POLITECNICO DI MILANO

A Game-Theoretic Approach for Runtime Capacity Allocation in MapReduce

Eugenio Gianniti [*], **Danilo Ardagna** [*], Michele Ciavotta [*], Mauro Passacantando [**]

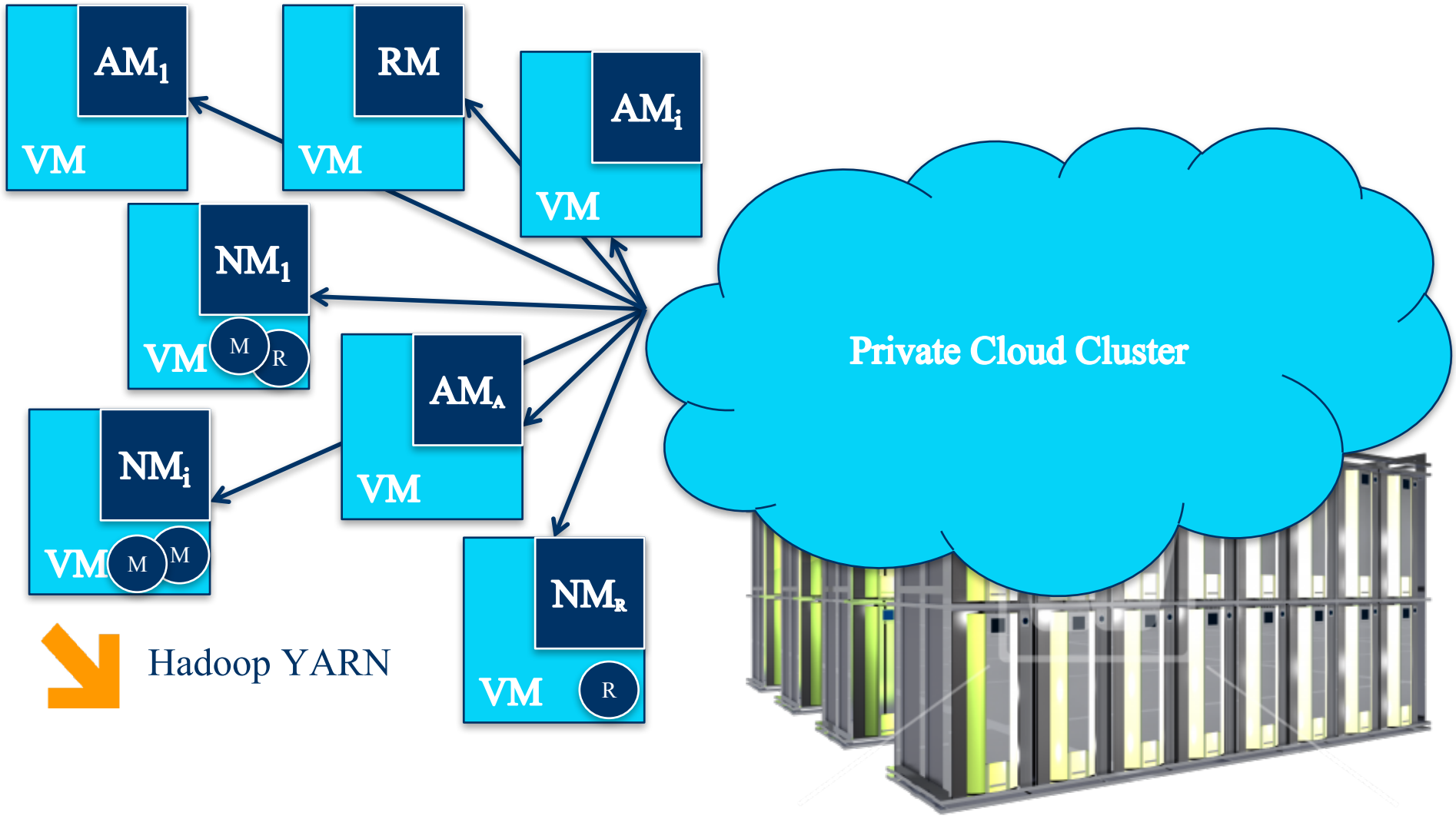
[*]Politecnico di Milano
[**]Università di Pisa

EUBRA BIGSEA
EUROPE - BRAZIL COLLABORATION OF BIG DATA SCIENTIFIC RESEARCH THROUGH CLOUD-CENTRIC APPLICATIONS

Goals

- Cost-effective deployment configuration for MapReduce systems hosted on private Clouds
- Distributed approach leveraging on local knowledge of problem parameters
- Efficient solution algorithm to support run-time cluster management

