Automatic Virtual Machine Clustering based on Bhattacharyya Distance for Multi-Cloud Systems

C. Canali
R. Lancellotti

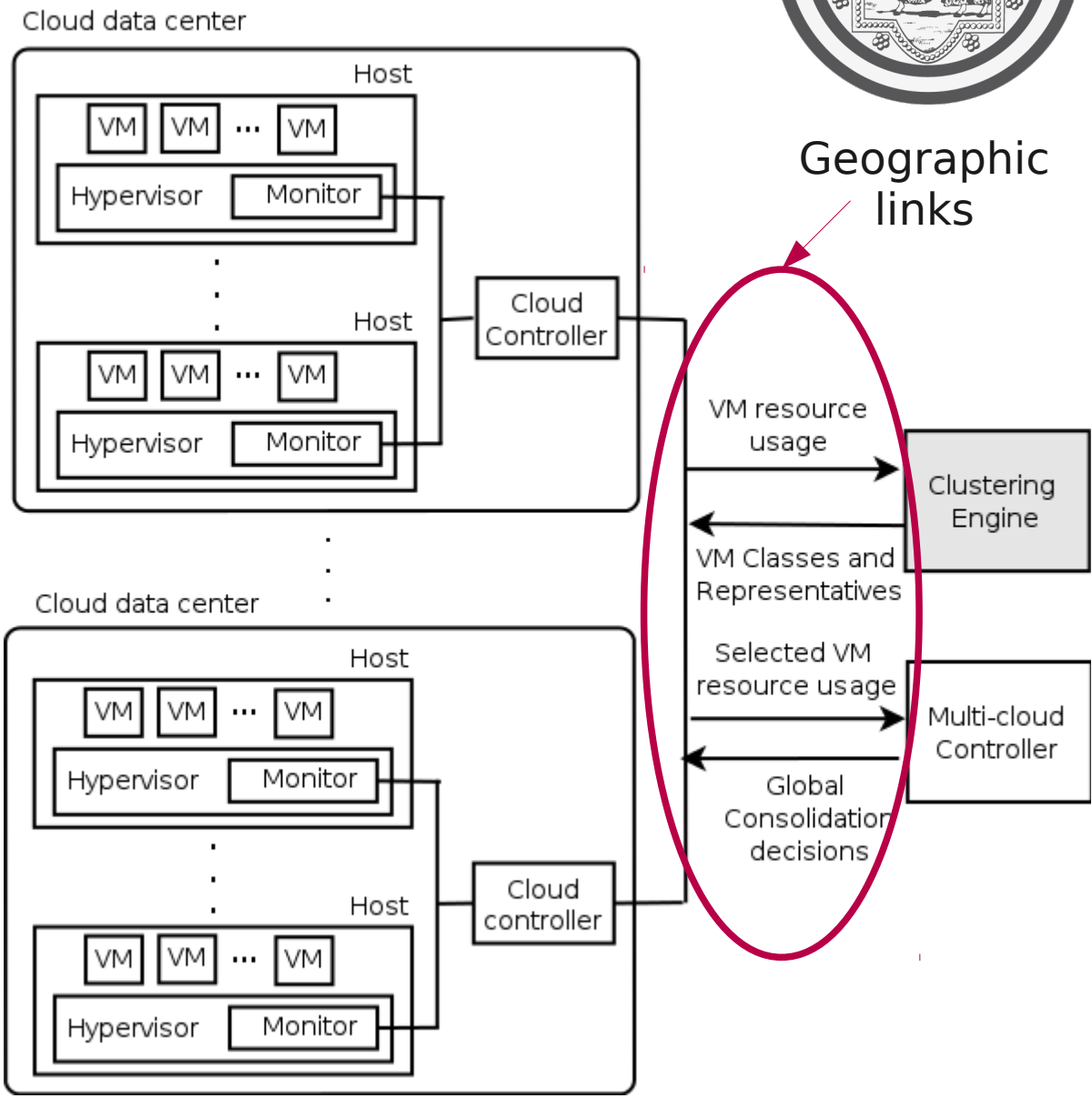
University of Modena and Reggio Emilia

Cloud computing challenges

- **Large data centers (> 10^5 VMs)**
  → huge amount of data
- **Multiple data centers**
  → geographic data exchange
- **→ Scalability problems**
- **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

