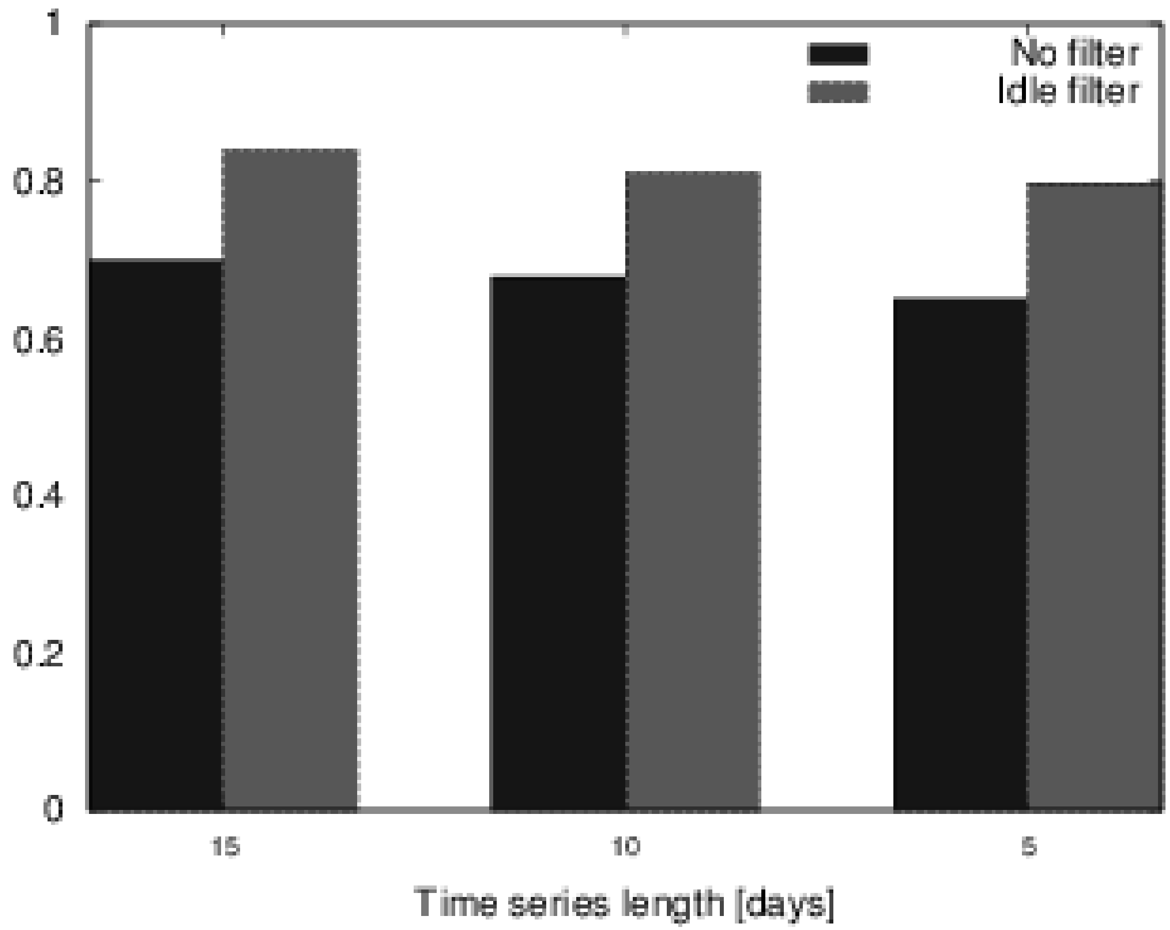
- **Reduction of available data → reduction in the purity of clustering**

- **Purity > 0.7 for time series > 20 dd**



Impact of filtering techniques

- **Application of data filtering:**
  - Remove idle periods in time series
- **Data filtering improves performance**
  - Removal of periods providing limited information
- Purity >0.8 even for 5 days time series



Impact of number of nodes

Number of VMs	Purity	Clustering time [s]
10	1	49.7
30	0.86	59.5
50	0.84	68.6
70	0.84	78.0
90	0.83	88.3
110	0.84	95.3

- **Purity is not adversely affected by # of VM**
  - **Purity ~ 0.85 for [30-110] VMs**

Proposed enhancement

- **The clustering time grows**
  - Linearly with # of VM
  - Quadratically with # of metrics

- **→ Potential scalability issue**

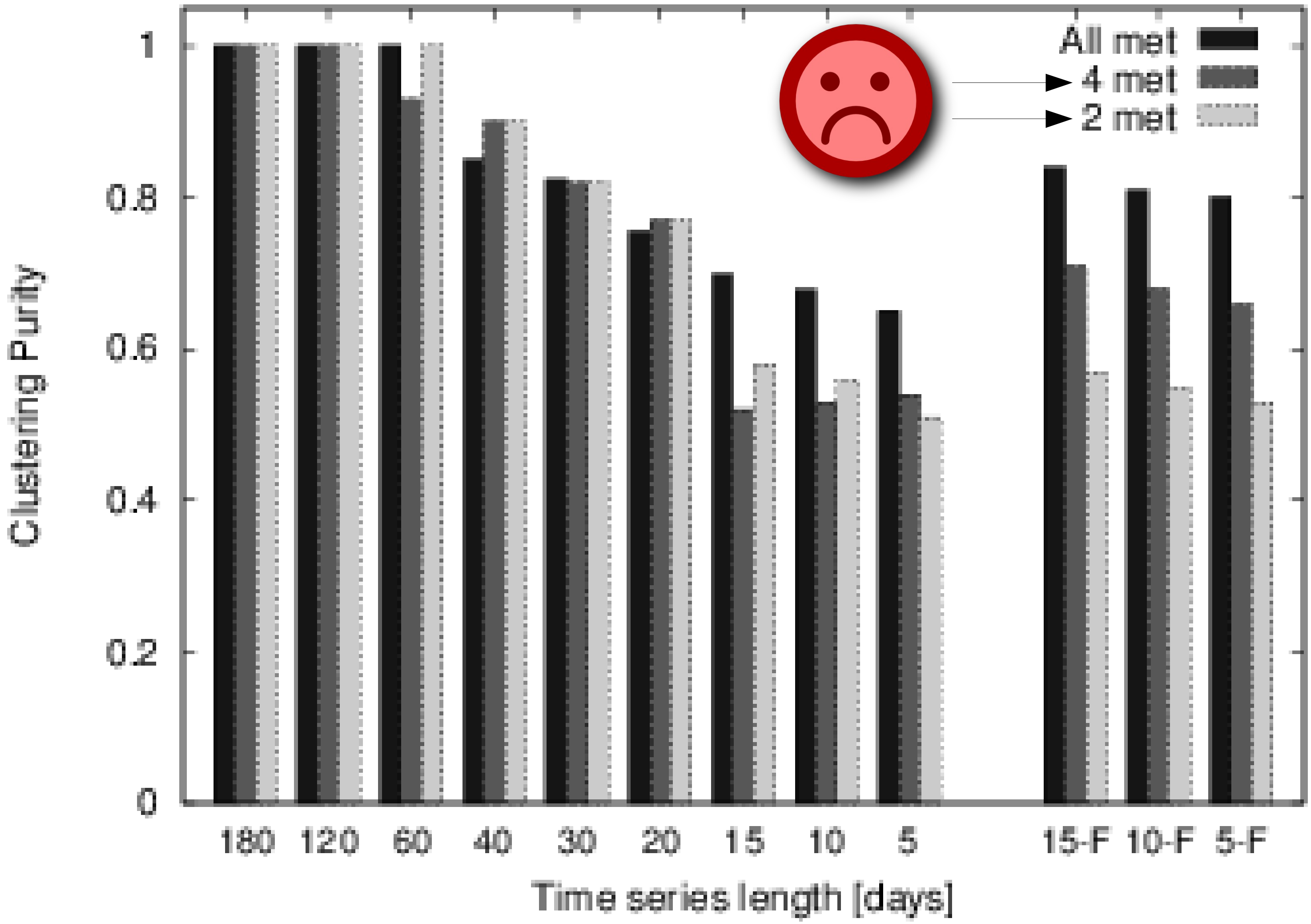
- **Can we reduce the number of metrics?**

- **Can we reduce the quadratic relationship?**

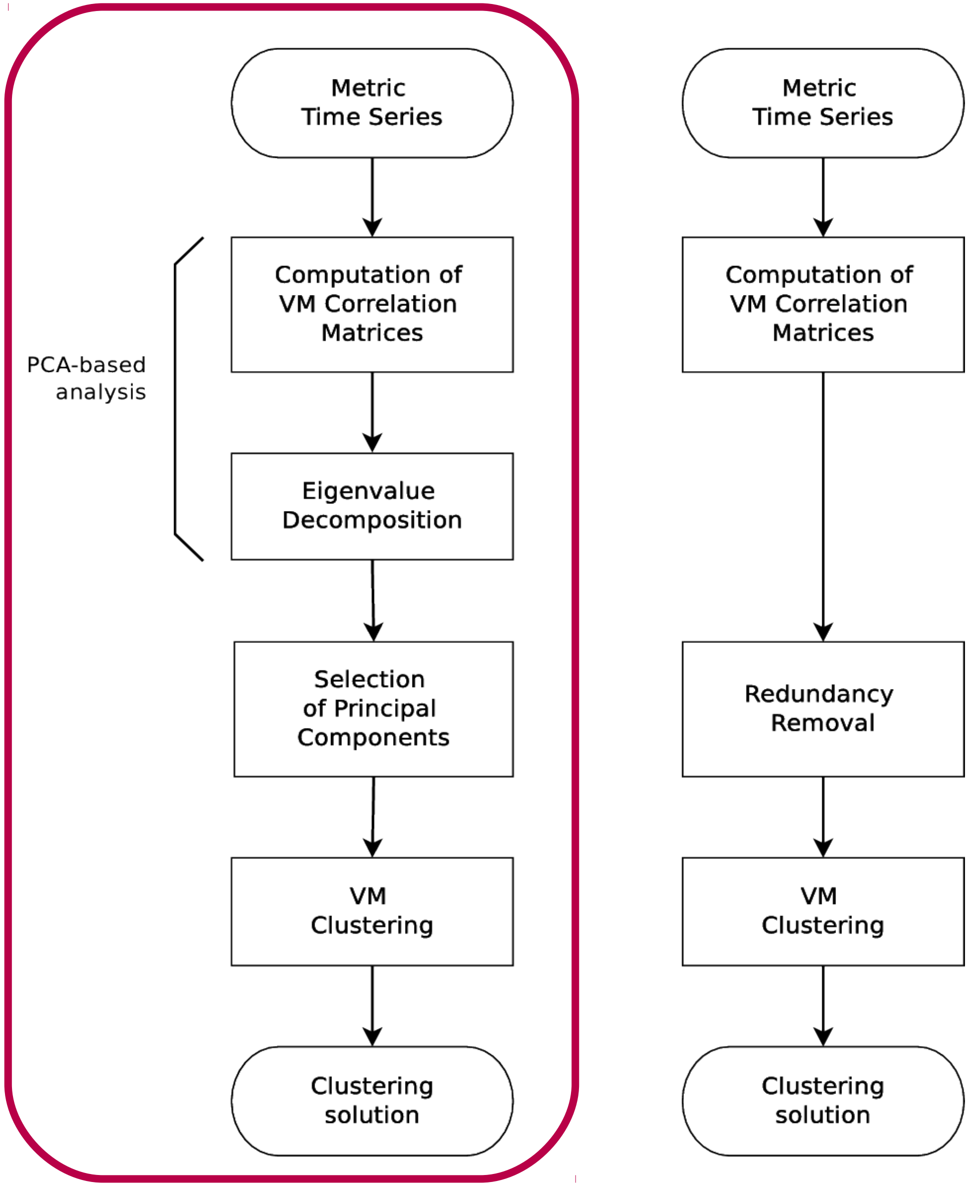
Proposed enhancement

- **The clustering time grows**
  - Linearly with # of VM
  - Quadratically with # of metrics
- **→ Potential scalability issue**
- **Can we reduce the number of metrics?**
  → NO: clustering purity is heavily affected
- **Can we reduce the quadratic relationship?**
  → YES: can exploit PCA techniques