Reference System



Preliminary Formulation

$$\min_{\mathbf{r},\mathbf{h},\mathbf{s}^M,\mathbf{s}^R} \quad \sum_{i=1}^{N} \bar{\rho} r_i + \sum_{i=1}^{N} \mathcal{P}_i(h_i)$$

subject to:

$$\sum_{i=1}^{N} r_i \leq R$$

$$H_i^{low} \leq h_i \leq H_i^{up}, \quad \forall i \in \mathcal{A}$$

$$\frac{A_i h_i}{s_i^M} + \frac{B_i h_i}{s_i^R} + E_i \leq 0, \quad \forall i \in \mathcal{A}$$

$$\frac{s_i^M}{c_i^M} + \frac{s_i^R}{c_i^R} \leq r_i, \quad \forall i \in \mathcal{A}$$

$$r_i \in \mathbb{N}_0, \quad \forall i \in \mathcal{A}$$

$$s_i^M \in \mathbb{N}_0, \quad \forall i \in \mathcal{A}$$

$$s_i^R \in \mathbb{N}_0, \quad \forall i \in \mathcal{A}$$

$$h_i \in \mathbb{N}_0, \quad \forall i \in \mathcal{A}$$

DECISION VARIABLES

h_i — concurrency level

r_i — virtual machines

s_i^M — Map slots

s_i^R — Reduce slots

Integer Nonlinear Problem

Non-convex constraints

Continuous relaxation

$h_i \rightarrow (\psi_i)^{-1}$

Centralized Problem

$$\min_{\mathbf{r},\boldsymbol{\psi},\mathbf{s}^M,\mathbf{s}^R} \quad \sum_{i=1}^{N} \bar{\rho} r_i + \sum_{i=1}^{N} (\alpha_i \psi_i - \beta_i)$$

subject to:

$$\sum_{i=1}^{N} r_i \leq R$$

$$\psi_i^{low} \leq \psi_i \leq \psi_i^{up}, \quad \forall i \in \mathcal{A}$$

$$\frac{A_i}{s_i^M \psi_i} + \frac{B_i}{s_i^R \psi_i} + E_i \leq 0, \quad \forall i \in \mathcal{A}$$

$$\frac{s_i^M}{c_i^M} + \frac{s_i^R}{c_i^R} \leq r_i, \quad \forall i \in \mathcal{A}$$

$$r_i \in \mathbb{R}_+, \quad \forall i \in \mathcal{A}$$

$$s_i^M \in \mathbb{R}_+, \quad \forall i \in \mathcal{A}$$

$$s_i^R \in \mathbb{R}_+, \quad \forall i \in \mathcal{A}$$

$$\psi_i \in \mathbb{R}_+, \quad \forall i \in \mathcal{A}$$

Decision Variables

ψ_i — reciprocal concurrency level

r_i — virtual machines

s_i^M — Map slots

s_i^R — Reduce slots

Continuous Nonlinear Problem

Convex constraints

KKT conditions are necessary and sufficient for optimality

Need to centralize information

Optimal Configuration Formulae

PROPOSITION

In the optimal configuration it holds:

$$s_i^M = \frac{c_i^M}{1 + \sqrt{\frac{B_i}{A_i}\frac{c_i^M}{c_i^R}}} r_i, \quad \forall i \in \mathcal{A}$$

$$s_i^R = \frac{c_i^R}{1 + \sqrt{\frac{A_i}{B_i}\frac{c_i^R}{c_i^M}}} r_i, \quad \forall i \in \mathcal{A}$$

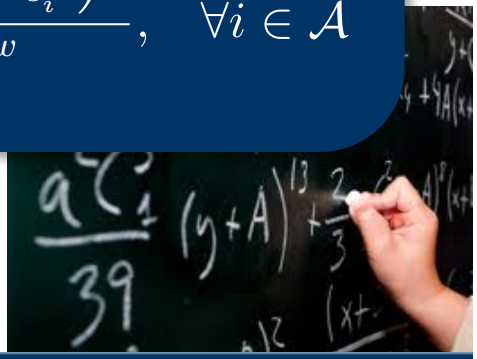
$$\psi_i = -\frac{\left(\sqrt{\frac{A_i}{c_i^M}} + \sqrt{\frac{B_i}{c_i^R}}\right)^2}{E_i} r_i^{-1}, \quad \forall i \in \mathcal{A}$$

✗ Full knowledge of the problem parameters

✓ Exact bounds on resources requirement

$$r_i^{low} = -\frac{\left(\sqrt{\frac{A_i}{c_i^M}} + \sqrt{\frac{B_i}{c_i^R}}\right)^2}{E_i \psi_i^{up}}, \quad \forall i \in \mathcal{A}$$

$$r_i^{up} = -\frac{\left(\sqrt{\frac{A_i}{c_i^M}} + \sqrt{\frac{B_i}{c_i^R}}\right)^2}{E_i \psi_i^{low}}, \quad \forall i \in \mathcal{A}$$



Application Masters Problems

$$\min_{\psi_i, \rho_i^a, s_i^M, s_i^R} \quad \alpha_i \psi_i - \beta_i$$

subject to:

$$\bar{\rho} \leq \rho_i^a \leq \rho_i^{up}$$

$$\psi_i^{low} \leq \psi_i \leq \psi_i^{up}$$

$$\frac{A_i}{s_i^M \psi_i} + \frac{B_i}{s_i^R \psi_i} + E_i \leq 0$$

$$\frac{s_i^M}{c_i^M} + \frac{s_i^R}{c_i^R} \leq r_i$$

$$s_i^M \in \mathbb{R}_+$$

$$s_i^R \in \mathbb{R}_+$$

$$\psi_i \in \mathbb{R}_+$$

$$\rho_i^a \in \mathbb{R}_+$$

Decision Variables

ψ_i — reciprocal concurrency level

ρ_i^a — bid for VMs

s_i^M — Map slots

s_i^R — Reduce slots

- Continuous Nonlinear Problem
- One instance per AM
- Only application-specific information