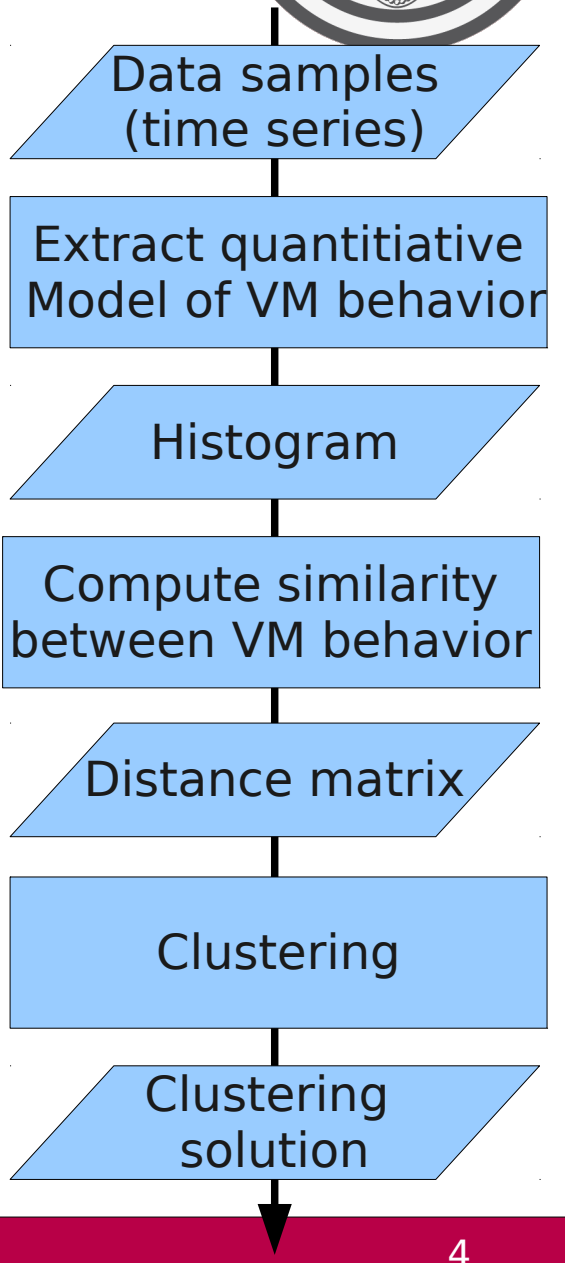
Reference scenario

- **IaaS with long term commitment**
- **Reactive VM relocation**
  - Local scope
  - Overload mgm
- **Periodic global consolidation**
  - Global scope
  - Server mgm



Impact on monitoring scalability

- **Methodology:**
  - Define quantitative model for VM behavior
  - Define VM similarity (dist. matrix)
  - 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



Impact on monitoring scalability

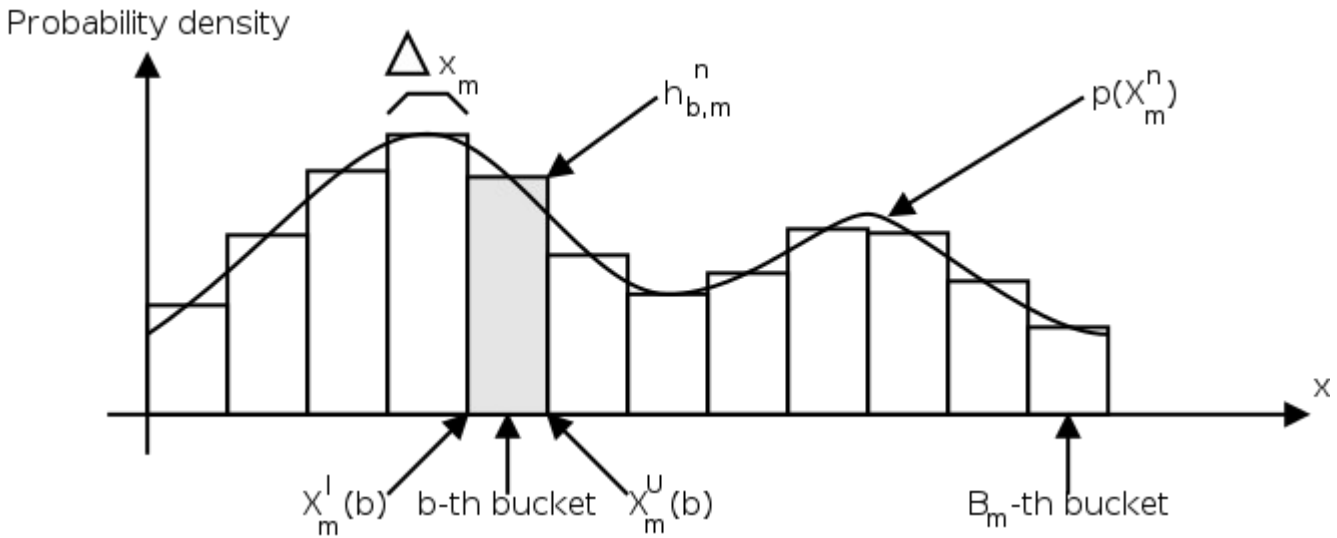
- **Case study:**
  - E-health, Web-based application
  - Deployed on cloud IaaS