Reducing number of metrics

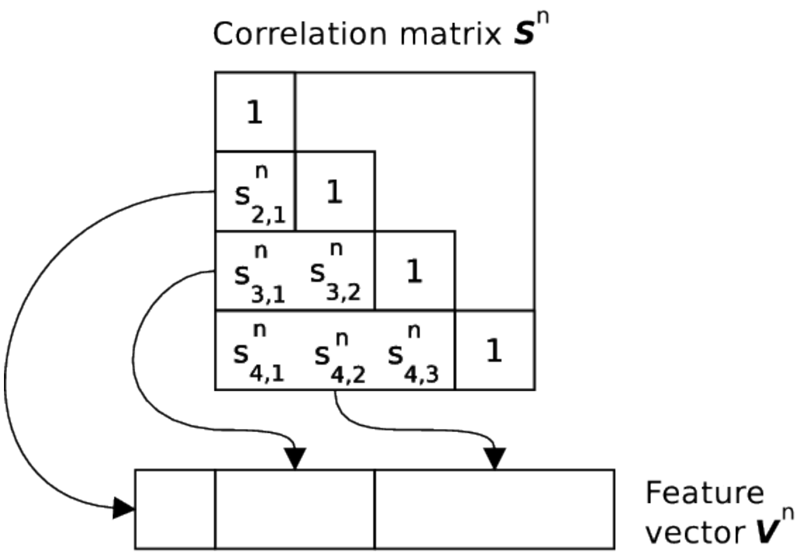


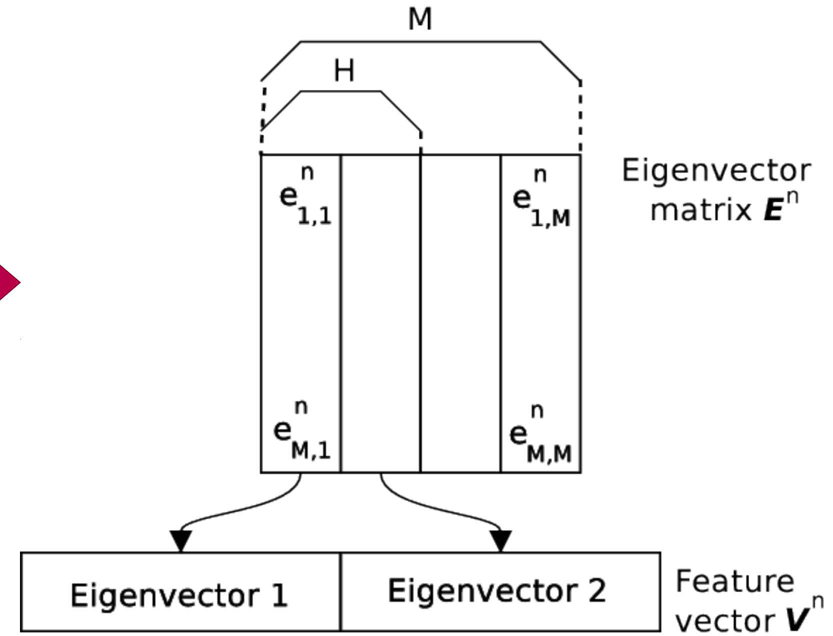
PCA-based technique



PCA-based technique

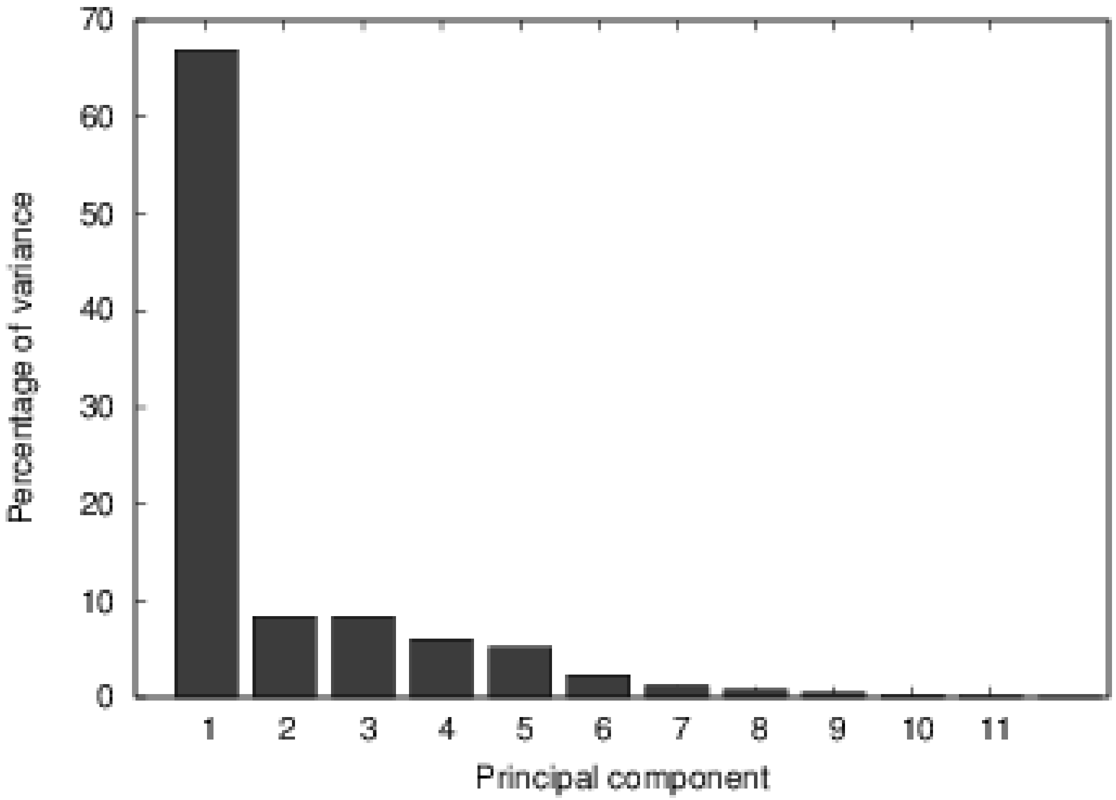
- **Building the feature vector:**



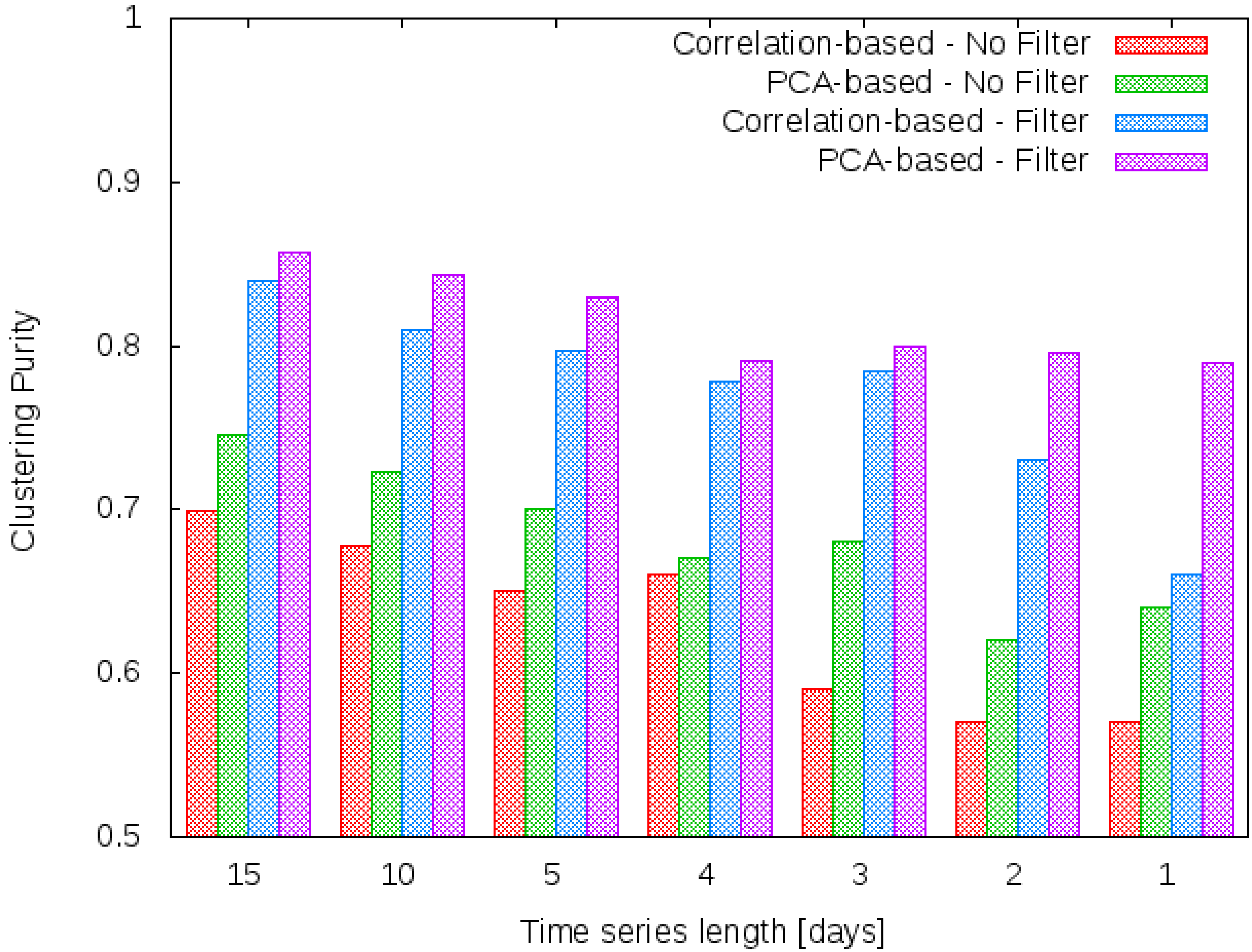


How many principal components?