Resource Manager Problem

$$\max_{\mathbf{r},\mathbf{y},\rho} \quad \sum_{i=1}^{N} (\rho - \bar{\rho})\, \tilde{r}_i - \sum_{i=1}^{N} p_i \left(r_i^{up} - r_i\right)$$

subject to:

$$\sum_{i=1}^{N} r_i \leq R$$

$$r_i \geq r_i^{low}, \quad \forall i \in \mathcal{A}$$

$$r_i \leq \left(r_i^{up} - r_i^{low}\right) y_i + r_i^{low}, \quad \forall i \in \mathcal{A}$$

$$\bar{\rho} \leq \rho \leq \tilde{\rho}$$

$$\rho - \rho_i^a \leq M\,(1 - y_i)\,, \quad \forall i \in \mathcal{A}$$

$$\rho_i^a - \rho \leq M y_i, \quad \forall i \in \mathcal{A}$$

$$r_i \in \mathbb{R}_+, \quad \forall i \in \mathcal{A}$$

$$y_i \in \{0,1\}, \quad \forall i \in \mathcal{A}$$

$$\rho \in \mathbb{R}_+$$

Decision Variables

ρ — price of VMs

r_i — virtual machines

y_i — AM i offers more than price

- Mixed Integer Nonlinear Problem
- One instance for the whole cluster
- Takes care of resource management only
- Two alternatives: $\tilde{r}_i = r_i$ and $\tilde{r}_i = r_i - r_i^{low}$

Generalized Nash Equilibrium Problems

Optimal Configuration Formulae

PROPOSITION

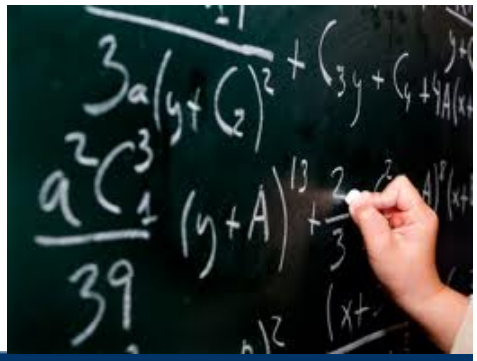
The optimal configuration for every AM, given an amount of resources r_i, is given by the following relations:

$$s_i^M = \frac{c_i^M}{1 + \sqrt{\frac{B_i}{A_i}\frac{c_i^M}{c_i^R}}} r_i$$

$$s_i^R = \frac{c_i^R}{1 + \sqrt{\frac{A_i}{B_i}\frac{c_i^R}{c_i^M}}} r_i$$

$$\psi_i = -\frac{\left(\sqrt{\frac{A_i}{c_i^M}} + \sqrt{\frac{B_i}{c_i^R}}\right)^2}{E_i} r_i^{-1}$$

- Local knowledge of application-specific parameters
- Each AM problem reduces to a quick algebraic update



Best Reply Algorithm

1: $r_i \leftarrow r_i^{low},\ \forall i \in \mathcal{A}$
2: $s_i^M \leftarrow s_i^{M,\,low},\ \forall i \in \mathcal{A}$
3: $s_i^R \leftarrow s_i^{R,\,low},\ \forall i \in \mathcal{A}$
4: $\psi_i \leftarrow \psi_i^{up},\ \forall i \in \mathcal{A}$
5: $\rho_i^a \leftarrow \bar{\rho},\ \forall i \in \mathcal{A}$
6: **repeat**
7: $\quad r_i^{old} \leftarrow r_i,\ \forall i \in \mathcal{A}$
8: $\quad$ solve RM problem
9: $\quad$ **for all** $i \in \mathcal{A}$ **do**
10: $\quad\quad$ solve AM i problem
11: $\quad\quad$ **if** $\psi_i > \psi_i^{low}$ **then**
12: $\quad\quad\quad \rho_i^a \leftarrow \max\{\rho_i^a, \rho\} + \lambda \rho_i^{up}$
13: $\quad\quad$ **end if**
14: $\quad$ **end for**
15: $\quad \varepsilon \leftarrow \sum_{i=1}^{N} \frac{|r_i - r_i^{old}|}{r_i^{old}}$
16: **until** $\varepsilon < \bar{\varepsilon}$

✓ Each iteration is performed in parallel by the RM and AMs

✗ Continuous equilibrium configuration

Experimental Overview

Preliminary Analysis

Ensure the model behavior is consistent with intuition when applied to simple problems

Scalability Analysis

Verify the feasibility of solving problem instances at a realistic scale in production environments

