

A Class-based Virtual Machine Placement Technique for a Greener Cloud

C. Canali

R. Lancellotti

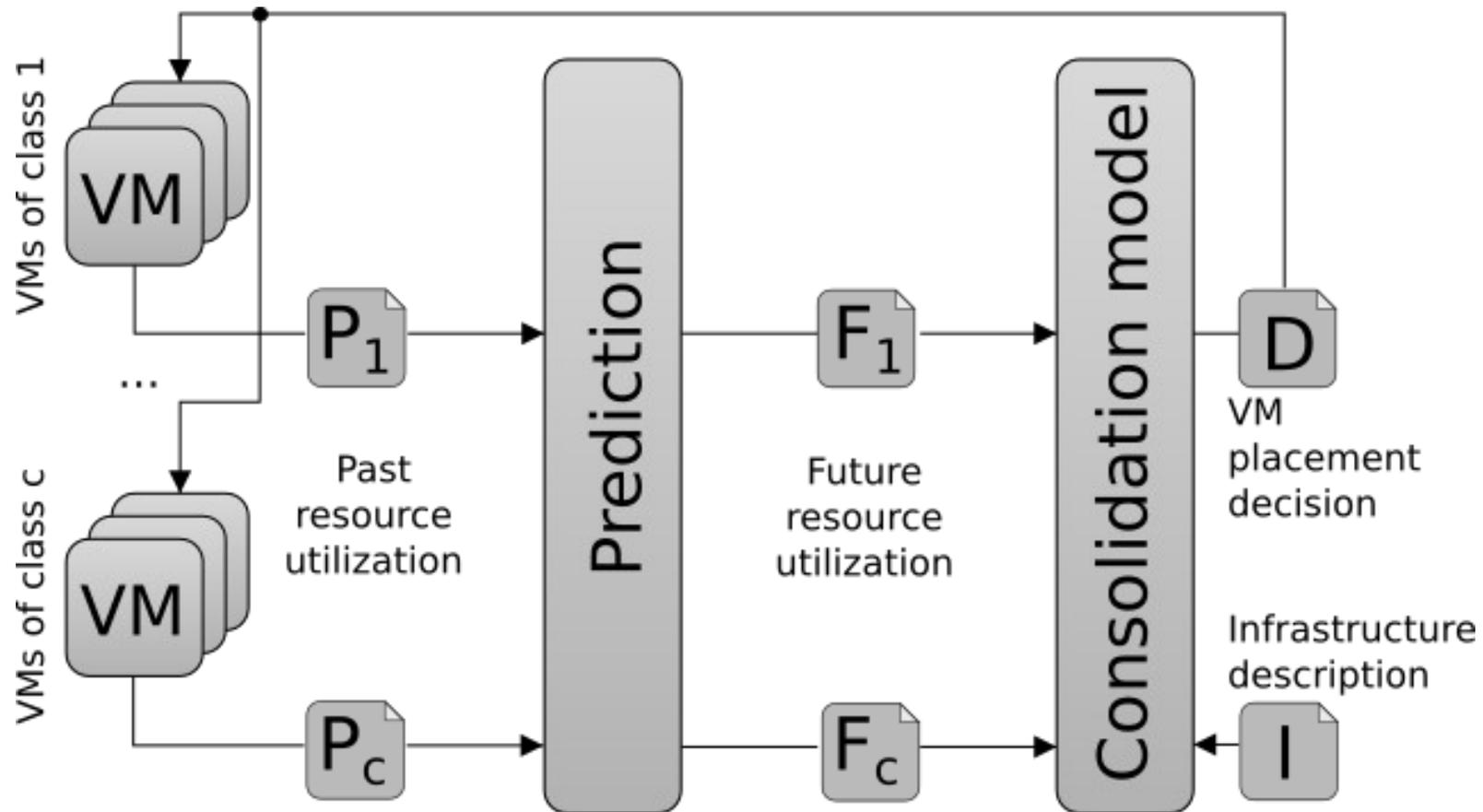
*Dipartimento di Ingegneria “Enzo Ferrari”
Università di Modena e Reggio Emilia*