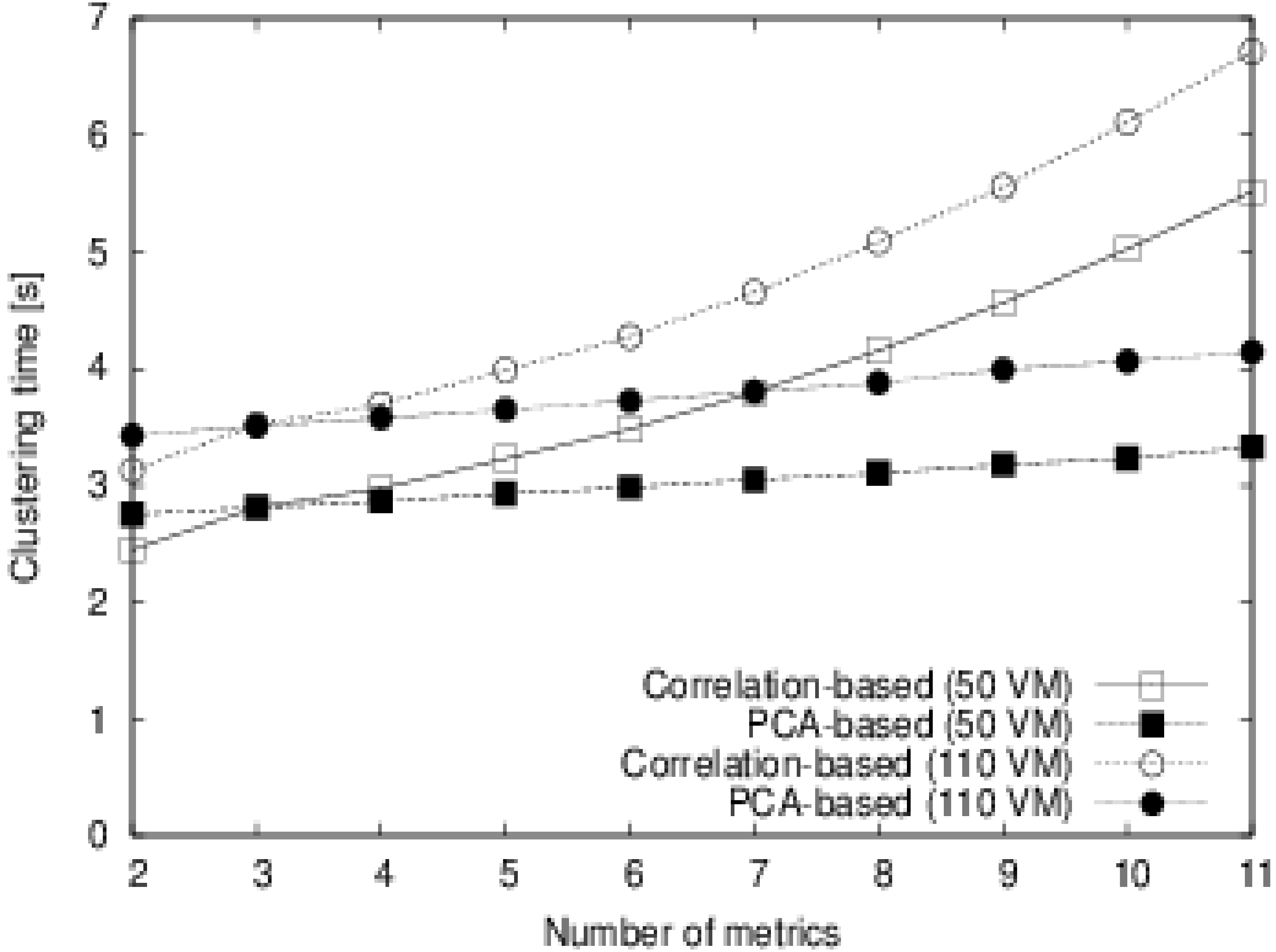
- **Use of Skree plot**
- **1 component captures ~60% of variance**
- **→ good enough for us**



Performance evaluation



Performance evaluation

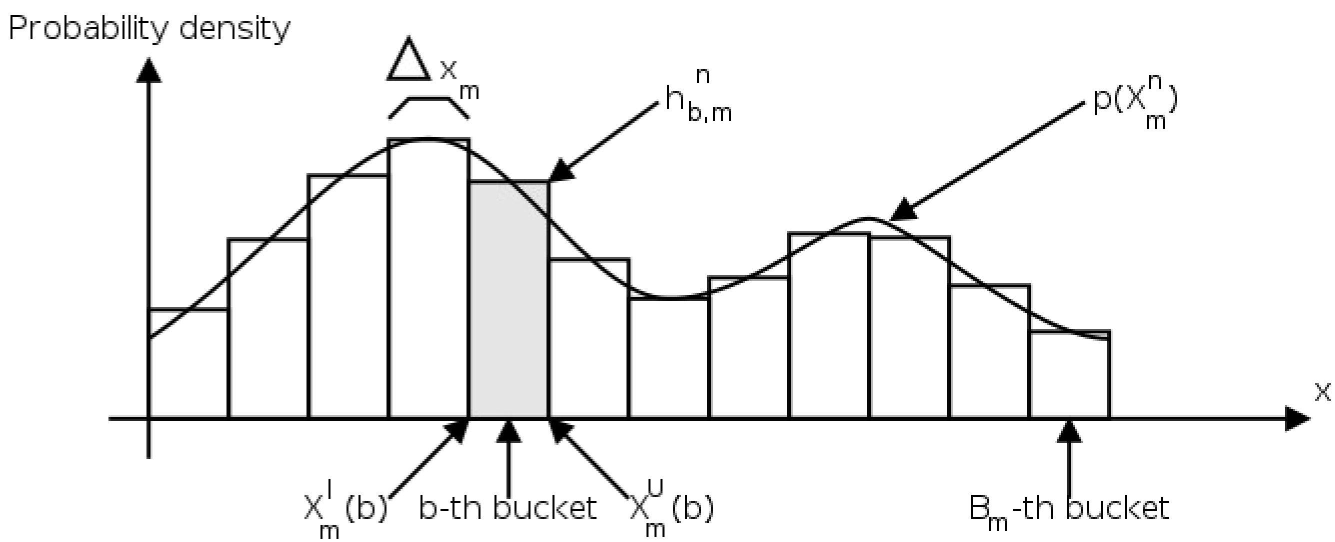


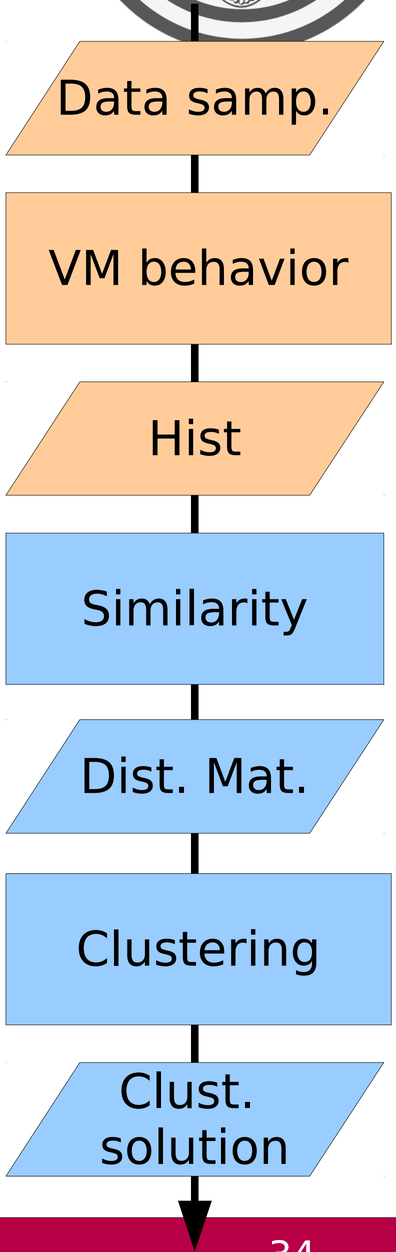
Agenda

- **Background and motivation**
  - IaaS Cloud
  - Reference scenario
  - Traditional approach vs. clustering
  - Impact on monitoring and management
- **Clustering based on metric correlation**
  - Theoretical model(s)
  - Experimental evaluation
- **Clustering based on Bhattacharyya distance**
  - Theoretical model(s)
  - Experimental evaluation
- **Conclusion and future work**

Modeling VM behavior

- **Model based on probability distribution of resource usage**
  - Multiple resources considered (metrics)
- **Histogram for every metric, every VM**
  - Normalized histogram (∑h=1)
  - B: number of buckets (critical)

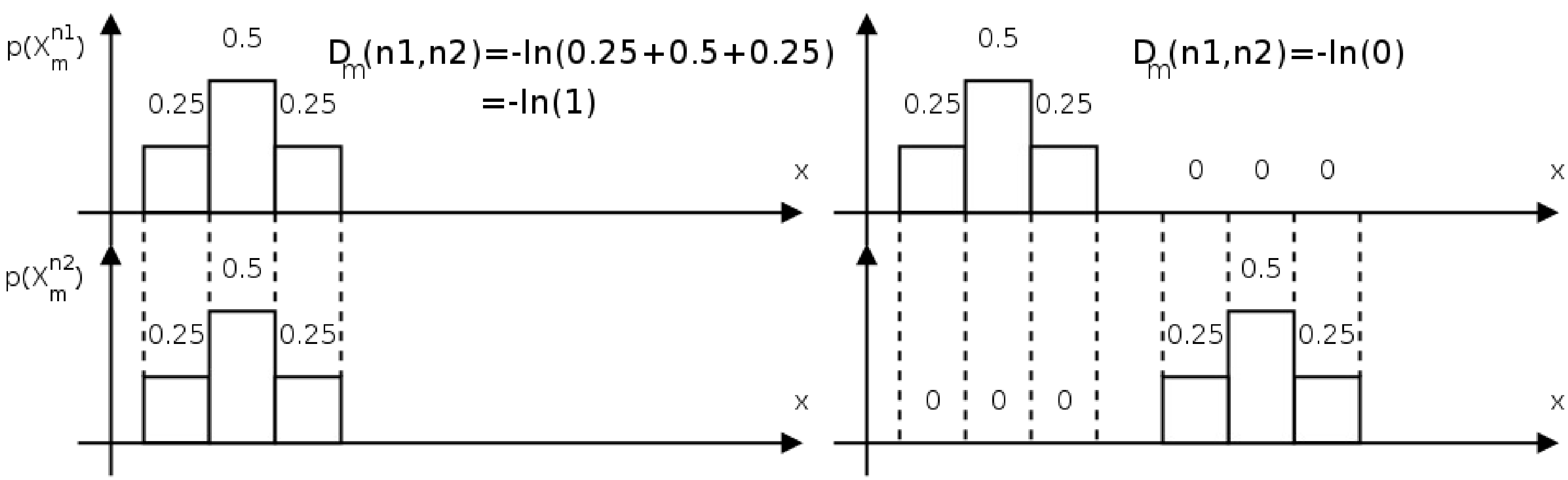


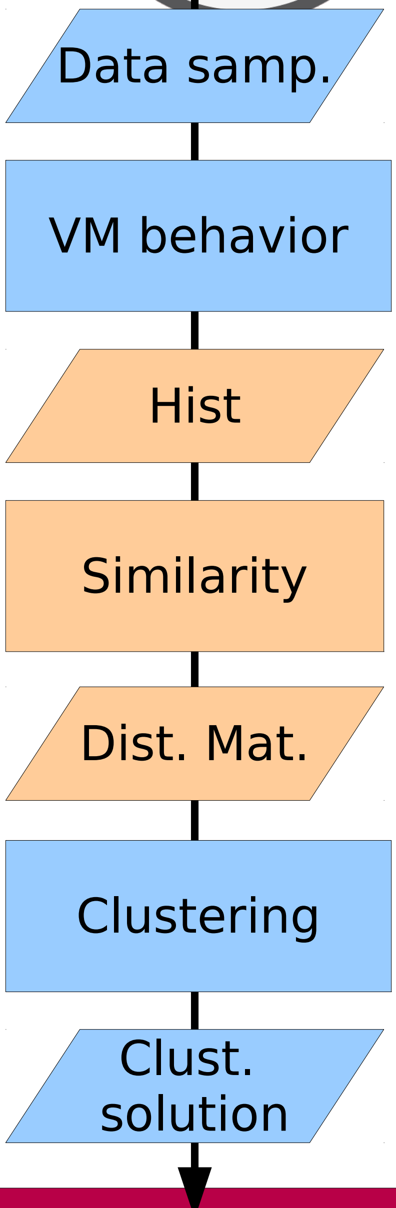


Defining VM similarity

- **Use of Bhattacharyya distance**
  - Determine distance matrix for each couple of VMs, each metric