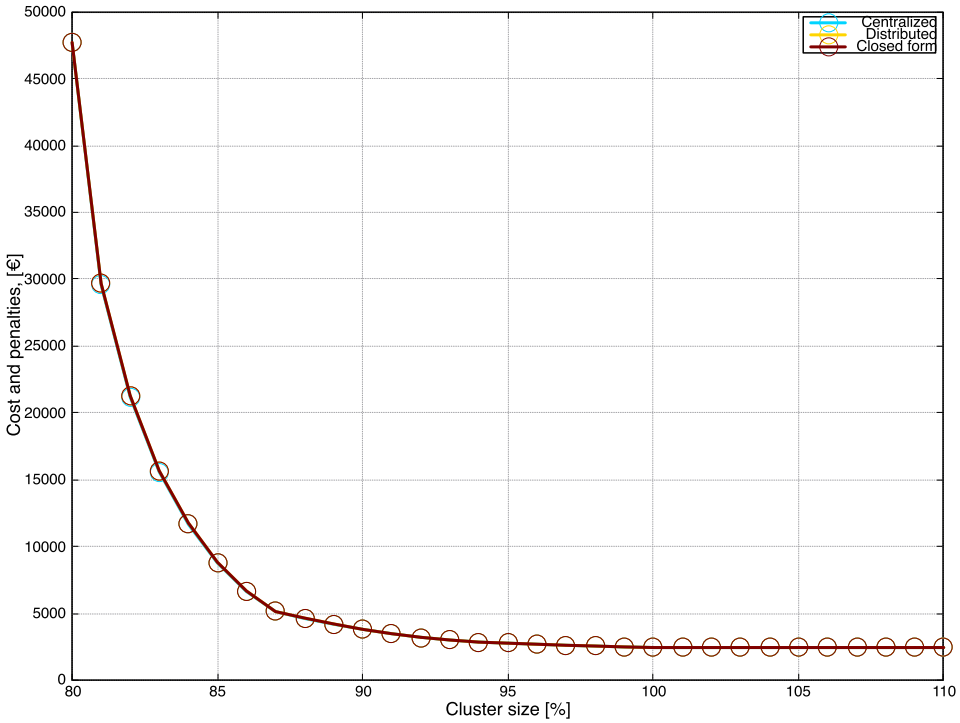
Stopping Criterion Tolerance Analysis

Check the sensitivity of the distributed algorithm with respect to the tolerance on the relative increment