Green Challenges of Cloud computing

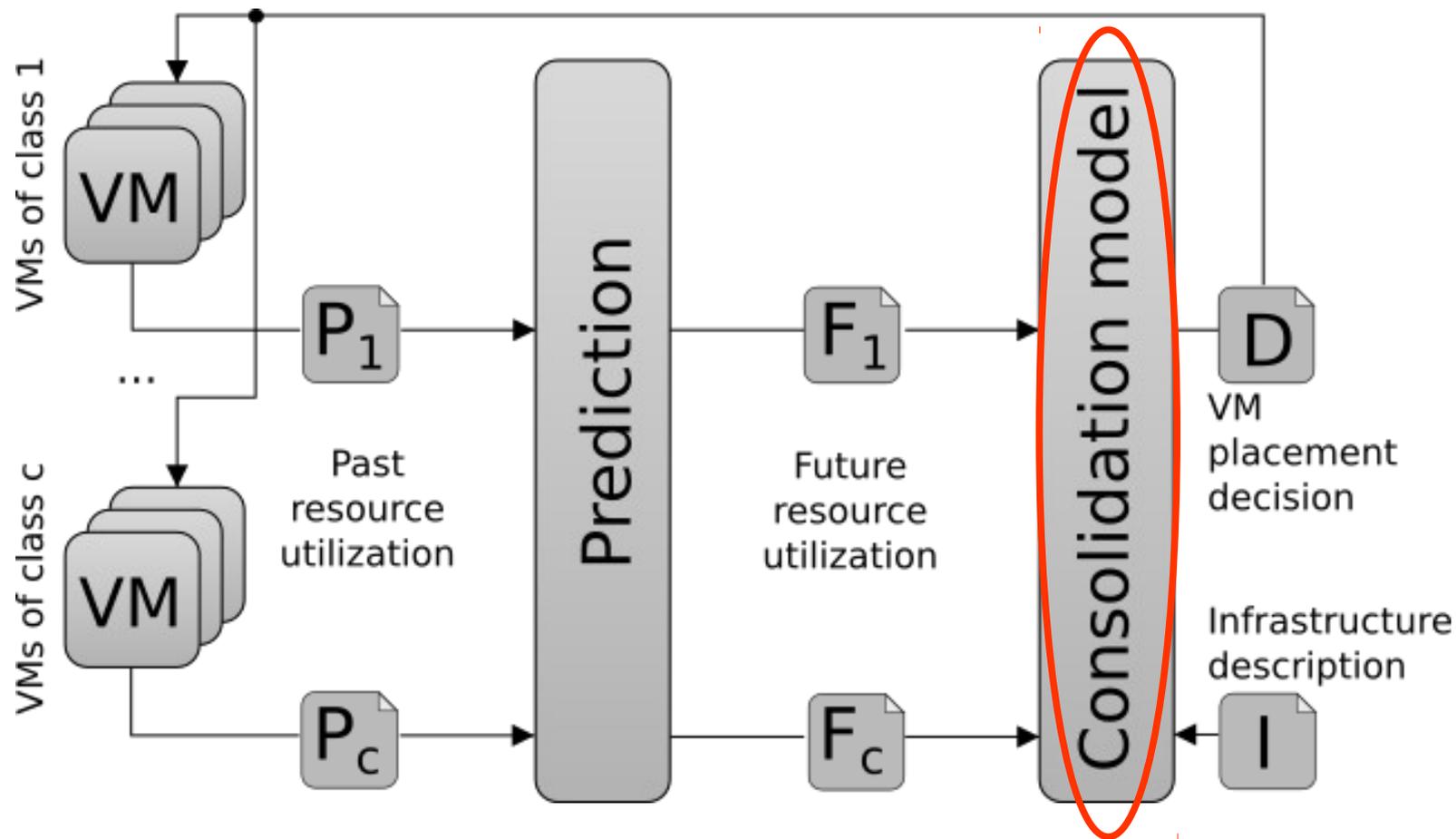


- Vision from a **laaS perspective**: → continuous growth
- More VMs, more data, ...
 - More data centers, Larger data centers
 - Higher carbon footprint
- **Some figures**:
 - Growth by 10^2 in 15 years in data
 - Power consumption: Internet comparable to Germany
- Need for **effective** data center management
- **Scalability problems** due to the infrastructure size:
 - placement of VMs over physical nodes

Reference scenario



Reference scenario



- **Assumptions:**

- VM placement: **periodic task**
- Knowledge of **VM classes** (similar resource requirements)

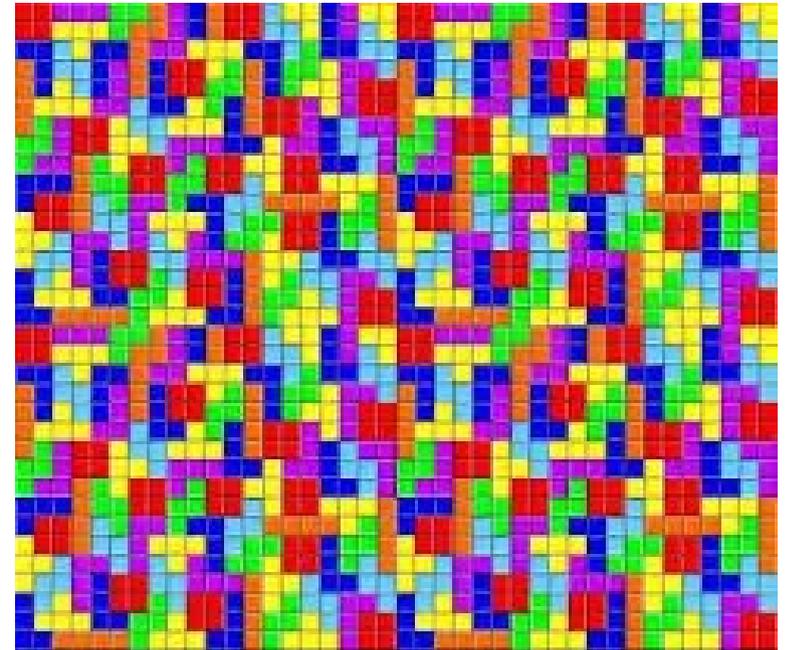
Consolidation model challenges



- **Large number of VMs → Many physical nodes**
- **Multiple metrics**
- **Sampling at multiple times**
 - Complementary workload patterns