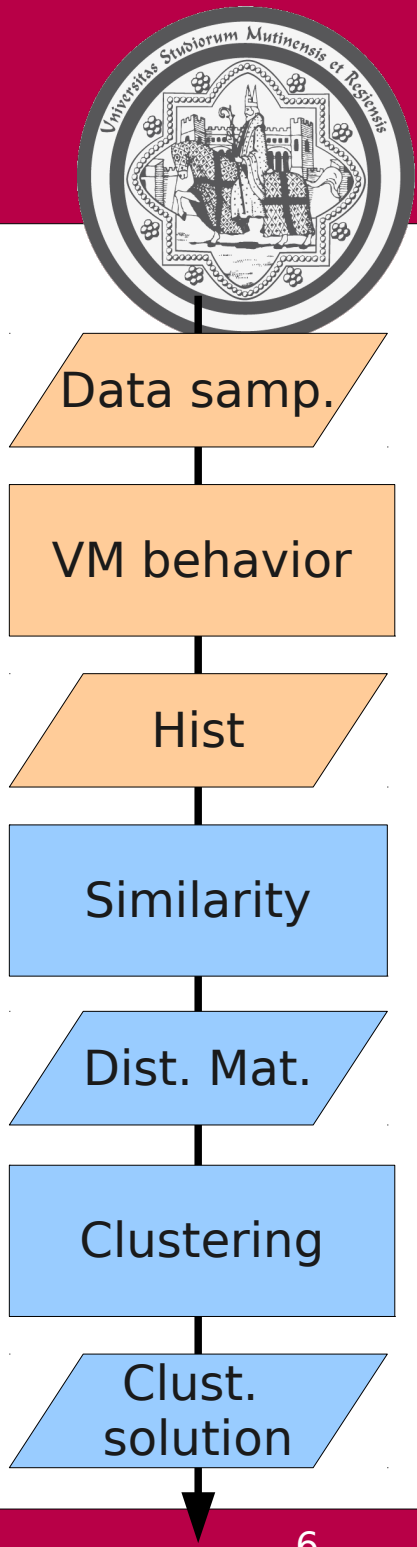
- **Numeric example:**
  - 110 VMs, K metrics, sampling frequency: 5 min.
    → ~3.2 10^4 K samples/day
  - 2 classes, 3 rep. per class
    → ~2.1 10^3 K samples/day