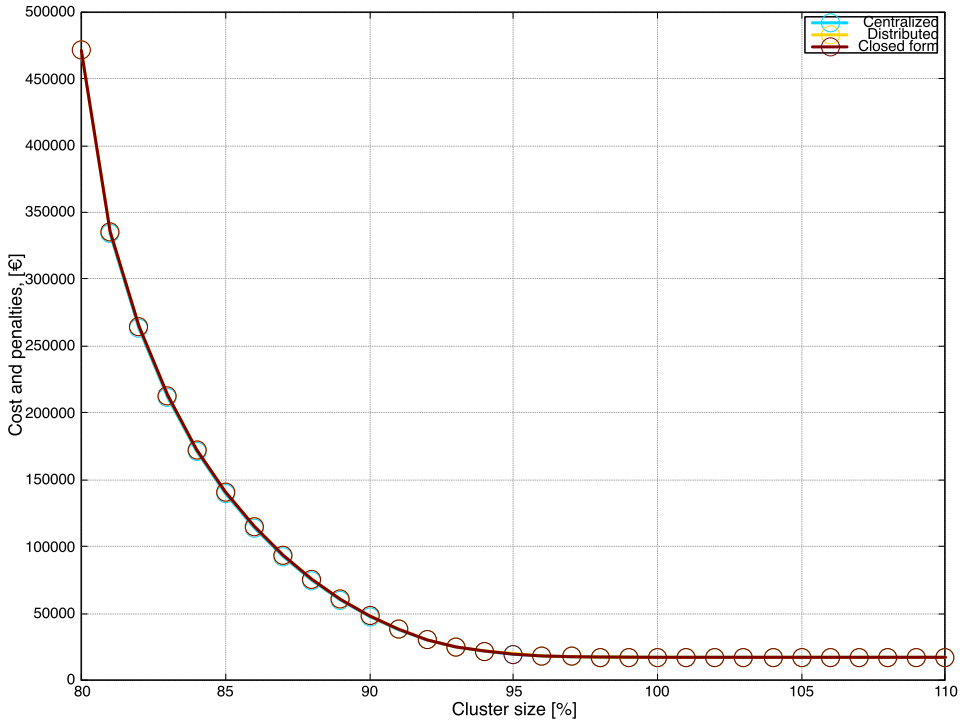
Validation with YARN SLS

Compare model solutions and timings measured on the official simulator

Preliminary Analysis



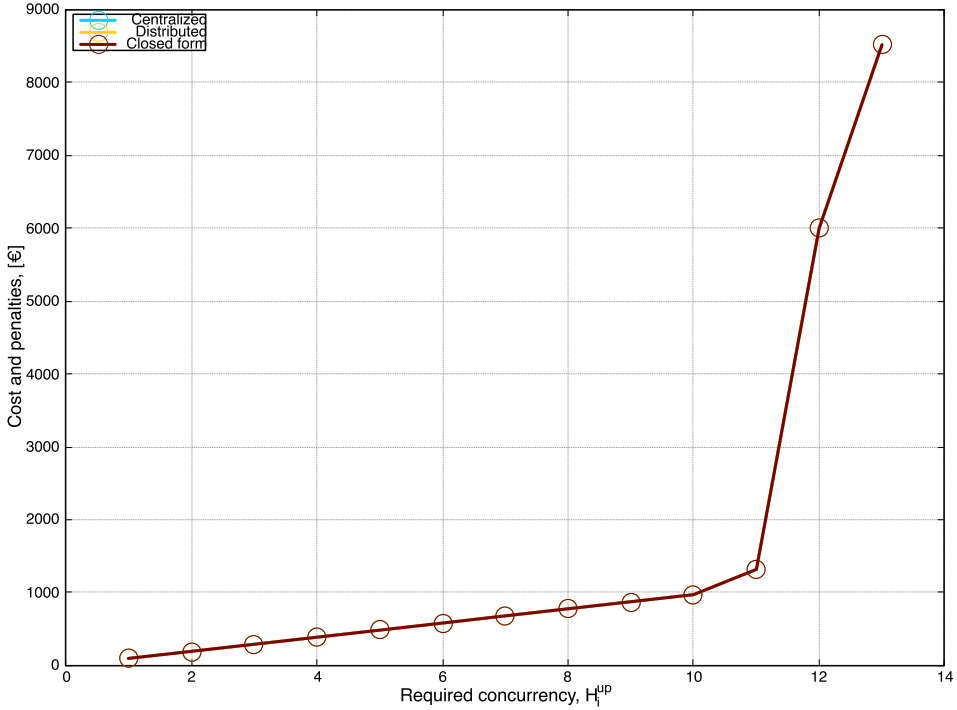
100 AMs



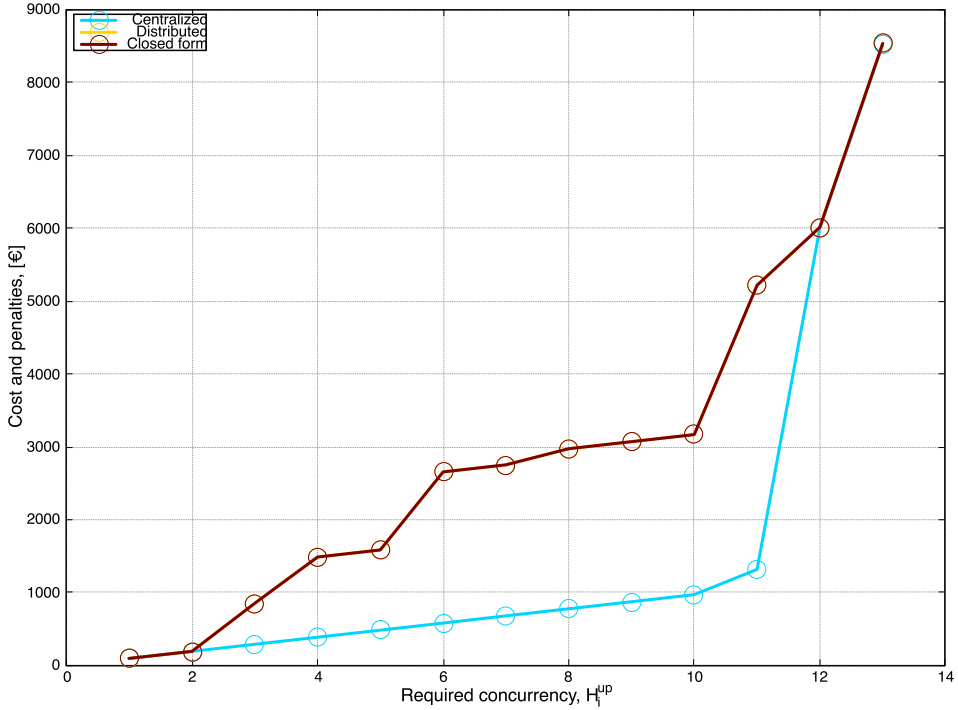
1,000 AMs

Decreasing cluster capacity experiment, $\tilde{r}_i = r_i - r_i^{low}$

Alternative Virtual Gain Terms



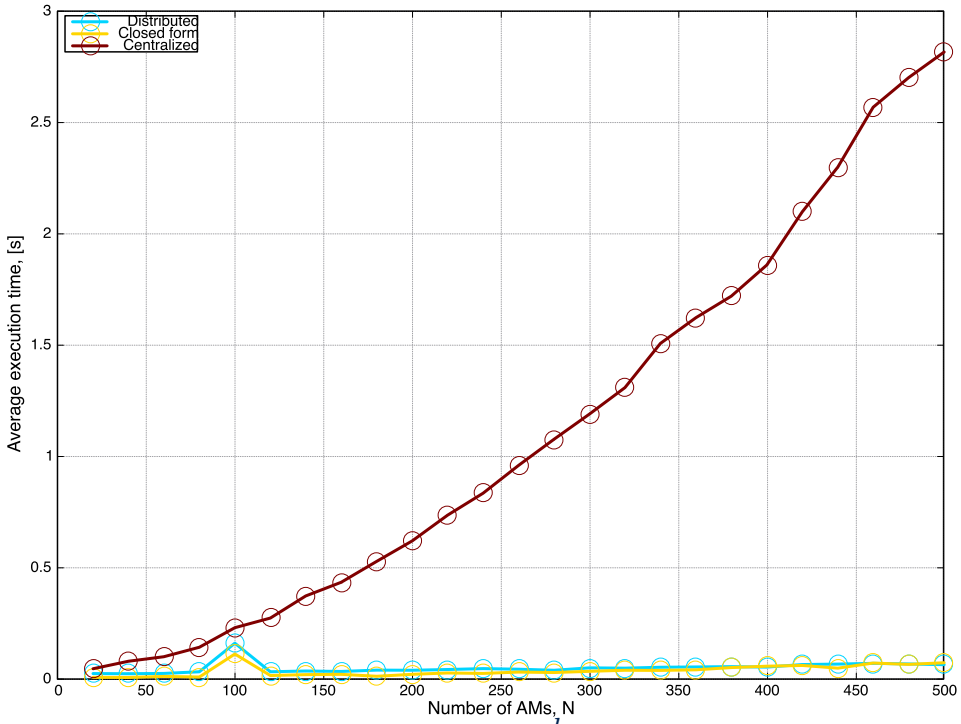
$$\tilde{r}_i = r_i - r_i^{low}$$



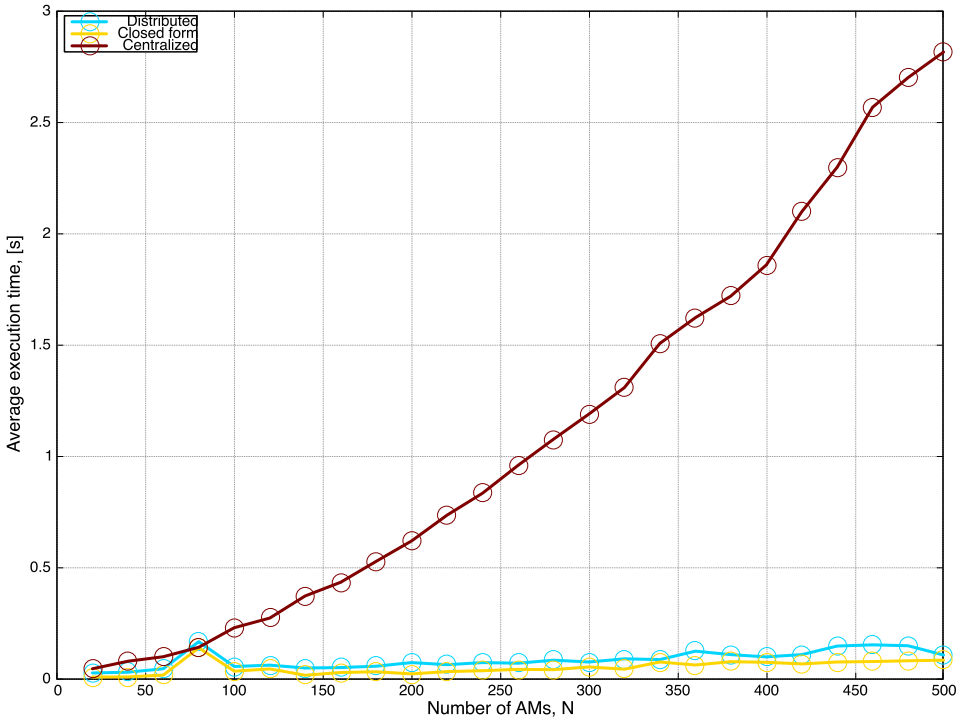
$$\tilde{r}_i = r_i$$

Increasing concurrency level experiment, 10 AMs

Scalability Analysis



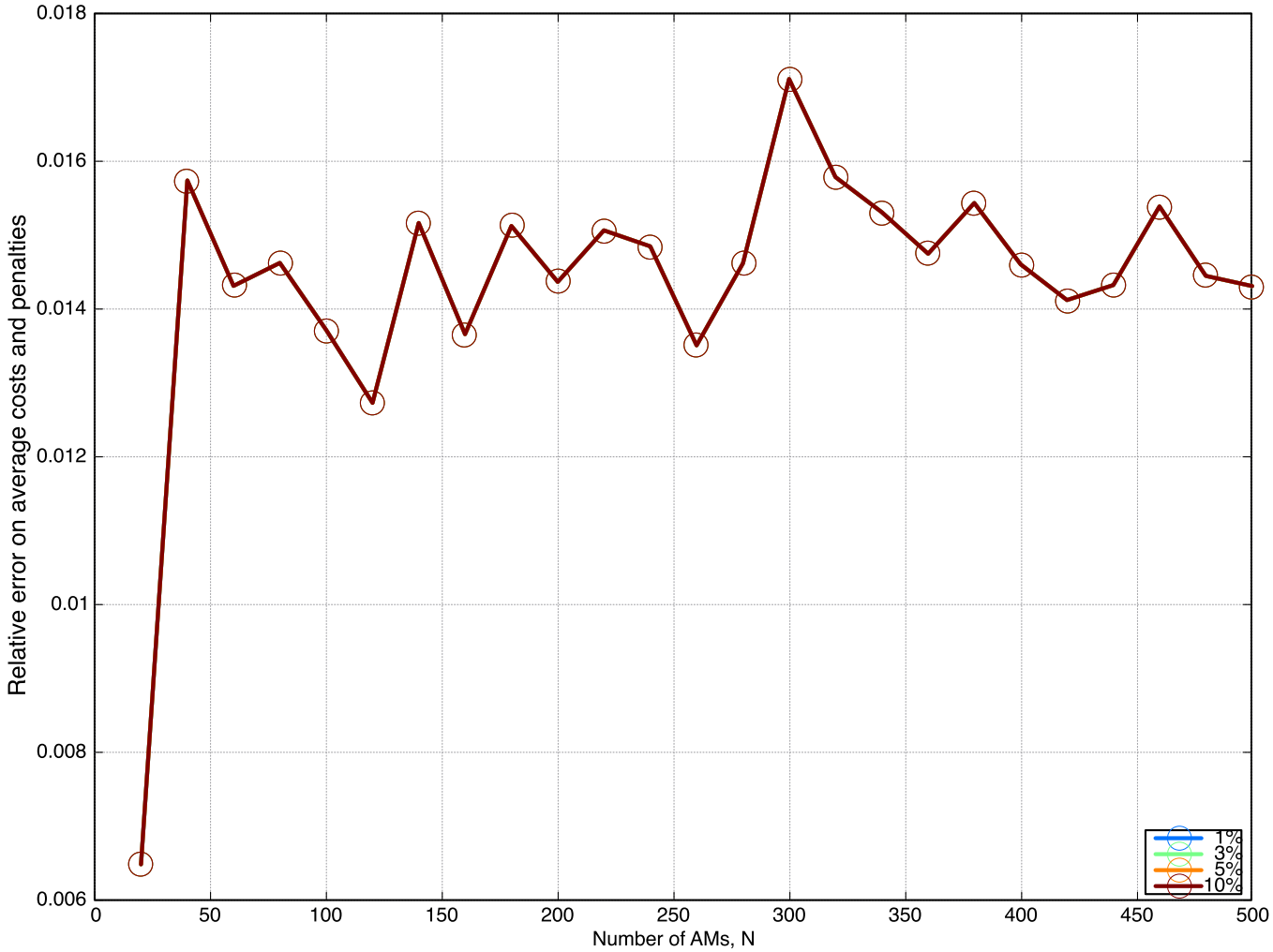
$$\tilde{r}_i = r_i - r_i^{low}$$



$$\tilde{r}_i = r_i$$

Execution time [s]

Stopping Criterion Tolerance Analysis



Relative error with respect to centralized solutions

Validation with YARN SLS