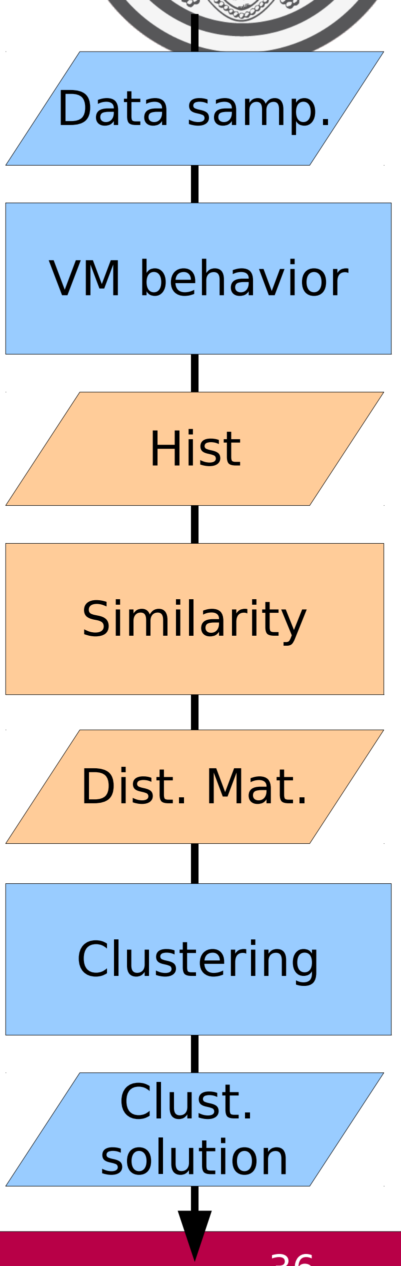
$$D_m(n1,n2) = -\ln\left(\sum_i \sqrt{h_{n1,i} \cdot h_{n2,i}}\right)$$





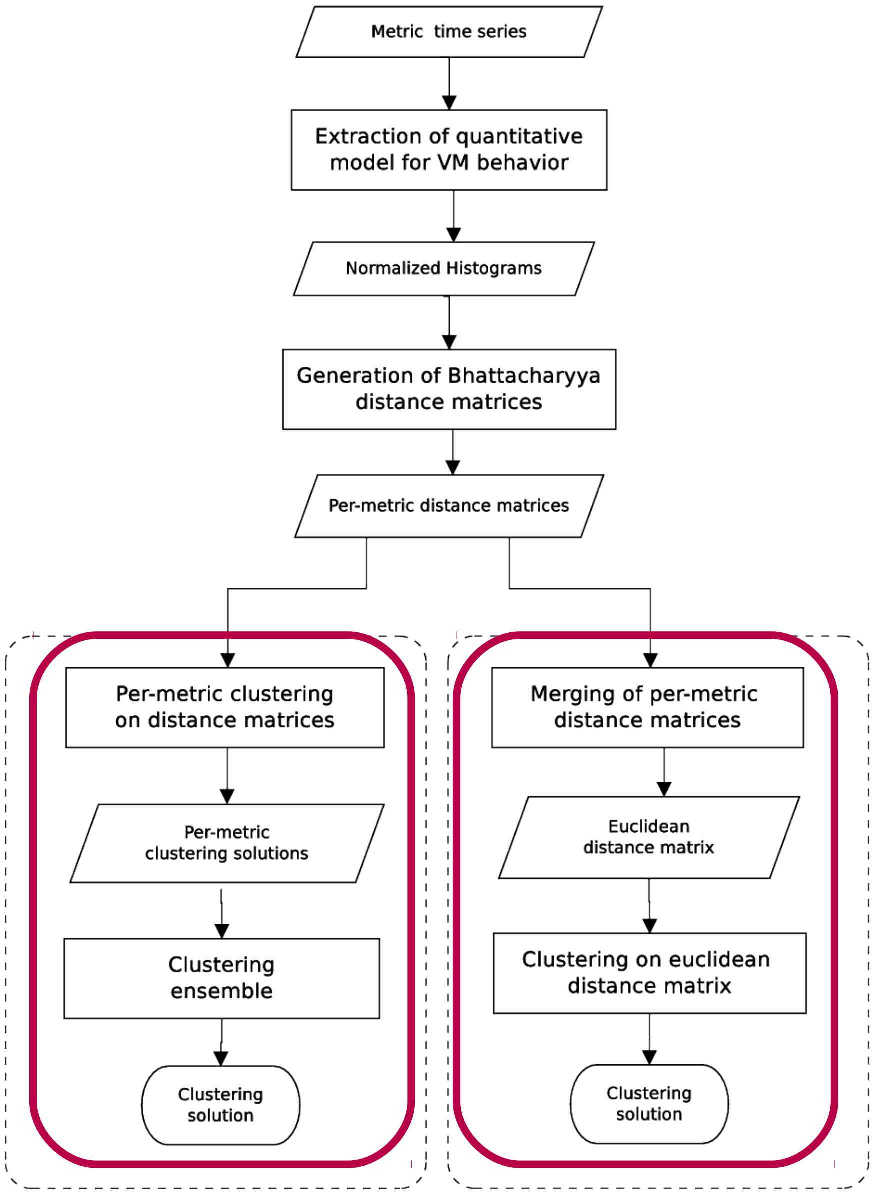
Merging multi-metric information

- **For each metric we have a different distance information**
- **How to merge the contribution of each metric?**
- **Two solutions:**
  - Euclidean distance merging
  - Solve separate clustering problems and merge clustering solutions (clustering ensemble)



Merging multi-metric information

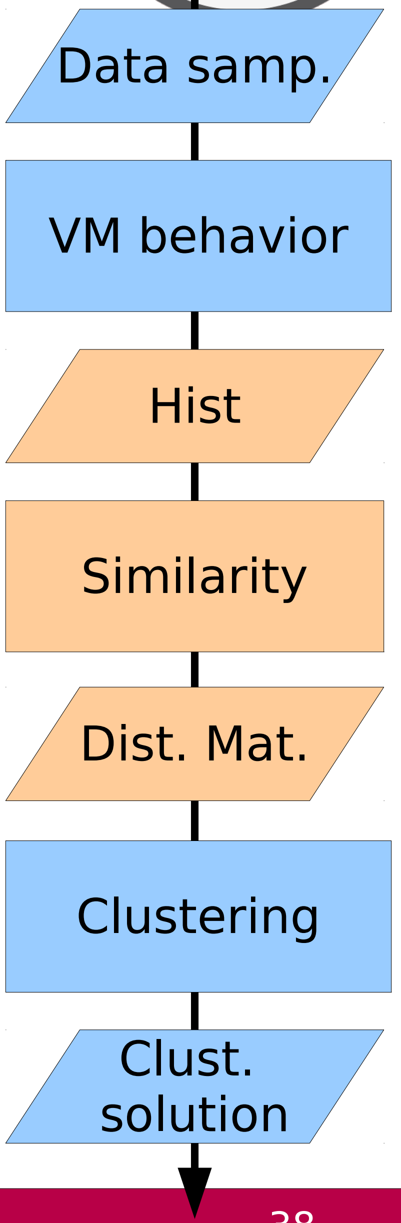
Clustering ensemble



Euclidean Distance merging

Euclidean distance merging

- **For each VMs n1, n2**
- $$D(n_1, n_2) = \sqrt{\sum D_m(n_1, n_2)^2 \cdot a_m}$$
- **Open problems:**
  - Is it correct to consider every metric together?
  - Is there a way to select the *right* metrics?

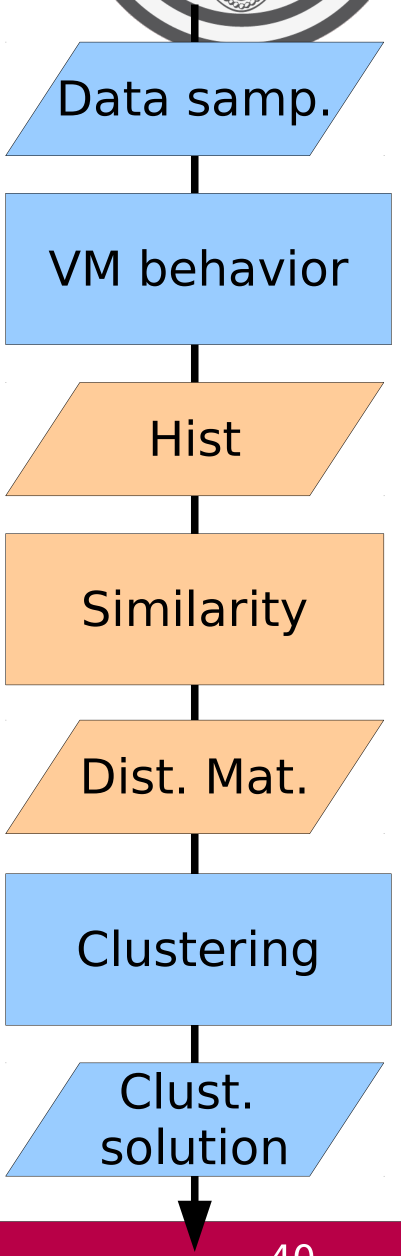


Choosing the right metrics

- **With** **euclidean merging** **multiple metrics determine the final distance matrix**
- **Not every metric provide significant information**
- **Proposal to identify relevant metrics**
  - Consider auto-correlation: ACF decreasing rapidly → random variations
  - Consider Coefficient of Variation:
    CF » 1 → spiky and noisy behavior
    CF « 1 → little information provided
- **→ Merge information from metrics with**
  - ACF decreasing slowly
  - CF ~ 1

Clustering ensemble

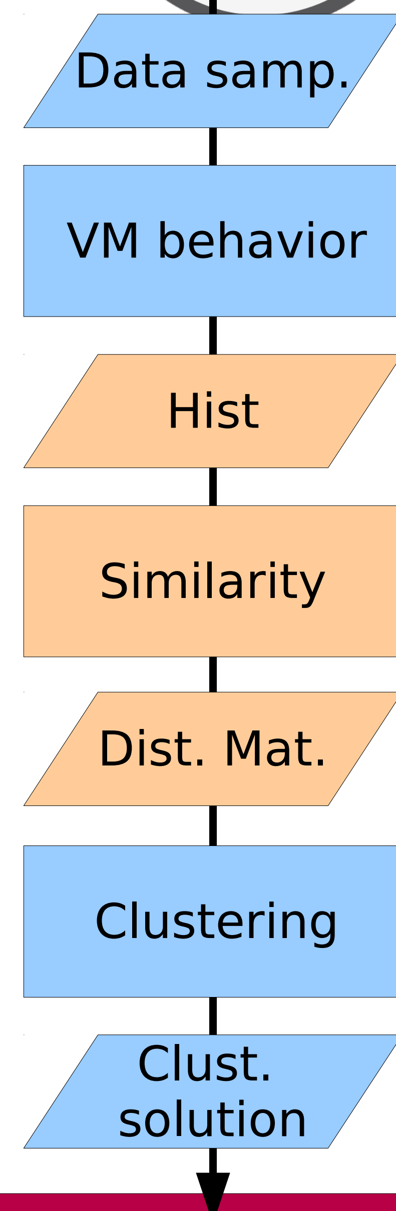
- **Two-step process**
- **For every metric m**
  - Compute Bhattacharyya distance matrix
  - Compute clustering solution
- **Compute co-occurence matrix A**
  - For each couple of VMs compute number of times they are in the same cluster
- **Clustering using matrix A as affinity**
- **OK to consider every metric?**
  - Quorum-based approach ensures good robustness of results



Clustering ensemble: example

Clustering solutions	Metric 1	Metric 2	Metric 3
VM1	CL1	CL2	CL2
VM2	CL1	CL2	CL1
VM3	CL2	CL1	CL1
VM4	CL2	CL1	CL1