- **→ Monitoring data reduced by 1 order of magnitude**

Modeling VM behavior

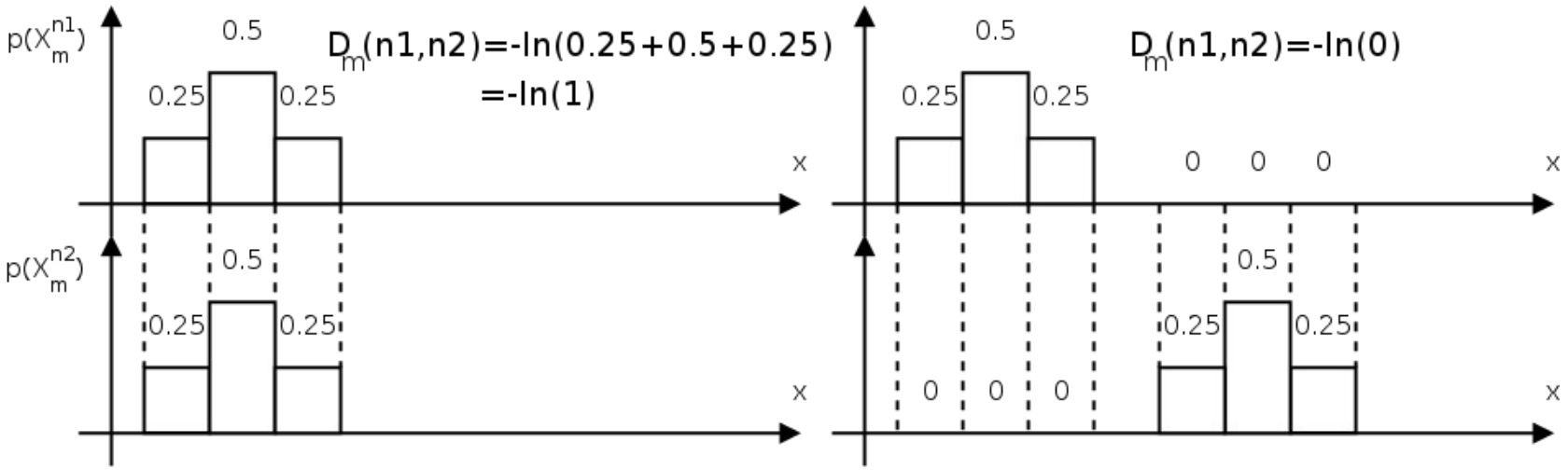
- **Model based on probability distribution of resource usage**
  - Multiple resources considered (metrics)
- **Histogram for every metric, every VM**
  - Normalized histogram (∑h=1)
  - B: number of buckets (critical)

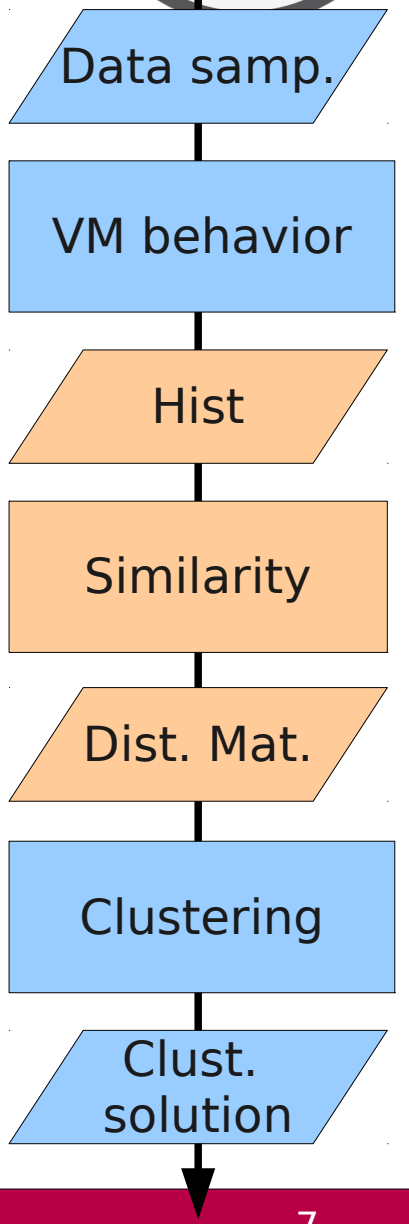




Defining VM similarity

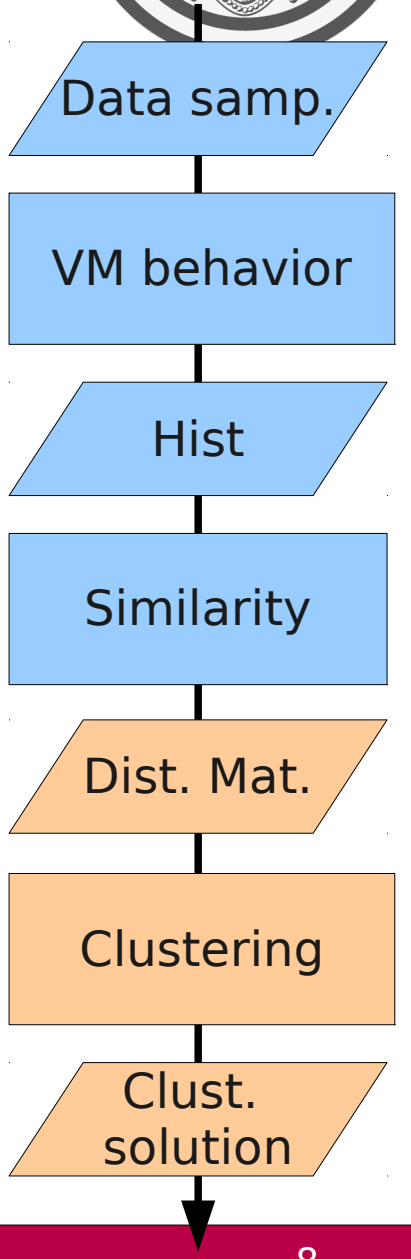
- **Use of Bhattacharyya distance**
  - Determine distance matrix for each couple of VMs, each metric
- **Euclidean combination of distance matrices**
  - Sum of squares of multiple distances