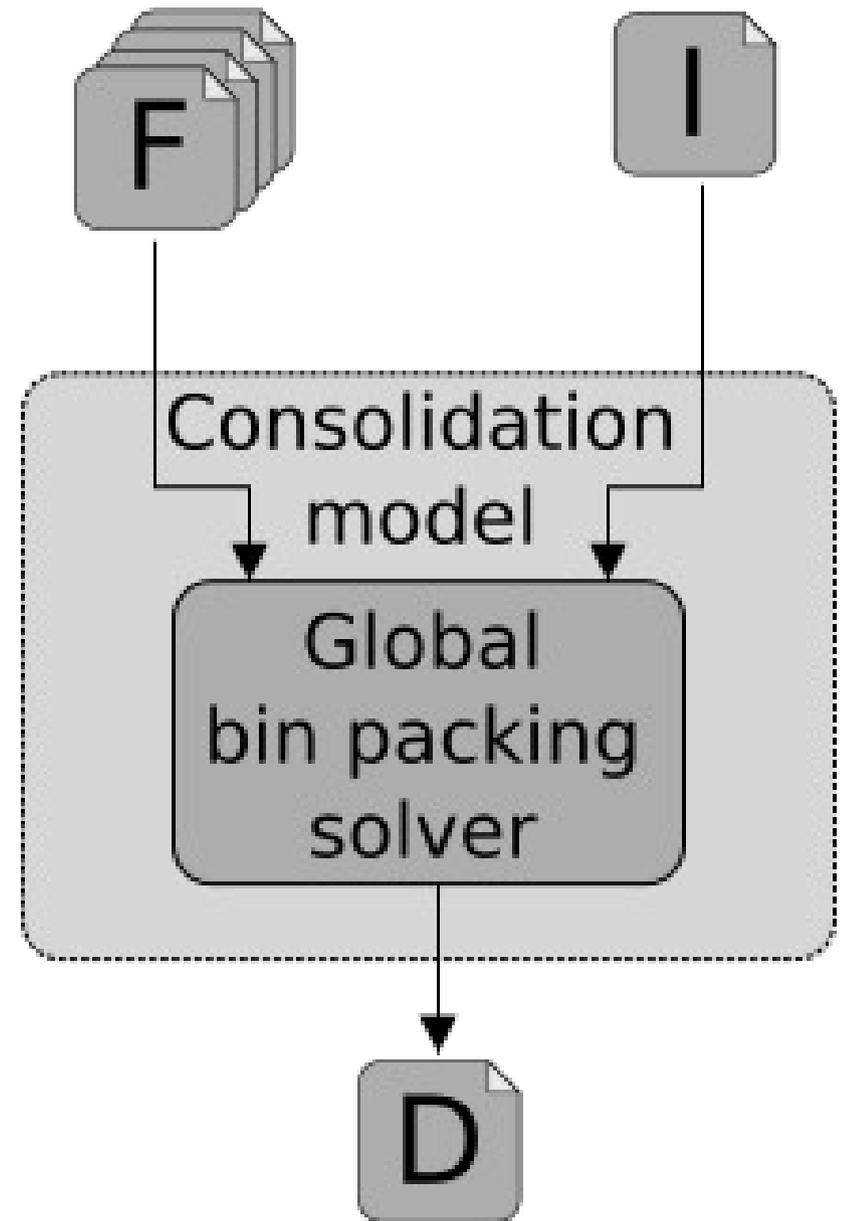
Consolidation model challenges

- Large number of VMs → Many physical nodes
- Multiple metrics
- Sampling at multiple times
 - Complementary workload patterns
- → **A huge, multidimensional Tetris game...**
- Trade-off:
 - Solution quality vs. Scalability



- **Consolidation model:**
 - Solution of optimization problem
- **Available alternatives:**
 - **Multi-dimensional bin packing (MBP)**
 - First Fit Decreasing Heuristic (FFD) – special case of bin packing: we consider only one dimension
 - **Class-based placement (CBP)**

- **Single bin-packing problem** for whole data center
- **Classes of VMs not considered**
- **Multi-dimensional problem:**
 - Multiple time intervals
 - ~~Multiple resources~~



Problem formulation

- **Objective function:**

$$\min \sum_{n \in N} O_n \quad \left. \vphantom{\sum_{n \in N} O_n} \right\} \text{Minimize number of nodes used}$$

- **Subject to:**

$$\sum_{n \in N} I_{n,m} = 1 \quad \left. \begin{array}{l} \text{Resource requirement} \\ \text{of VM } m \text{ at time } t \end{array} \right\} \forall m \in M \quad \left. \vphantom{\sum_{n \in N} I_{n,m} = 1} \right\} \text{1 VM in exactly one node}$$

$$\sum_{m \in M} R_{m,t} I_{n,m} \leq V_n O_n \quad \left. \vphantom{\sum_{m \in M} R_{m,t} I_{n,m} \leq V_n O_n} \right\} \forall n \in N, \forall t \in T \quad \left. \vphantom{\sum_{m \in M} R_{m,t} I_{n,m} \leq V_n O_n} \right\} \text{Node capacity constraint}$$

