Capacity Allocation for Big Data Applications in the Cloud

27th April 2017

QUDOS 2017@ICPE Workshop, L'Aquila

Michele Ciavotta

Eugenio Gianniti

Danilo Ardagna

DICE Horizon 2020 Project
Grant Agreement no. 644869
http://www.dice-h2020.eu

Funded by the Horizon 2020
Framework Programme of the European Union

Outline

- Background and motivations
- D-SPACE4Cloud Tool
- Experimental results
- Conclusions and future work

Background

- Data intensive applications (DIAs) hosted on public Clouds

- The goal is to optimize resource allocation at design time, taking into account quality of service constraints

D-SPACE4Cloud Tool

The problem:

- Minimize costs and suggest the optimal deployment architecture that provides QoS guarantees

What does the tool do?

- Automatic analysis of multiple candidate alternative configurations to identify the minimum cost one

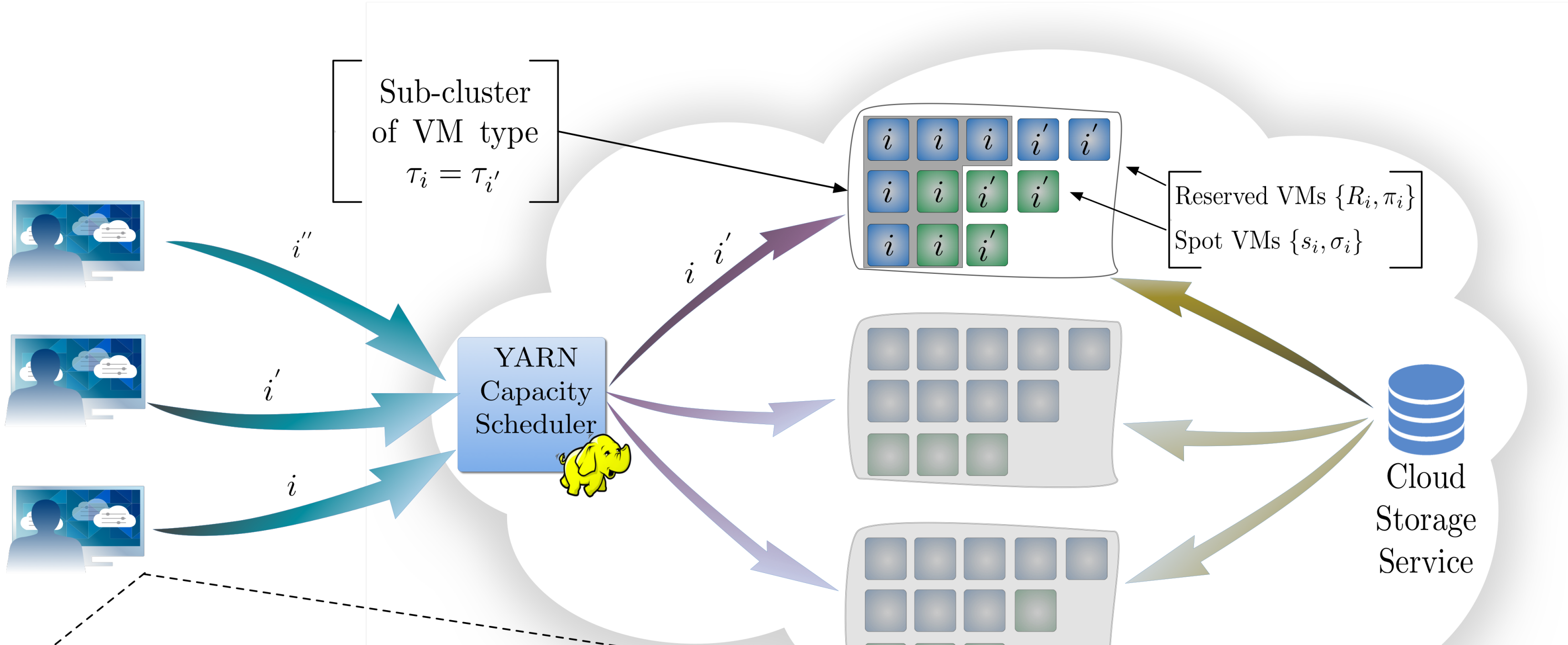
Innovation:

- Design space exploration has been increasingly sought in traditional multi-tier applications, but not in the design of DIAs

Impact & stakeholders:

- Designers and operators make more informed decisions about the technology to use
- Reduce costs of a shared cluster running multiple DIAs