Clustering algorithm

- **Use of spectral clustering algorithm**
  - Input: Square, symmetric distance matrix
  - Output: Cluster ID for every VM
- **Additional feature:**
  - Number of clusters can be automatically determined through spectral gap analysis
- **Open problems:**
  - Is it correct to consider every metric together?
  - Is there a way to select the *right* metrics?



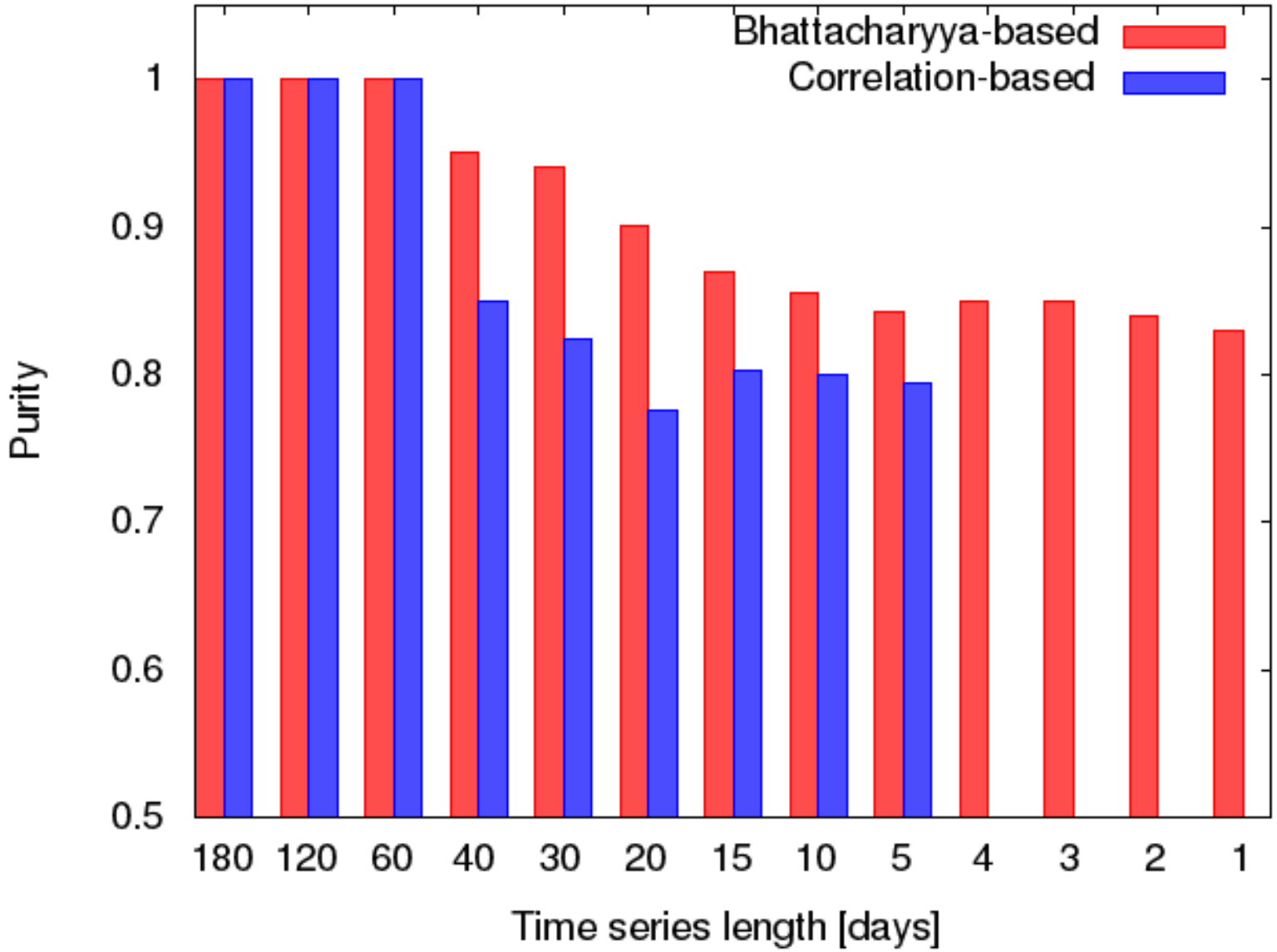
Choosing the right metrics

- **Multiple metrics are merged into 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