Users	R	D_i [s]	S [s]	η [%]
(10, 15, 20)	256	2740	2767.58	-0.996658
(10, 15, 20)	512	1508	1447.12	4.20698
(10, 20, 15)	256	2900	2708.3	7.07837
(10, 20, 15)	512	1589	1379.58	15.18
(15, 10, 10)	256	2575	2257.81	14.0487
(15, 10, 10)	512	1456	980.087	48.5583
(15, 15, 15)	256	3070	3183.02	-3.55061
(15, 15, 15)	512	1678	1456.93	15.174
(15, 15, 20)	256	3210	3088.88	3.92116
(15, 15, 20)	512	1748	1547.92	12.9255
(15, 20, 10)	256	3220	3070.87	4.85639
(15, 20, 10)	512	1755	1328.58	32.0956
(15, 20, 20)	256	3512	3951.58	-11.1242
(15, 20, 20)	512	1899	1574.1	20.6404
(15, 25, 10)	256	3520	3276.66	7.42646
(15, 25, 10)	512	1906	1524.83	24.9975

Users	R	D_i [s]	S [s]	η [%]
(20, 10, 10)	256	3090	2487.54	24.2191
(20, 10, 10)	512	1694	1142.3	48.2977
(20, 10, 20)	256	3380	2948.78	14.6237
(20, 10, 20)	512	1835	1299.59	41.1987
(20, 15, 10)	256	3390	2849.85	18.9536