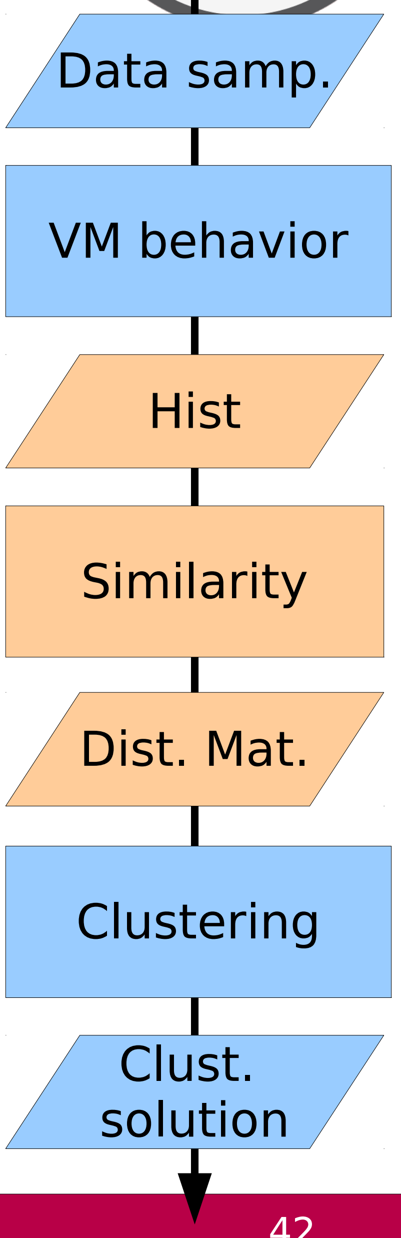
A	VM1	VM2	VM3	VM4
VM1	3	2	0	0
VM2	2	3	1	1
VM3	0	1	3	3
VM4	0	1	3	3



Clustering ensemble: example

Clustering solutions	Metric 1	Metric 2	Metric 3
VM1	CL1	CL2	CL2
VM2	CL1	CL2	CL1
VM3	CL2	CL1	CL1
VM4	CL2	CL1	CL1

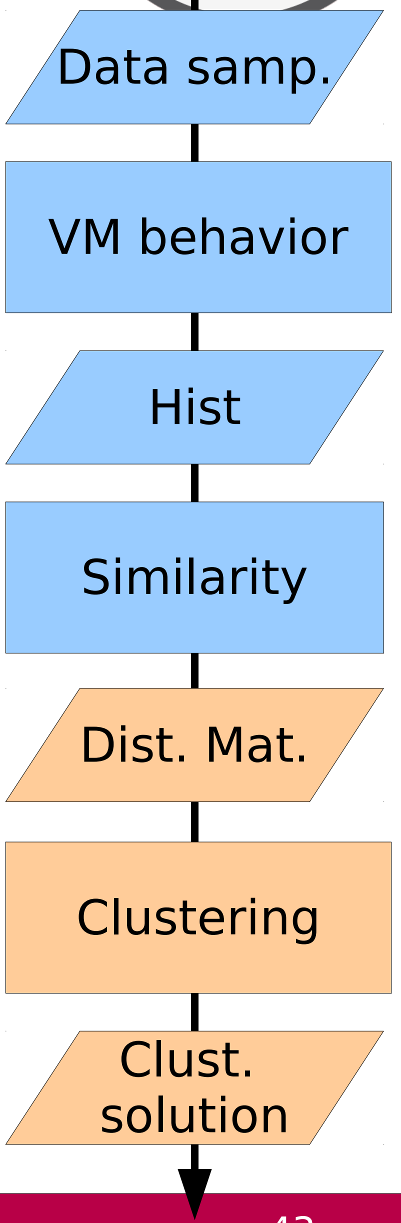
A	VM1	VM2	VM3	VM4
VM1	3	2	0	0
VM2	2	3	1	1
VM3	0	1	3	3
VM4	0	1	3	3



Clustering algorithm

- **Use of spectral clustering algorithm**
  - Input: Square, symmetric distance/affinity matrix
  - Output: Cluster ID for every VM

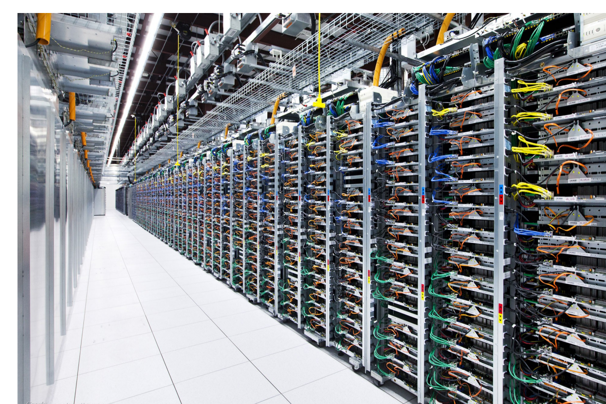
- **Additional feature:**
  - Number of clusters can be automatically determined through spectral gap analysis



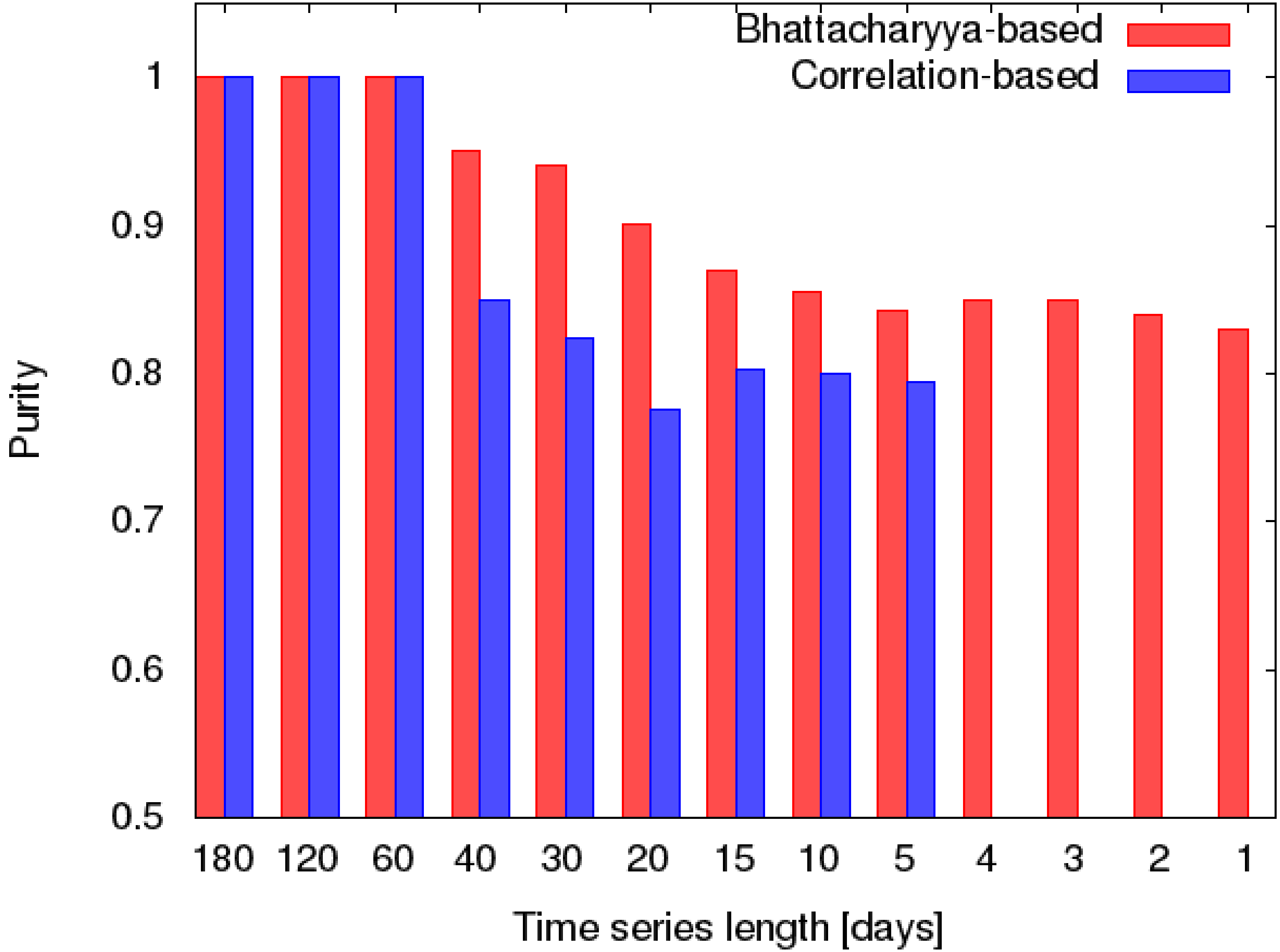
Case study

- **IaaS cloud supporting e-health**
  - Web server and DBMS
  - 110 VMs
  - 10 metrics for each VM,
  - Sampling frequency: 5 min
  - Euclidean merging of metrics

- **Goal: separate Web servers and DBMS**
  - Main metric: Purity of clustering
- **Three types of analyses**
  - Impact of time series length
  - Impact of metric selection
  - Impact of histogram charact.