Reference System



$$\mathcal{P}_{ij} = \begin{cases} n_i^M, \quad n_i^R \\ \mathcal{M}_{ij}^{max}, \quad \mathcal{R}_{ij}^{max}, \quad \mathcal{S}_{1,ij}^{max}, \quad \mathcal{S}_{typ,ij}^{max} \\ \mathcal{M}_{ij}^{avg}, \quad \mathcal{R}_{ij}^{avg}, \quad \mathcal{S}_{typ,ij}^{avg} \end{cases} \quad \forall i \in \mathcal{C}, \forall j \in \mathcal{V}$$

Complete Optimization Problem

$$\min_{\mathbf{x},\boldsymbol{\nu},\mathbf{s},\mathbf{R}} \sum_{i\in\mathcal{C}} (\sigma_{\tau_i} s_i + \pi_{\tau_i} R_i) \quad \text{(P1a)}$$

subject to:

$$\sum_{j\in\mathcal{V}} x_{ij} = 1, \quad \forall i \in \mathcal{C} \quad \text{(P1b)}$$

$$\mathcal{P}_{i,\tau_i} = \sum_{j\in\mathcal{V}} \mathcal{P}_{ij} x_{ij}, \quad \forall i \in \mathcal{C} \quad \text{(P1c)}$$

$$\sigma_{\tau_i} = \sum_{j\in\mathcal{V}} \sigma_j x_{ij}, \quad \forall i \in \mathcal{C} \quad \text{(P1d)}$$

$$\pi_{\tau_i} = \sum_{j\in\mathcal{V}} \pi_j x_{ij}, \quad \forall i \in \mathcal{C} \quad \text{(P1e)}$$

$$x_{ij} \in \{0,1\}, \quad \forall i \in \mathcal{C}, \forall j \in \mathcal{V} \quad \text{(P1f)}$$

$$(\boldsymbol{\nu}, \mathbf{s}, \mathbf{R}) \in \arg\min \sum_{i\in\mathcal{C}} (\sigma_{\tau_i} s_i + \pi_{\tau_i} R_i) \quad \text{(P1g)}$$

subject to:

$$s_i \leq \frac{\eta_i}{1-\eta_i} R_i, \quad \forall i \in \mathcal{C} \quad \text{(P1h)}$$

$$\nu_i = R_i + s_i, \quad \forall i \in \mathcal{C} \quad \text{(P1i)}$$

$$\mathcal{T}(\mathcal{P}_{i,\tau_i}, \nu_i; H_i, Z_i) \leq D_i, \quad \forall i \in \mathcal{C} \quad \text{(P1j)}$$

$$\nu_i \in \mathbb{N}, \quad \forall i \in \mathcal{C} \quad \text{(P1k)}$$

$$R_i \in \mathbb{N}, \quad \forall i \in \mathcal{C} \quad \text{(P1l)}$$

$$s_i \in \mathbb{N}, \quad \forall i \in \mathcal{C} \quad \text{(P1m)}$$

- Many integer variables and constraints make the problem intractable with exact methods
- We split the problem in two layers