(20, 15, 10)	512	1845	1409.88	30.8625
(20, 20, 10)	256	3700	3339.28	10.8023
(20, 20, 10)	512	1995	1512	31.9444
(20, 20, 15)	256	3845	3694.03	4.08677
(20, 20, 15)	512	2063	1745.9	18.1626
(20, 20, 20)	256	3987	4041.44	-1.34704
(20, 20, 20)	512	2140	1877.4	13.9874
(25, 15, 25)	256	4300	3893.71	10.4346
(25, 15, 25)	512	2290	1773.53	29.1213
(25, 25, 25)	512	2597	2653.64	-2.13455

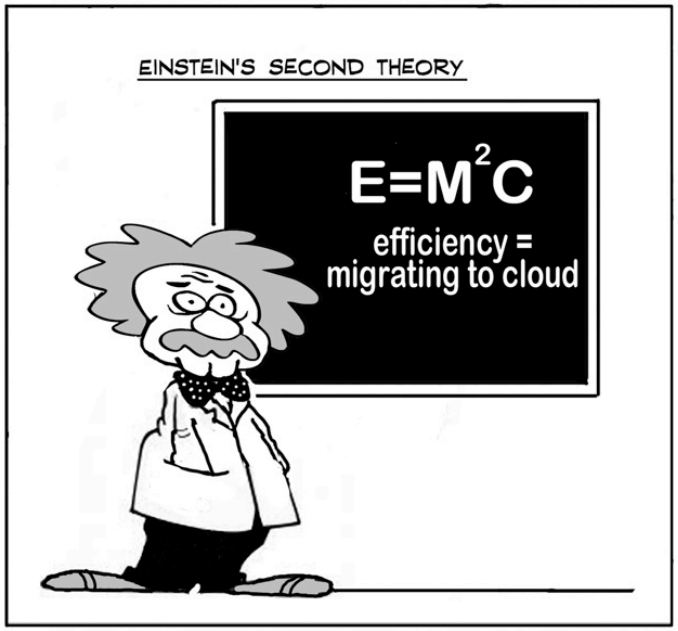
Relative error absolute values average: 16.999 %

Conclusions and Future Work

- The distributed approach yields accurate approximations of optimal solutions, even without theoretical guarantees
- Extend the formulation to Apache Tez and Spark (based on DAGs)
- Couple the proposed algorithm with a local search method based on Colored Petri Nets simulations

Thanks for your attention...

...any questions?