$$I_{n,m} = \{0,1\} \quad \forall n \in N, \forall m \in M$$

$$O_n = \{0,1\} \quad \forall n \in N$$

Decision variable: VM m on node n

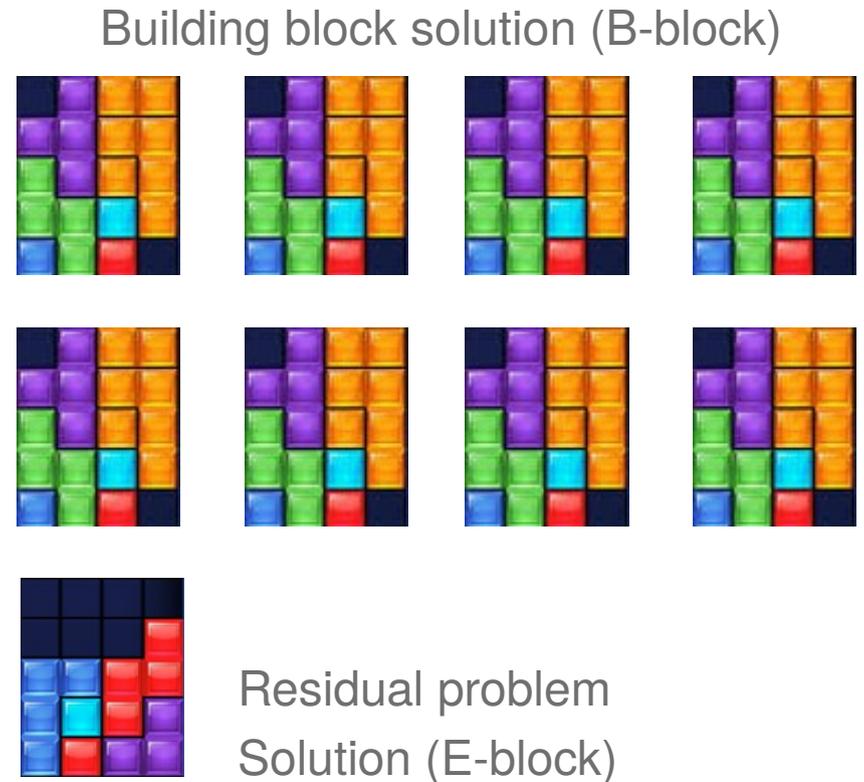
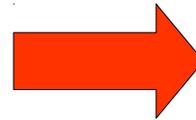
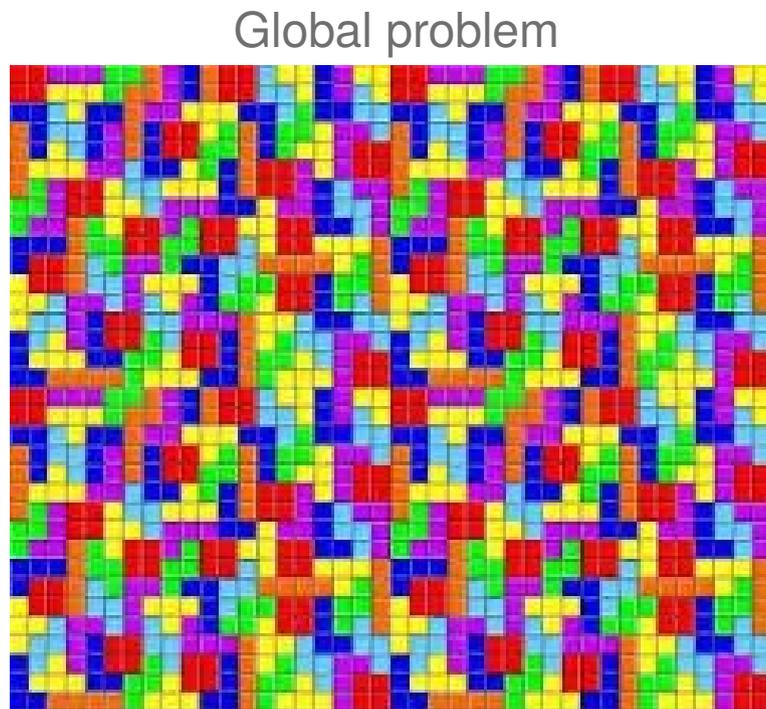
Decision variable: Node n is on/off