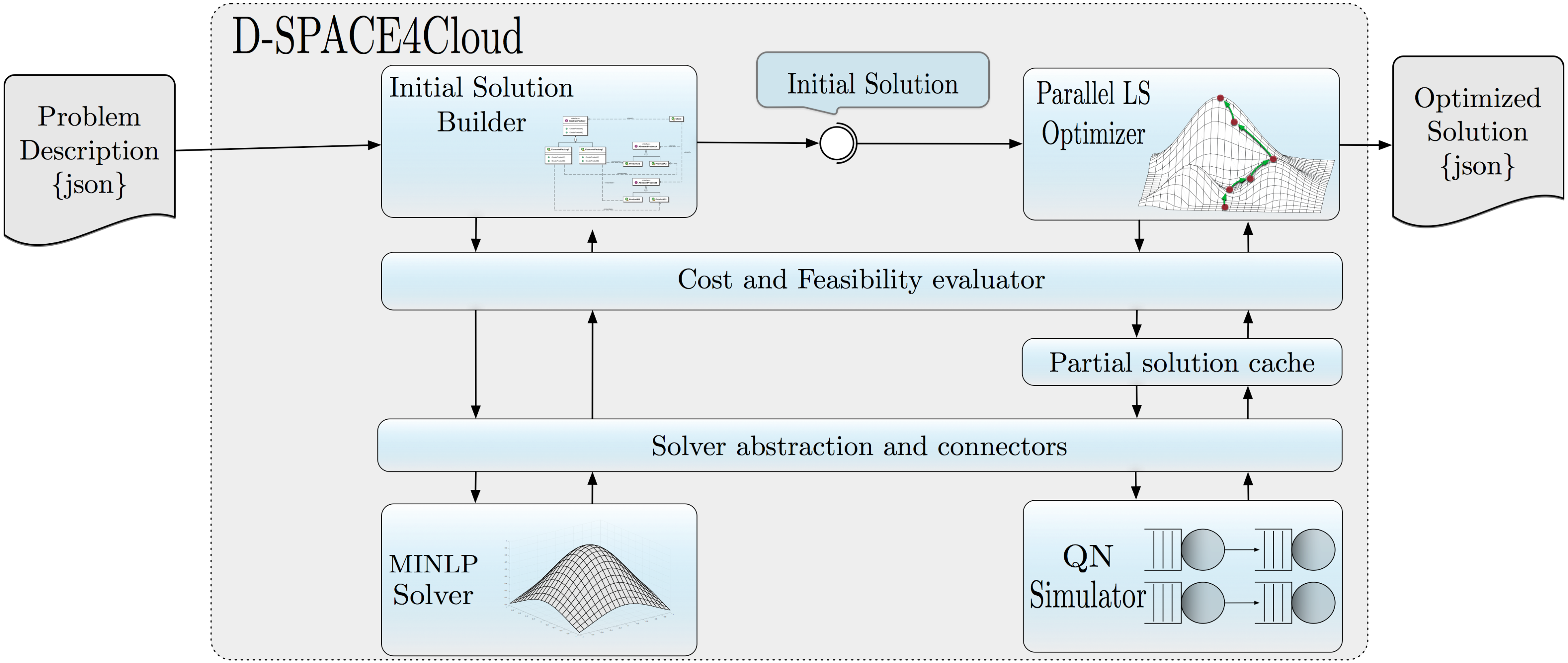
Local Search Motivations

- The mathematical programming problem is written with a raw performance prediction formula

- The optimum should also be accurate, hence we rely on simulation models

- There is the need to explore the design space
  - The initial guess might turn out to be infeasible
  - The initial guess might be overprovisioned

D-SPACE4Cloud Architecture



Local Search Method

- Apply hill climbing per class varying the VM allocation

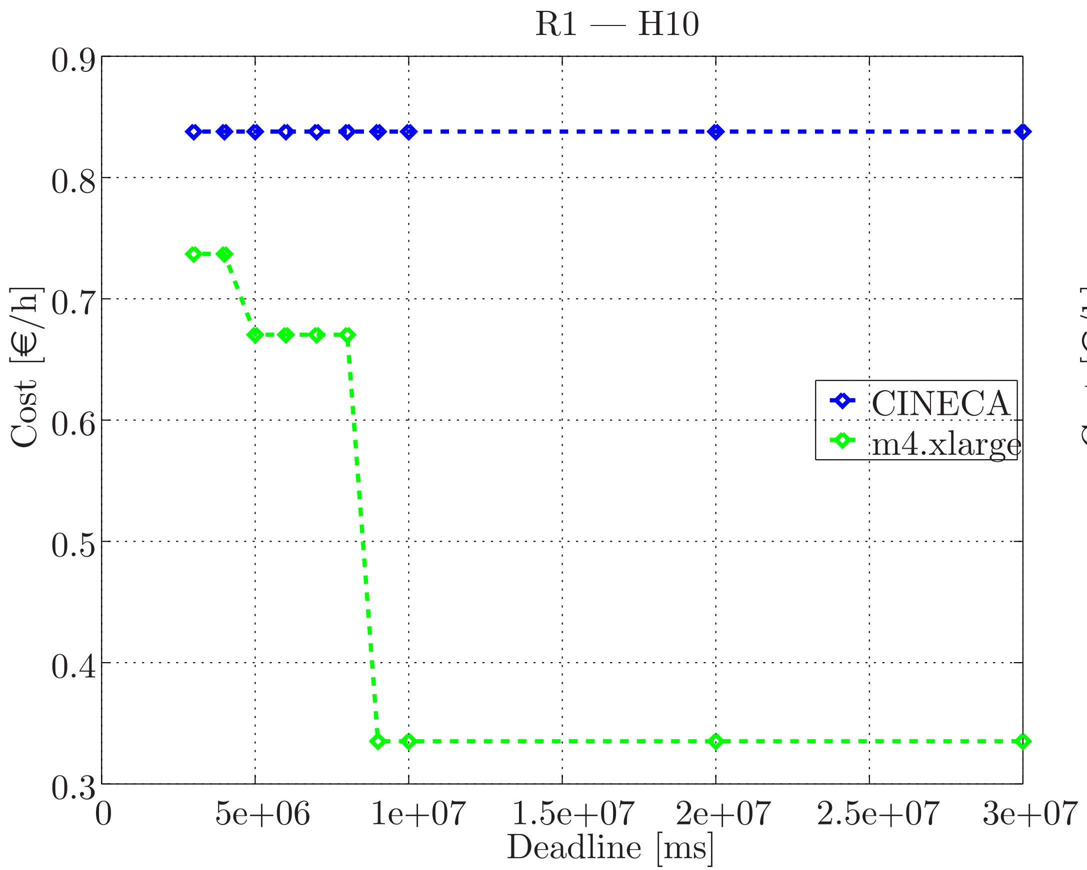
- Evaluate the optimal configuration returned by (P1) to choose the climbing direction
  - Remove instances if feasible
  - Add more VMs if infeasible