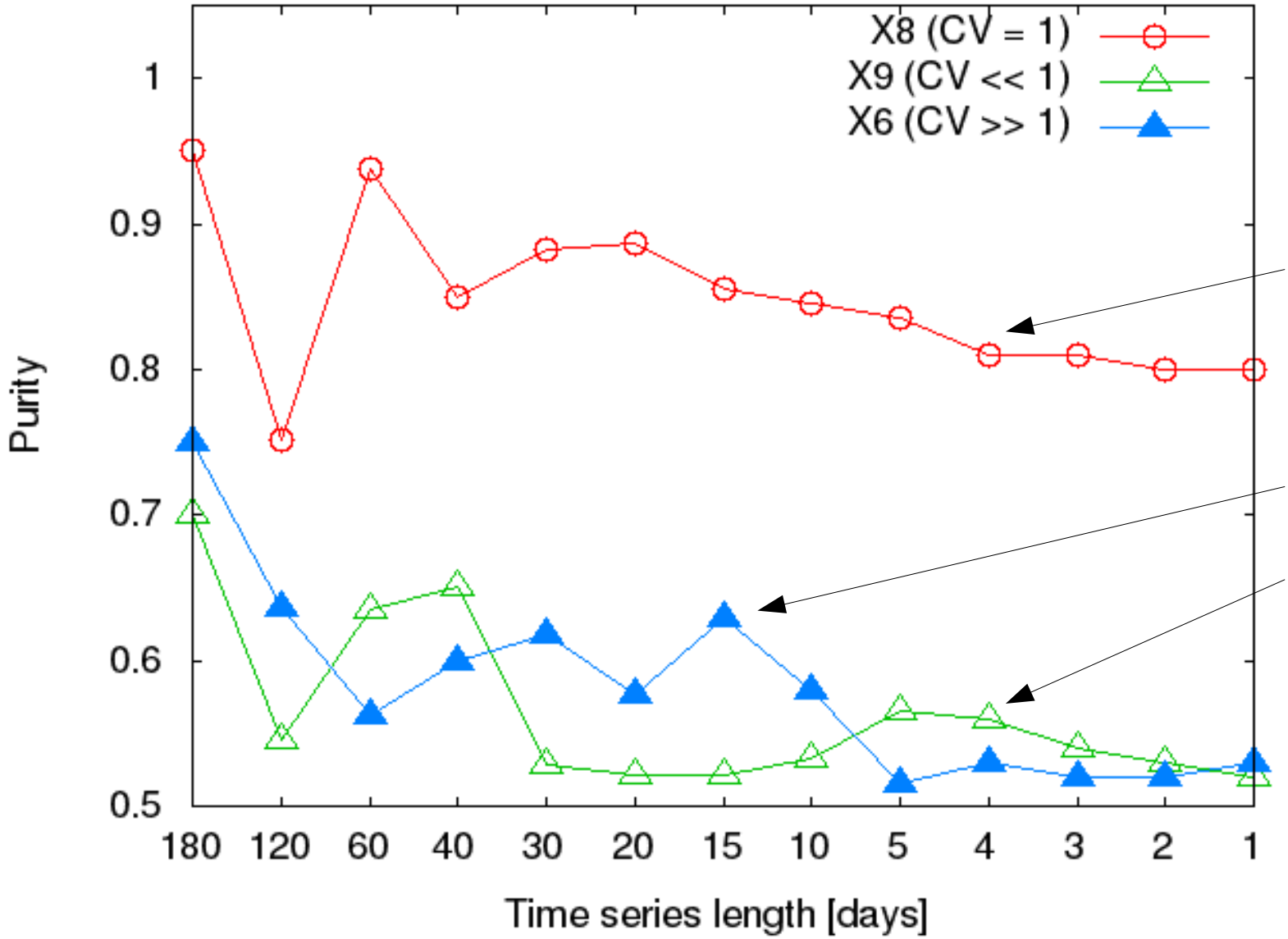
Case study

- **IaaS cloud supporting e-health**
  - Web server and DBMS
  - 110 VMs
  - 10 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 metric selection techniques
  - Impact of histogram characteristics