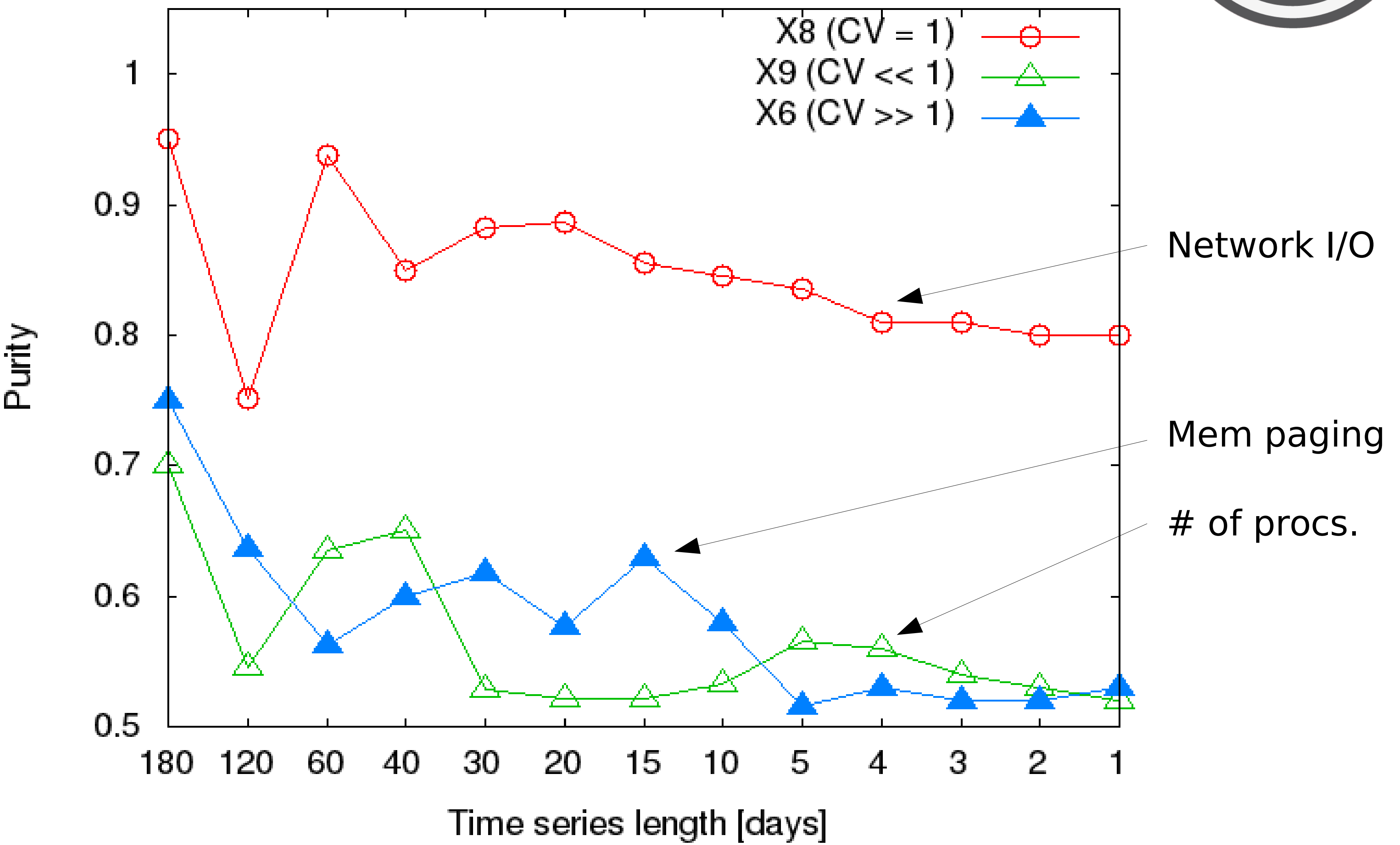


Impact of time series length

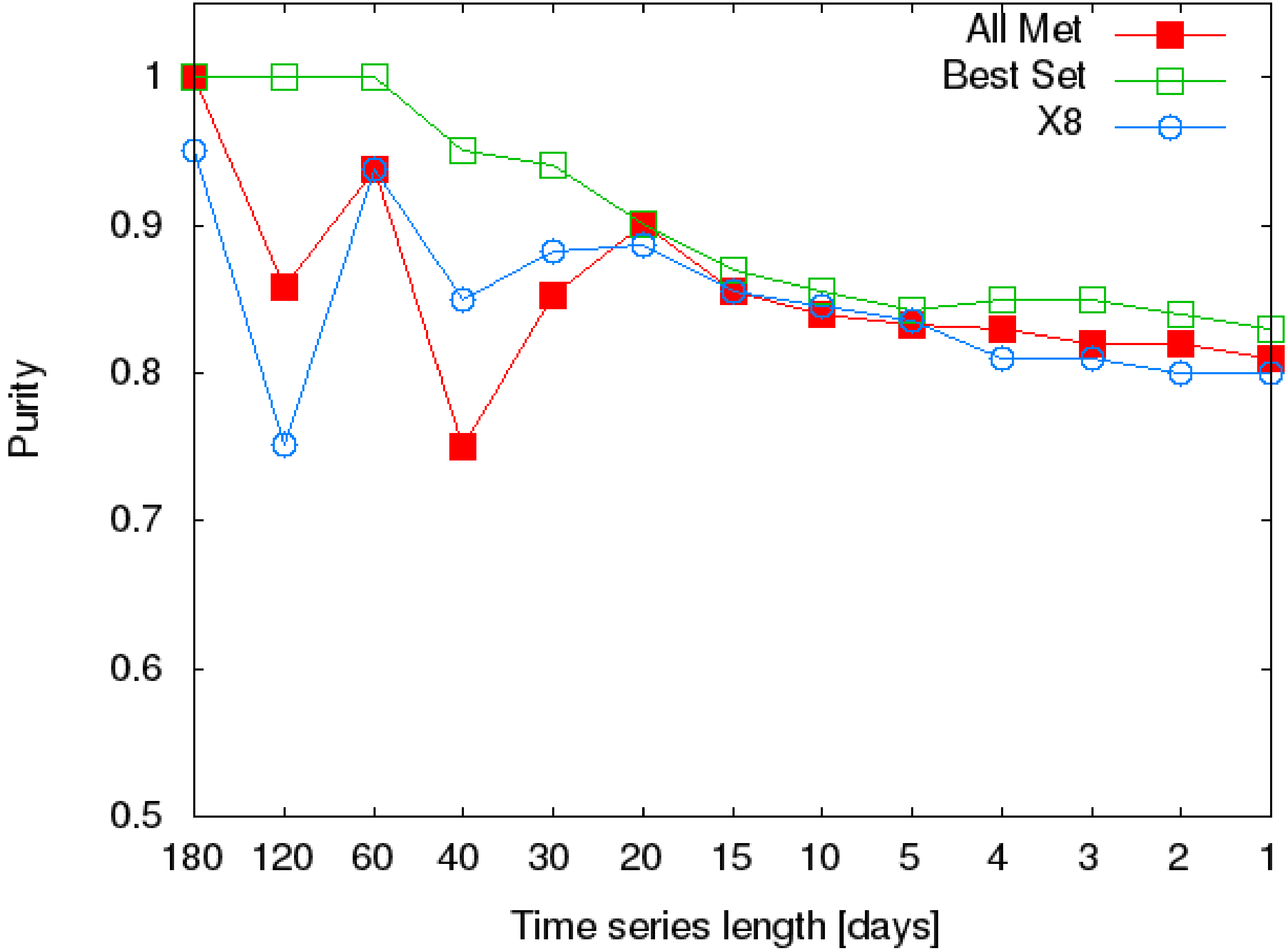


NOTE:
We consider Euclidean distance merging

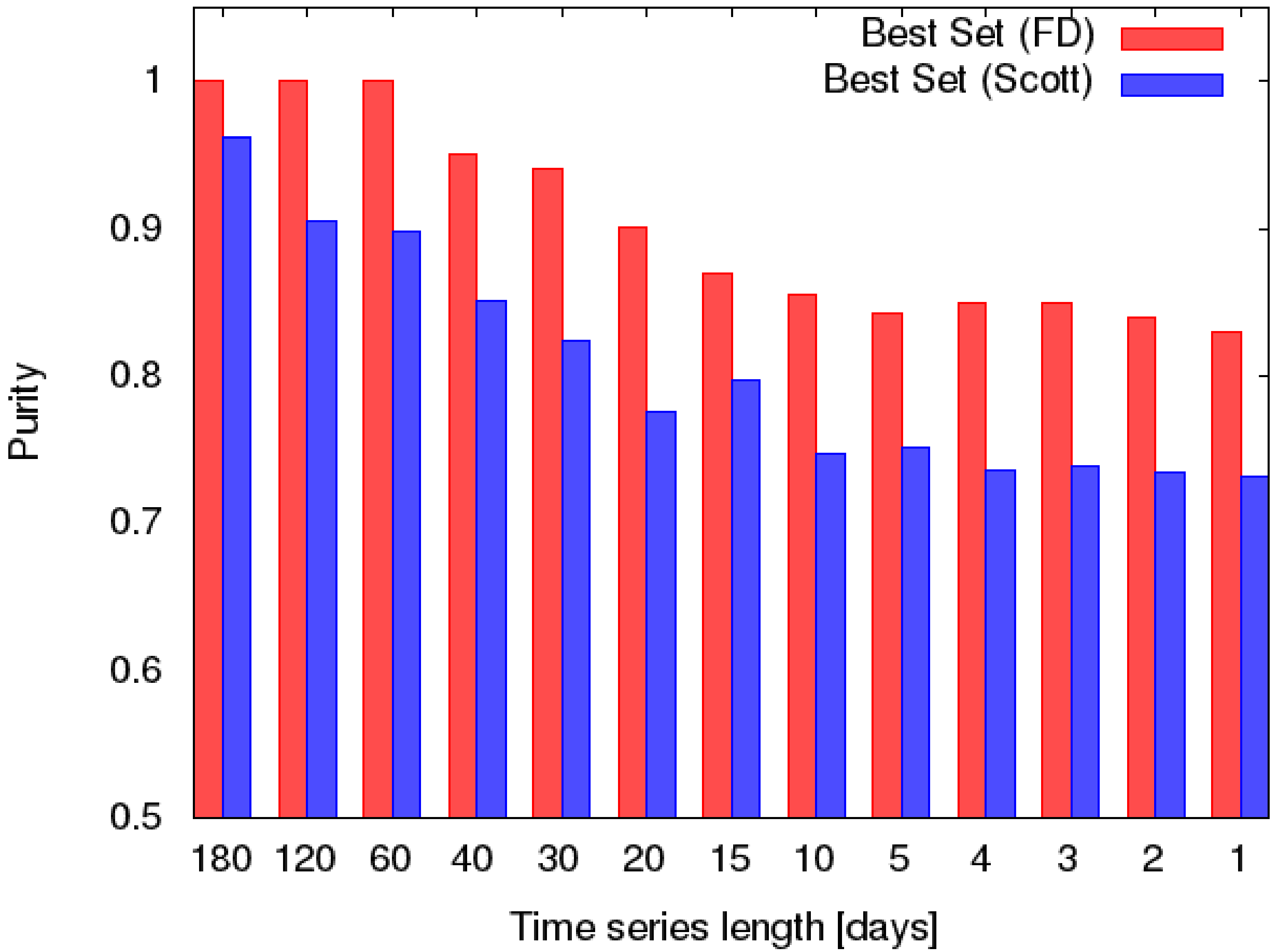
Impact of metric selection



Impact of metric selection

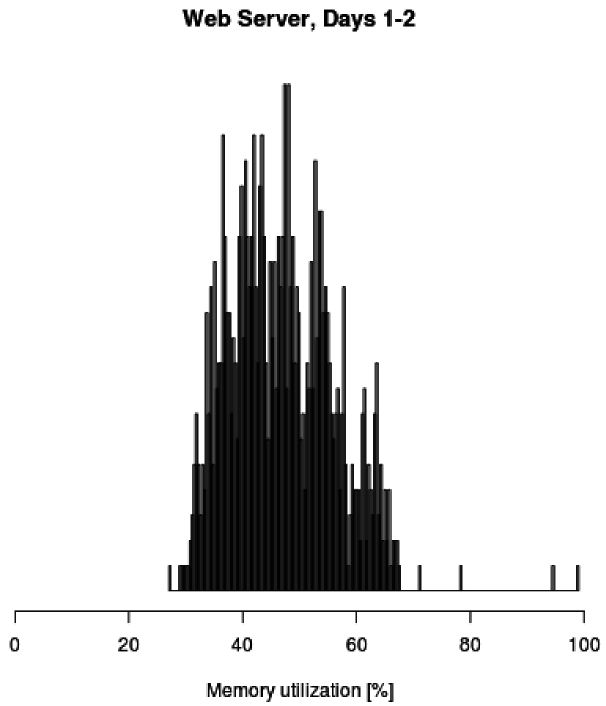


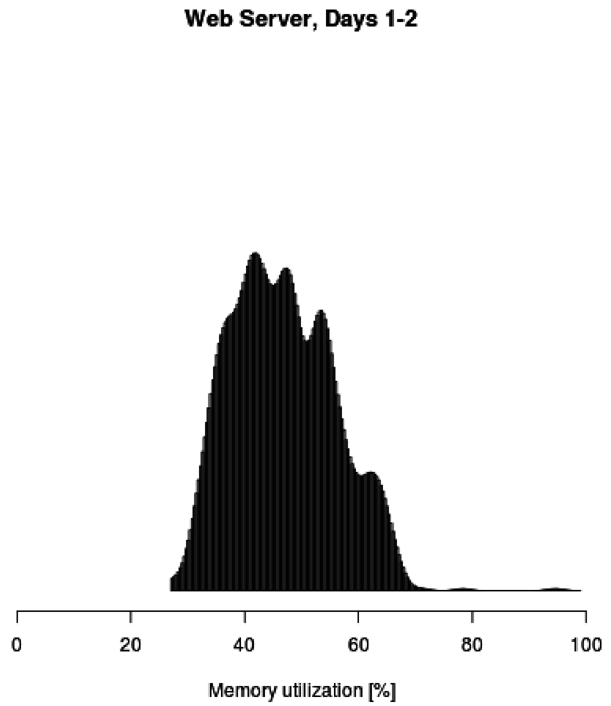
Impact of histogram characteristics



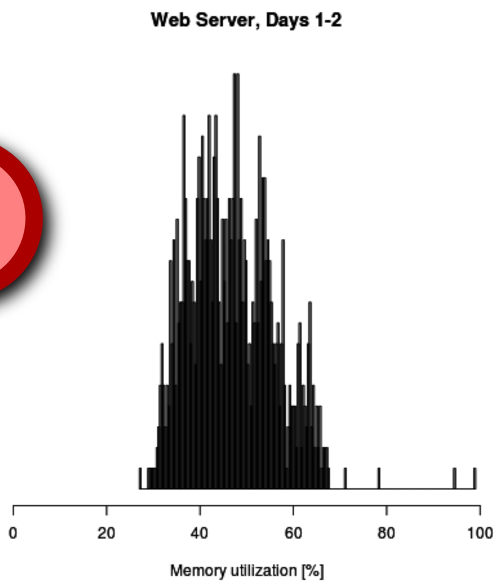
Histogram smoothing

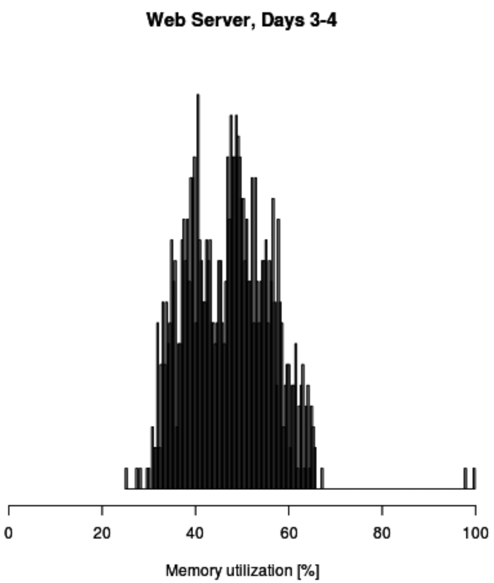
- **Bhattacharyya distance affected by quantization errors in histograms → sensitivity to histogram characteristics**