Available resources on node n

- **Number of node capacity constraints grows with:**
 - Number of nodes
 - Number of time intervals considered
- **Addressing scalability problems:**
 - **Wall time limit** on optimizer
 - **Reduce** number of **time intervals** (e.g., instead of 5min intervals can consider 1h, 4h, 12h, 1d...)
 - Use of **heuristics** instead of optimal solution
 - Special case: if only one time interval is considered multi-dimensional bin packing → bin packing (FFD)

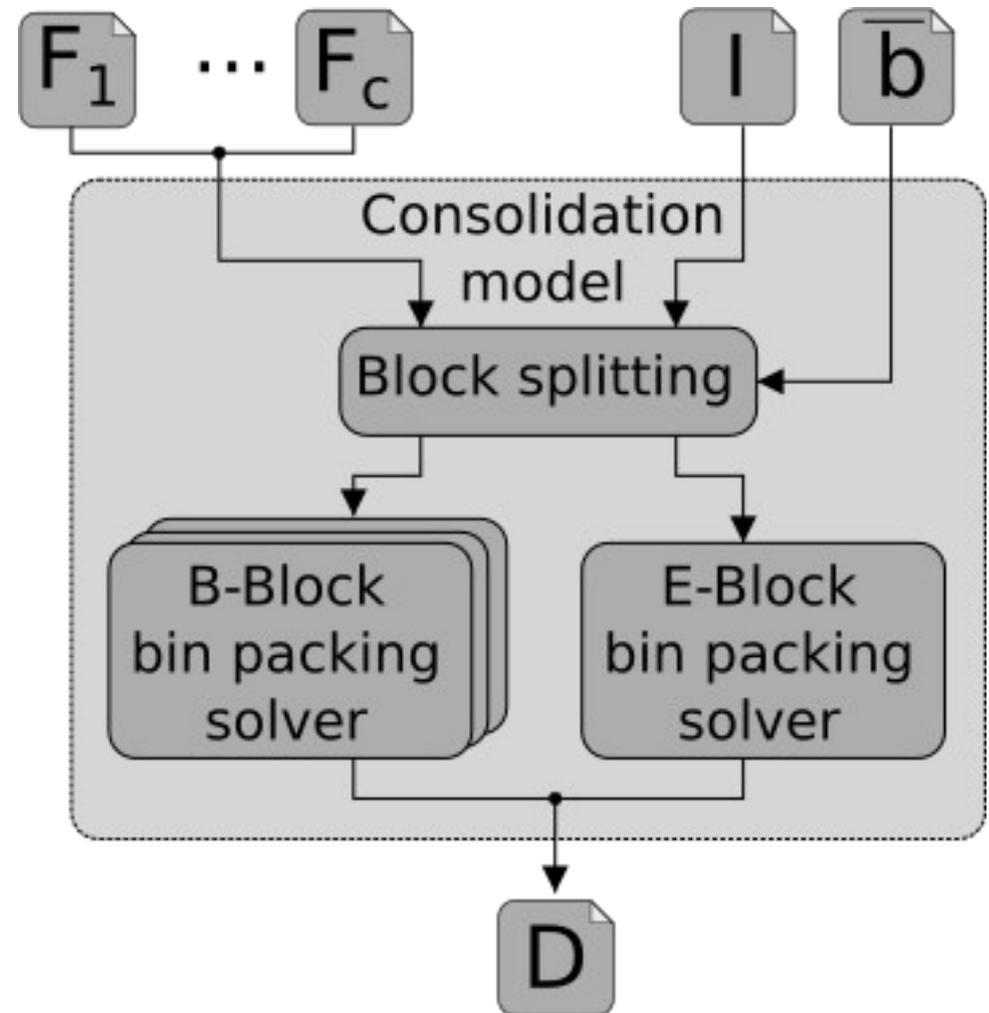
Class-based VM placement

- Build a small consolidation solution (B-block)
- Replicate solution as a building block
- Solve residual problem (E-Block)