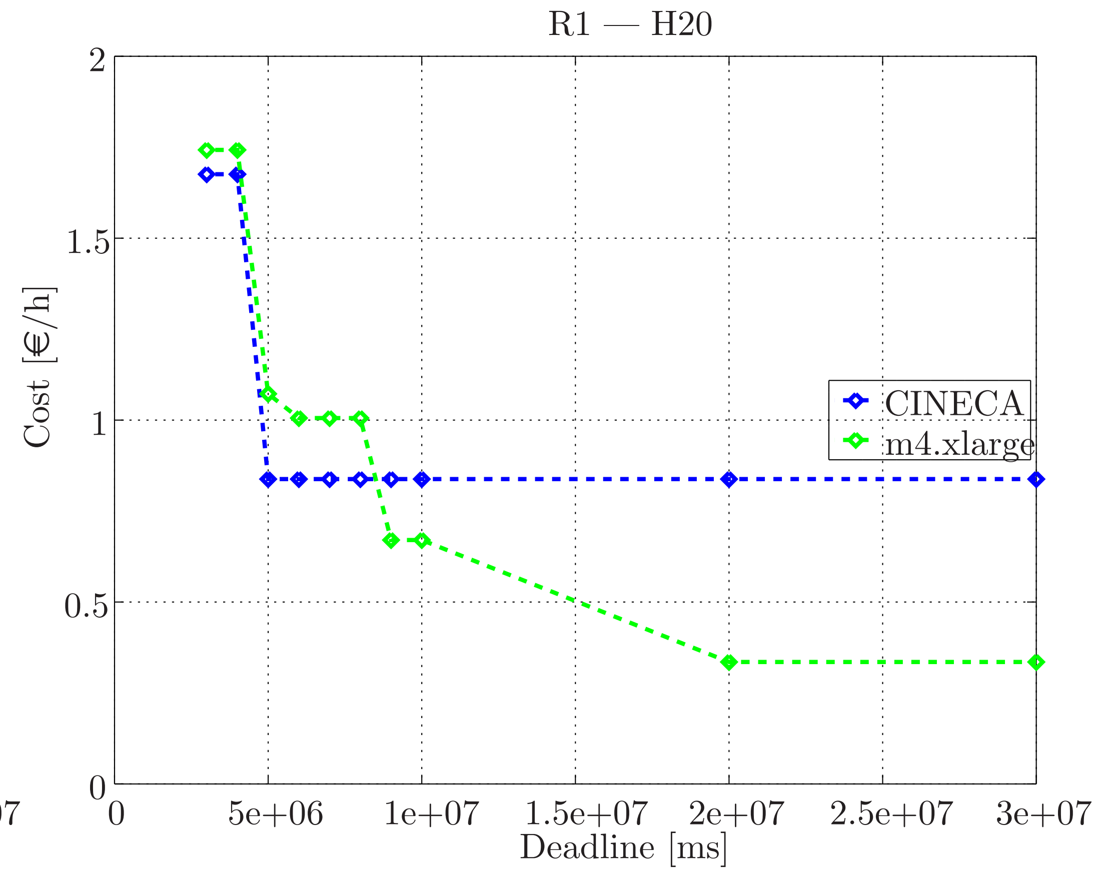
- Stop after reaching the local optimum

Simulation Models Validation

- TPC-DS benchmark, datasets ranging from 250 GB to 1 TB
- Experiments run on Amazon EC2, Cineca, Flexiant, with cluster sizes ranging from 20 to 240 cores
- Overall, 27,000 CPU hours worth of experiments

Optimal Cluster Cost





Conclusions

- D-SPACE4Cloud minimizes the overall cost under QoS constraints
- The tool supports a search technique to compare various providers and offerings
- Since we rely on accurate simulation models, we can reasonably trust the optimal configuration returned

Future Work

- Exploit machine learning and insight on the problem to improve heuristics efficiency
- Consider private or hybrid Clouds by adding capacity constraints
- Address other technologies: Spark and Storm

Thanks!

www.dice-h2020.eu