Impact of time series length



Impact of metric selection (1)

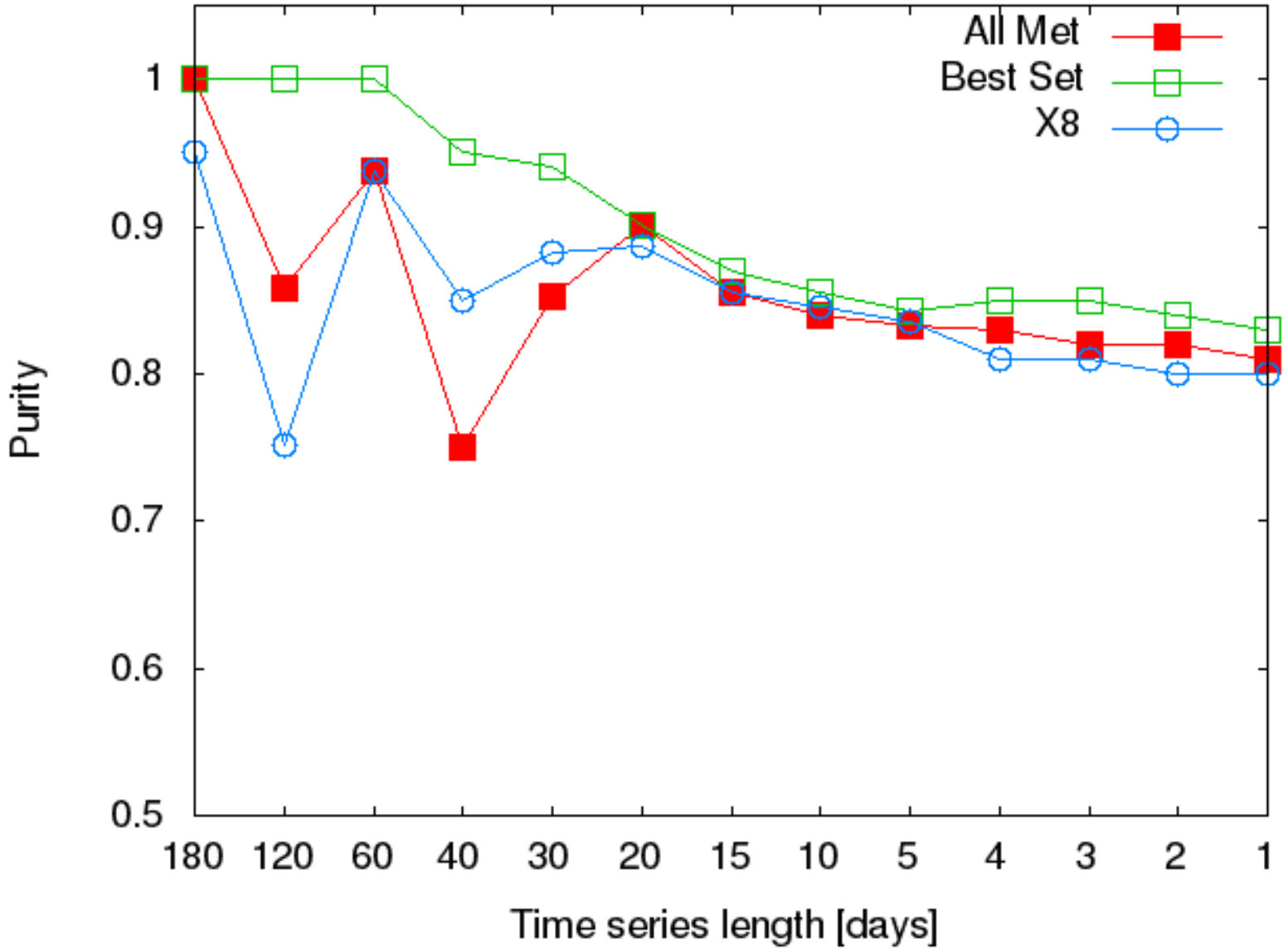


Network I/O

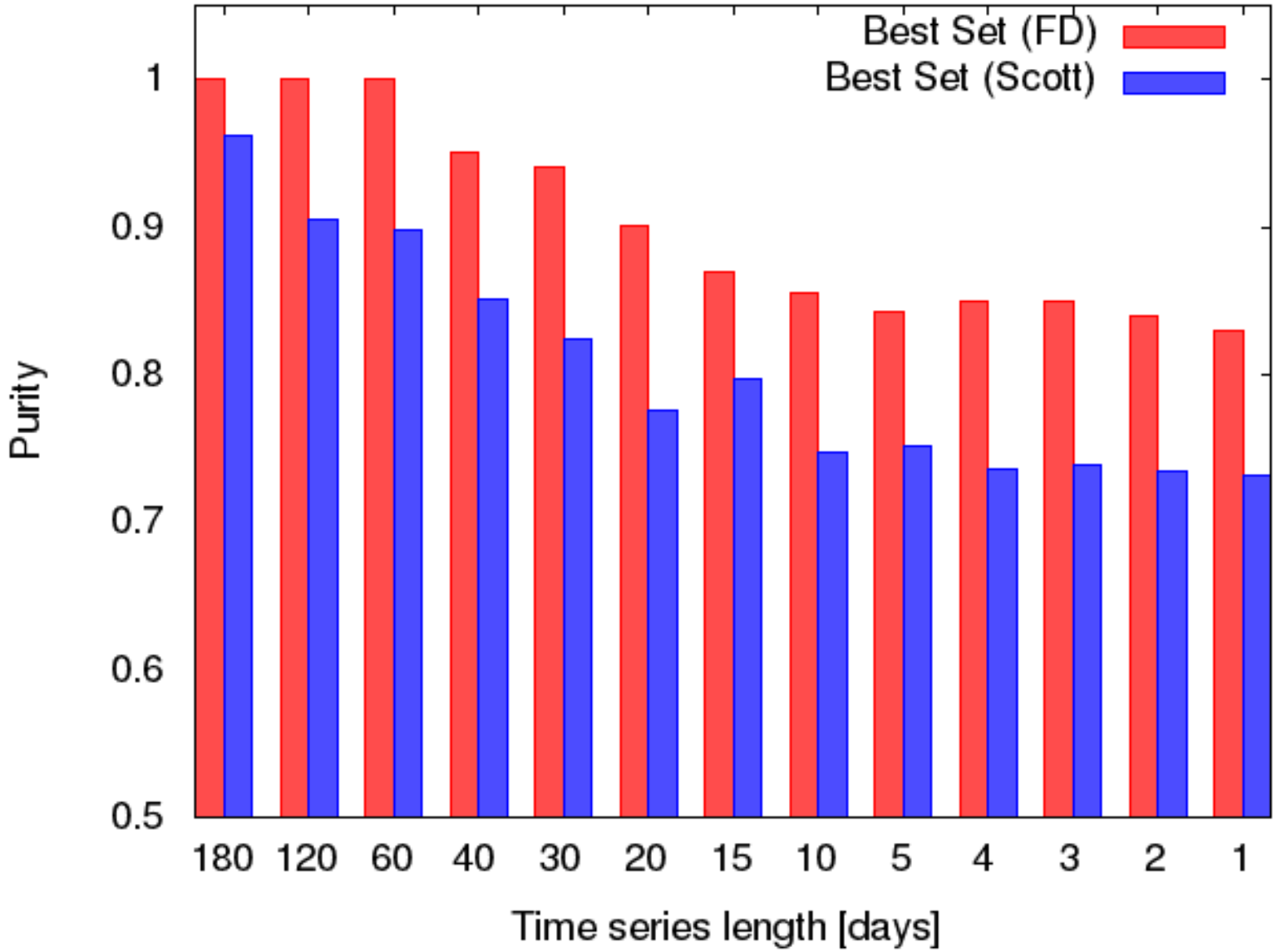
Mem paging

# of procs.

Impact of metric selection (2)



Impact of histogram characteristics



Conclusion and future work

- **Scalability in (multi)cloud systems → open issue**
- **Proposal of novel methodology to improve scalability through clustering of similar VMs**
- **Experimental results are encouraging**
  - Purity >0.83 even for very short time series
- **Future research directions:**
  - Validation with more data set *(Help!)*
  - Improving stability of the results w.r.t histogram parameters
  - Evaluate different models for VM behavior
  - Application of clustering to improve scalability of VM management

Automatic Virtual Machine Clustering based on Bhattacharyya Distance for Multi-Cloud Systems

C. Canali
R. Lancellotti

University of Modena and Reggio Emilia