- **Proposal: gaussian smoothing of histograms before computing Bhattacharyya distance**



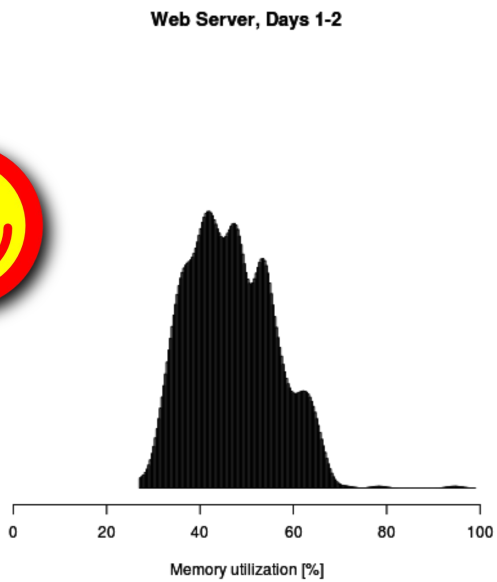


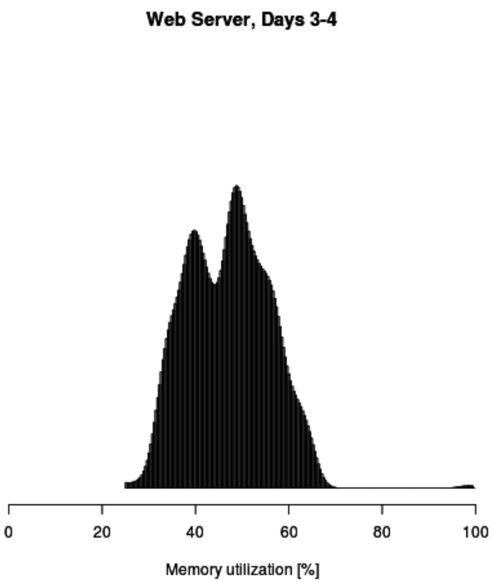
Histogram smoothing





No smoothing:
Bhattacharyya distance
0.496

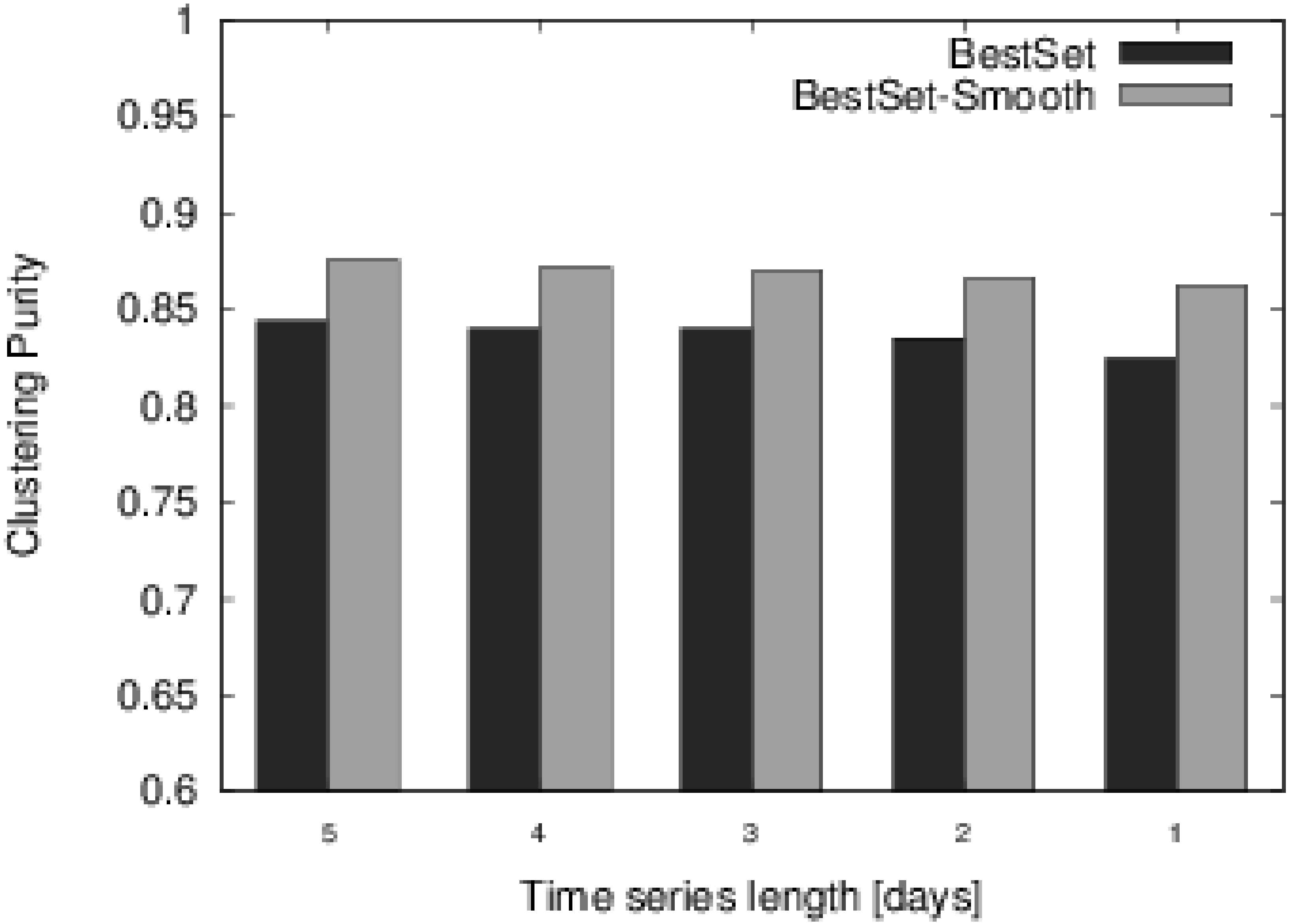




Smoothing:
Bhattacharyya distance
0.297

Reduction by 40%

Effect of histogram smoothing

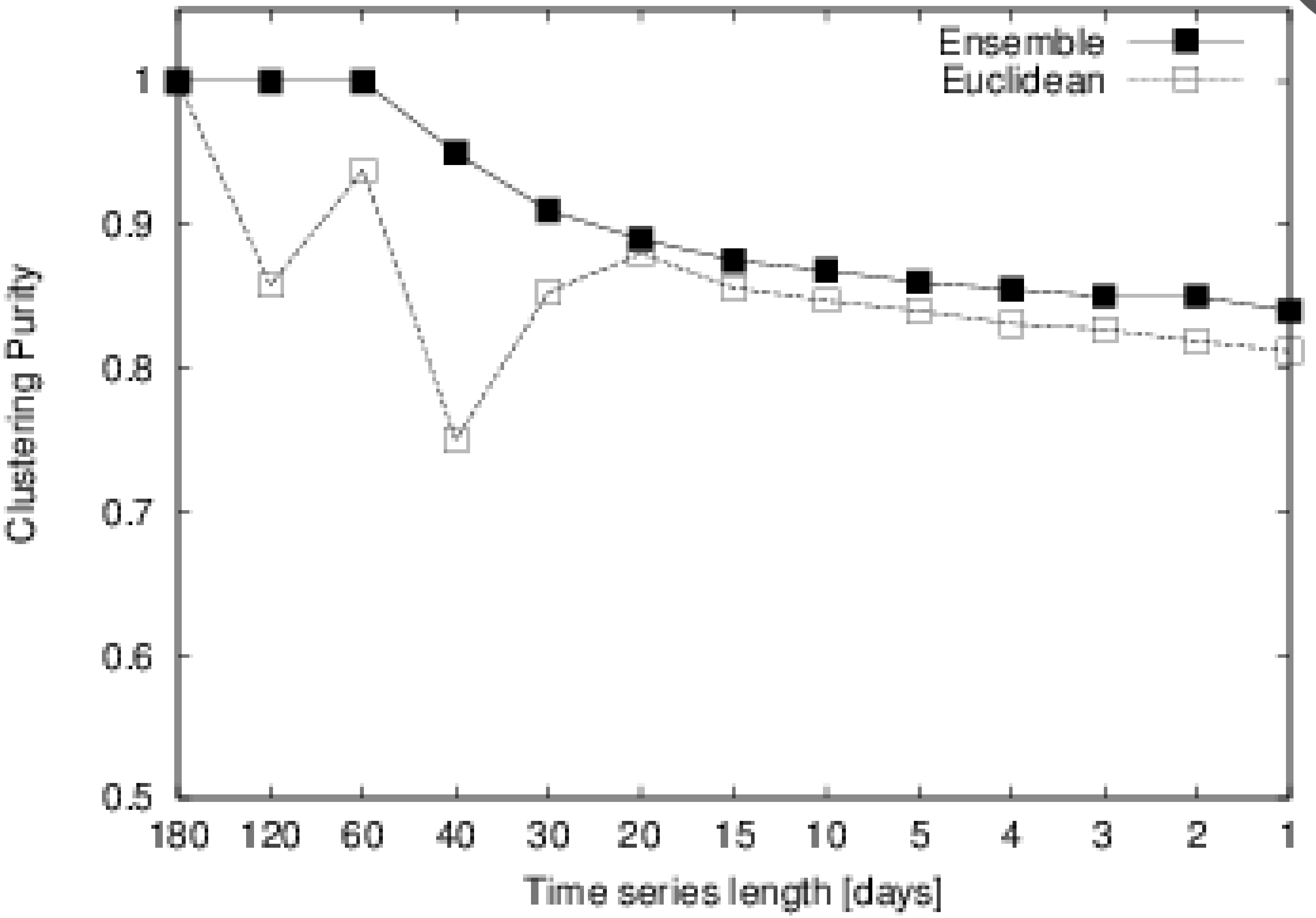


Clustering Ensemble

- **Overall goal:**
  - Reduce sensitivity to histogram characteristics (number of histogram bkts)
  - No need to select significant metrics
  - No smoothing required