Class-based VM placement

- **Additional input:**
→ number of B-blocks \bar{b}
- **Choice:** \bar{b} =n. of VMs in class with minimum cardinality
- **Impact of \bar{b}**
→ open issue
- **Two bin packing problems (B- E-blocks)**
- **Major dimensionality reduction → Scalability**



B-block problem formulation

- **Objective function:**

$$\min \sum_{n \in N_b} O_n \quad \left. \vphantom{\sum_{n \in N_b} O_n} \right\} \text{Minimize number of nodes used}$$

- **Subject to:**

$$\sum_{c \in C} \sum_{m \in B_c} R_{c,t} I_{n,m} \leq V_n O_n$$

$\sum_{n \in N_b} I_{n,m} = 1$ Resource requirement for class c at time t
 $I_{n,m} = \{0,1\}$
 $O_n = \{0,1\}$

Set of VM classes
 Set of VMs of class c in B-Block
 Set of VMs in B-Block
 Set of Nodes for B-Block

Class-based node capacity constraint

$$\forall m \in M_b$$

$$\forall n \in N_b, \forall t \in T$$

$$\forall n \in N, \forall m \in M_b$$

$$\forall n \in N_b$$

- **E- block problem formulation is similar**

Experimental setup



- **Number of VMs from 150 to 1200**
- **44 classes, each class [8-50] VMs**
- **Focus on CPU (only trace available) – Utilization: [0-100%]**
- **Each physical node has capacity of 800%**
- **Time intervals considered:**
 - 5m (288 int.)
 - 1h (24 int.)
 - 12h (2 int.)
 - 1d (1 int.)
- **IBM ILOG CPLEX Optimizer v12**
- **Maximum time for consolidation: 1800s (30m)**

Experimental results



- **MBP:**

- Optimal solution only for small problems (≤ 200)
- Reducing dimensionality improves scalability
- **No acceptable solutions for large problems (≥ 1200)**

- **CBP:**

- **Always reaches solution** even with 5m time interval
- Solves to optimality for medium problems (≤ 700)

VMs	CBP 5m	MBP 1d	MBP 12h	MBP 1h	MBP 5m
150	S/S	S	S	S	S
200	S/S	S	S	S	S
250	S/S	L	L	L	L
300	S/S	L	L	L	L
400	S/S	L	L	L	N
500	S/S	L	L	L	N
600	S/S	L	L	N	N
700	S/S	L	L	N	N
800	L/S	L	L	N	N
900	L/S	L	L	N	N
1000	L/S	L	L	N	N
1100	L/S	L	N	N	N
1200	L/S	N	N	N	N

Computation time

- **FFD: very fast but inaccurate**
- **When problem size grows, MBP becomes slower may result in sub-optimal solutions (quality reduction)**
- **CBP: very fast → scalable solution for larger problems**

Consolidation model	150 VMs	200 VMs	250 VMs	300 VMs
CBP 5m (B/E)	0.43/0.46	0.49/0.28	0.54/0.49	0.98/0.40
FFD 1d	0.05	0.05	0.06	0.07
MBP 1d	0.21	11.36	45.28	147.73
MBP 12h	4.13	79.39	1800(L)	1800(L)
MBP 1h	32.87	91.20	1800(L)	1800(L)
MBP 5 min	233.09	270.59	1800(L)	1800(L)

Solution quality: large number of VMs

- **MBP:**

- VM set size growth

→ lower quality

→ need time interval
reduction

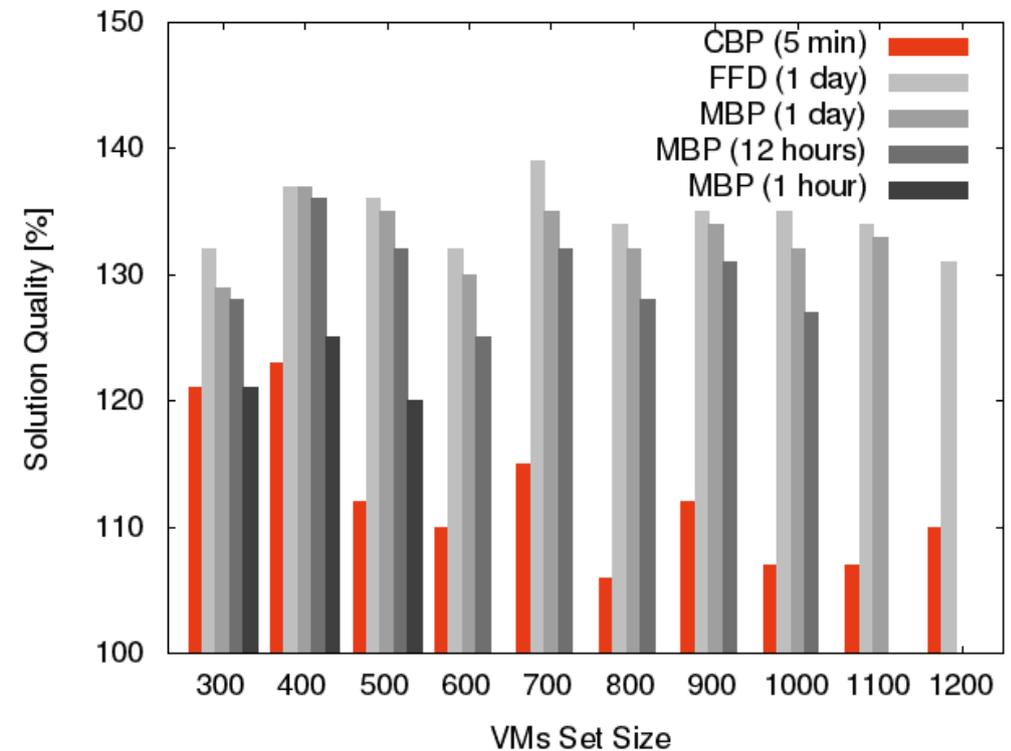
- **FFD: always worst performing**

- **CBP:**

- VM set size growth

→ always reaches solution

→ solution quality improves



- The challenge of VM placement in cloud computing
- **Proposal of Class-based placement technique**
- **Better scalability compared to alternatives:**
 - Can manage larger problems
 - Higher quality solution within the same time frame
- **Future work:**
 - New experiments: larger data centers, more resources
 - Analysis of B-block size (\bar{b} parameter):
 - impact on performance, automatic estimation
 - Different optimization strategies (e.g., dynamic programming)

- The authors acknowledge the support of the UniMORE-FAR2014 project **SAMMClouds**
- SAMMClouds: **S**ecure and **A**daptive **M**anagement of **M**ulti**C**louds
- **Open position available (12 month research grant)**



Exploiting Classes of Virtual Machines for Scalable IaaS Cloud Management

C. Canali

R. Lancellotti

*Dipartimento di Ingegneria “Enzo Ferrari”
Università di Modena e Reggio Emilia*

Solution quality: small number of VMs

- **Solution quality: relative to LP relaxation of the problem**
 - Lower is better
- **FFD: low quality results**
- **MBP:**
 - 5m: best solution
 - Time interval reduction
→ lower quality
 - VM set size growth
→ lower quality
- **CBP quality remains stable with problem size**