- **Potential drawback**
  - Higher computational cost

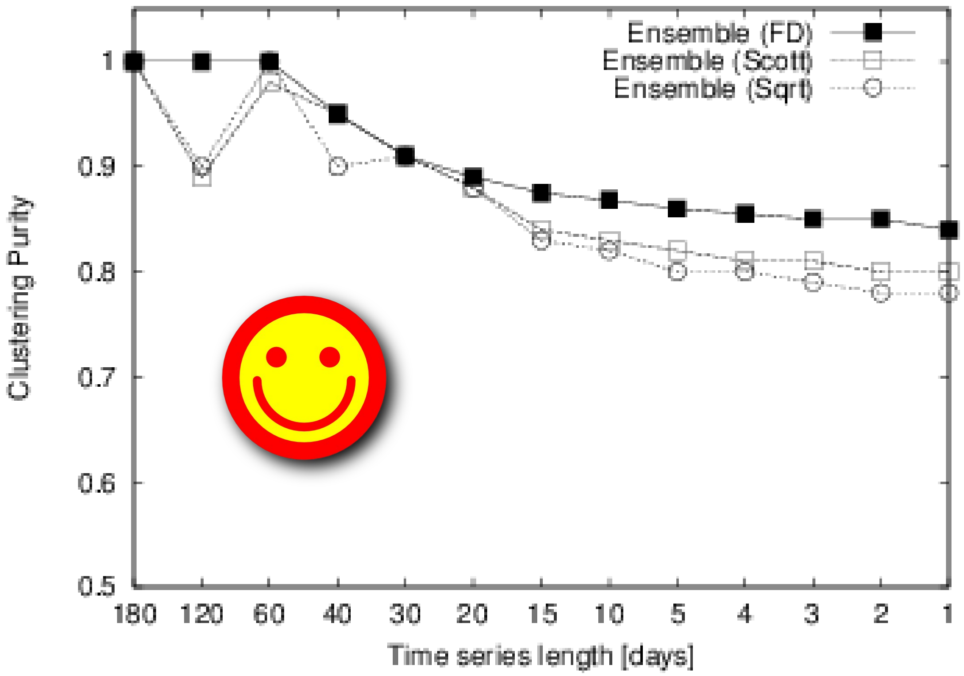
Clustering Ensemble



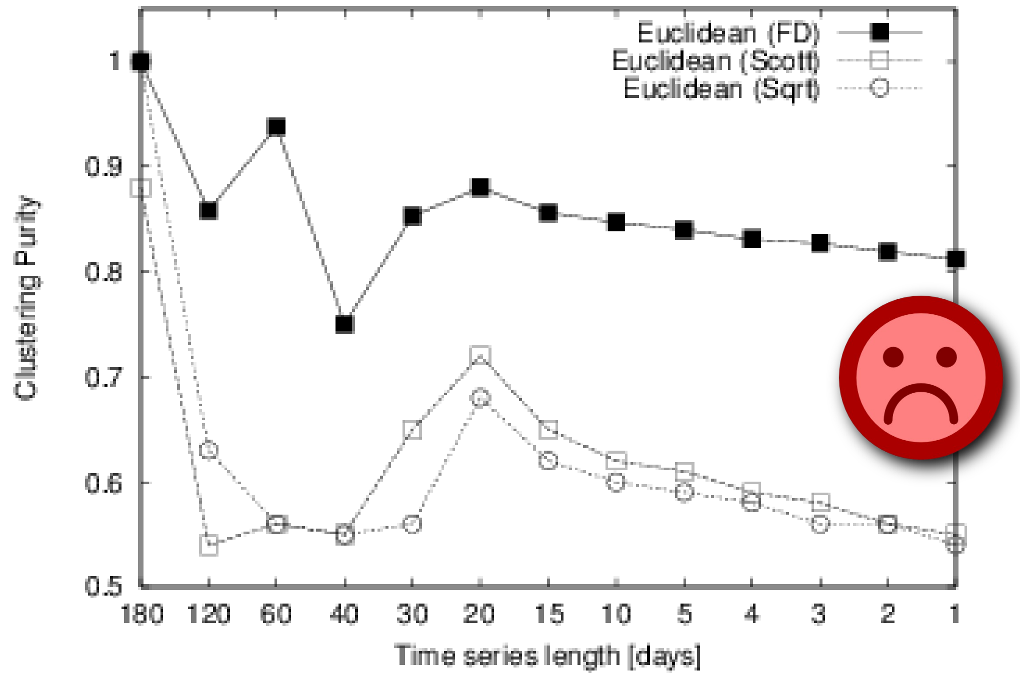
Clustering Ensemble

- **Major stability improvement**
- **Almost insensitive to histogram number of buckets**

Clustering ensemble

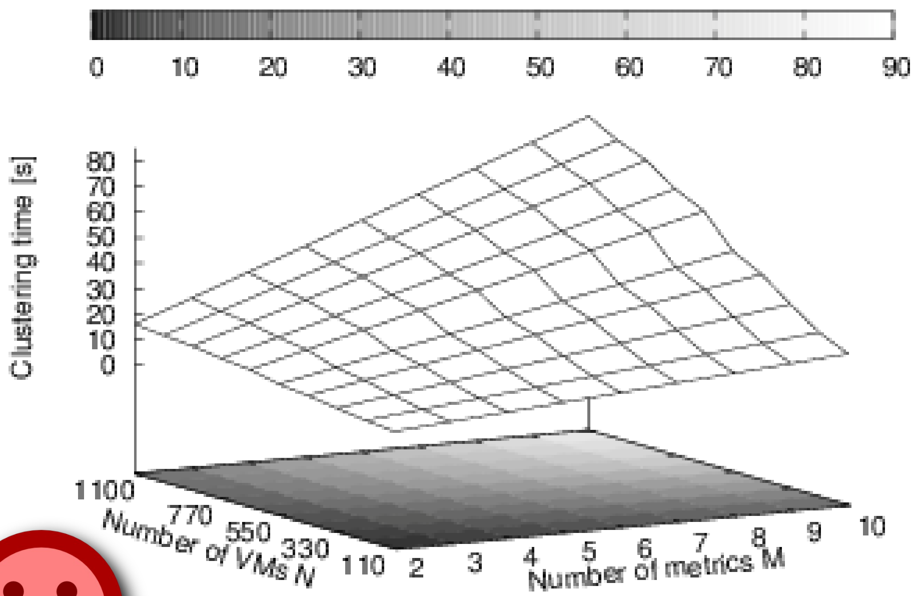


Euclidean merging

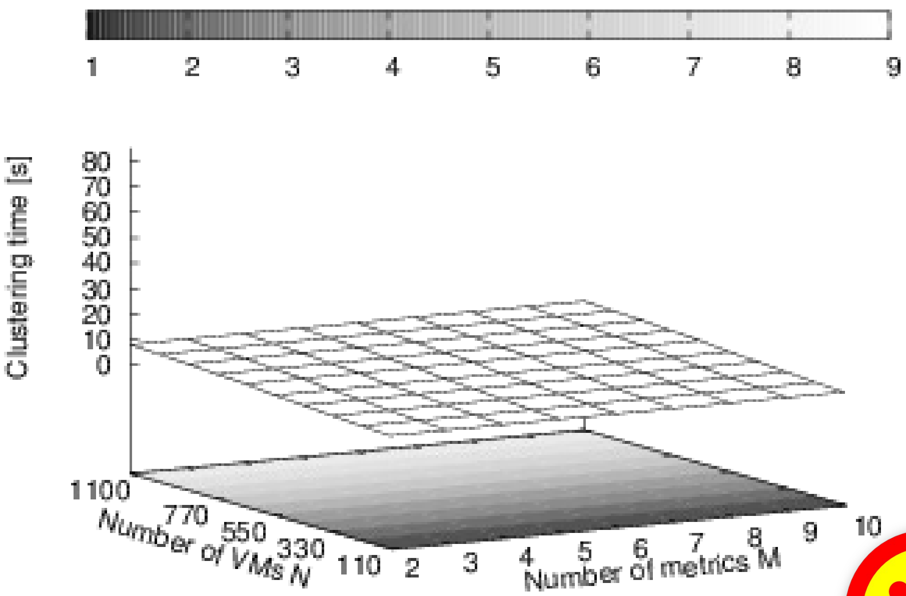


Clustering Ensemble

- **Significant performance penalty:**
- **1 clustering for each metric**
- **Typically uses more metric than euclidean merging**



Clustering ensemble



Euclidean merging

Agenda

- **Background and motivation**
  - IaaS Cloud
  - Reference scenario
  - Traditional approach vs. clustering
  - Impact on monitoring and management
- **Clustering based on metric correlation**
  - Theoretical model(s)
  - Experimental evaluation
- **Clustering based on Bhattacharyya distance**
  - Theoretical model(s)
  - Experimental evaluation
- **Conclusion and future work**

Conclusion and future work

- **Scalability in IaaS cloud systems**
  → open issue
- **Proposal an analysis of mutiple methodologies to improve scalability through clustering of similar VMs**
    - **Representing VM behavior using correlation**
    - Reduction of correlation data with PCA
    - **Representing VM behavior with histograms**
    - Euclidean merging of distances
    - Metric selection
    - Histogram smoothing
    - Clustering ensemble

Conclusion and future work

- **Experimental results are encouraging**
  - Can achieve high clustering purity
  - Can provide accurate clustering even with very short time series
  - Can provide stable results
  - Time for clustering is acceptable

- **This is not a crystal ball**
  - But may be a useful tool to improve monitoring and management of cloud data centers

Conclusion and future work

- **Future research directions:**

- **Evaluate different models for VM behavior**

- **Application of clustering to improve scalability of data center management**

References

- Claudia Canali, Riccardo Lancellotti, **"Automatic Virtual Machine Clustering based on Bhattacharyya Distance for Multi-Cloud Systems"**, Proc. of 1st International Workshop on Multi-cloud Applications and Federated Clouds (Multi-Cloud'13), Prague, April 2013
- Claudia Canali, Riccardo Lancellotti, **"Automated Clustering of Virtual Machines based on Correlation of Resource Usage"**, Journal of Communications Software and Systems (JCOMSS), Vol. 8, No. 4, Dec. 2012
- Claudia Canali, Riccardo Lancellotti, **"Automated Clustering of VMs for Scalable Cloud Monitoring and Management"**, 20th International Conference on Software, Telecommunications and Computer Networks (SOFTCOM'12), Split, Croatia, 11-13 Sept. 2012
- Claudia Canali, Riccardo Lancellotti, **"Exploiting Ensemble Techniques for Automatic Clustering of Virtual Machine Clustering in Cloud Systems"**, to appear on Automated Software Engineering
- Claudia Canali, Riccardo Lancellotti, **"Improving scalability of cloud monitoring through PCA-based Clustering of Virtual Machines"**, to appear on Journal of Computer Science and Technology

Automatic clustering of similar VM to improve the scalability of monitoring and management in IaaS cloud infrastructures

C. Canali
R. Lancellotti

University of Modena and Reggio Emilia
Department of Engineering “Enzo Ferrari”