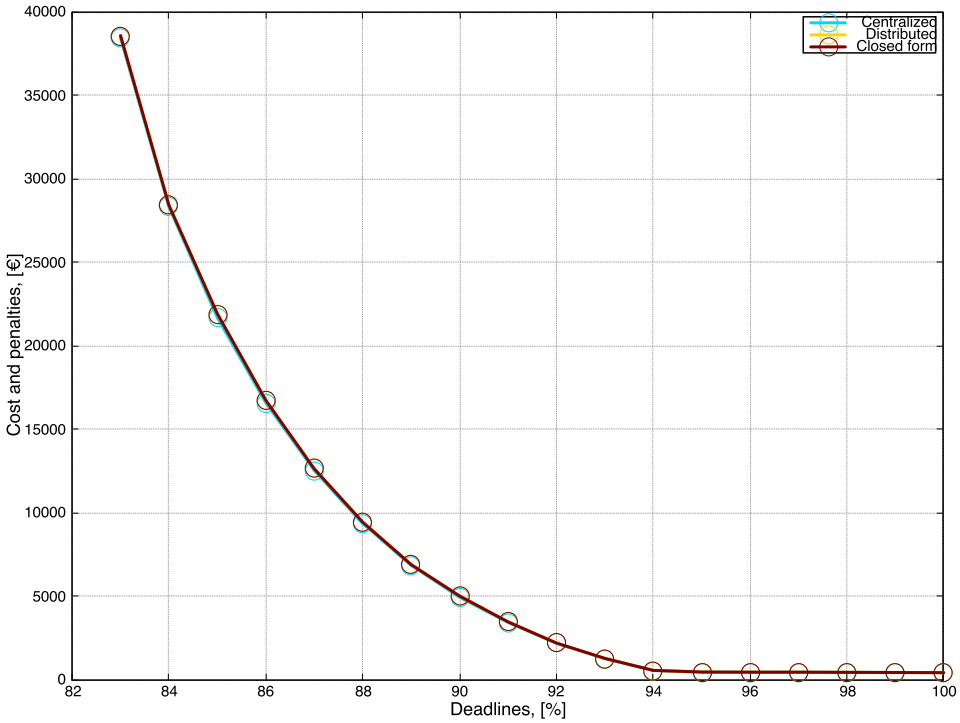


Solution Rounding Algorithm

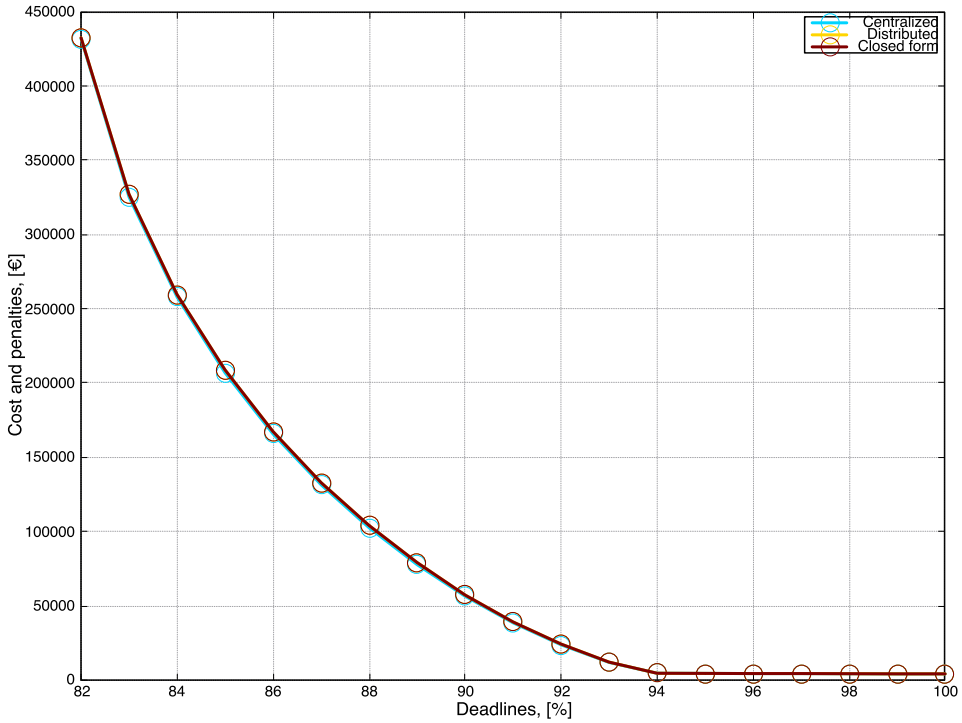
1: sort $\mathcal{A}$ according to increasing α_i
2: $r_i \leftarrow \lceil \hat{r}_i \rceil$, $\forall i \in \mathcal{A}$
3: **for all** $j \in \mathcal{A}$ **do**
4: **if** $\sum_{i=1}^{N} r_i > R$ **then**
5: $r_j \leftarrow r_j - 1$
6: **end if**
7: **end for**
8: $s_i^M \leftarrow \lceil \hat{s}_i^M \rceil$, $\forall i \in \mathcal{A}$
9: $s_i^R \leftarrow \lceil \hat{s}_i^R \rceil$, $\forall i \in \mathcal{A}$
10: **for all** $j \in \mathcal{A}$ **do**
11: **while** $s_j^M / c_j^M + s_j^R / c_j^R > r_j$ **do**
12: $s_j^R \leftarrow s_j^R - 1$
13: **if** $s_j^M / c_j^M + s_j^R / c_j^R > r_j$ **then**
14: $s_j^M \leftarrow s_j^M - 1$
15: **end if**
16: **end while**
17: **end for**

- The RM runs an *O(N)* loop
- Each AM runs an *O(1)* loop, concurrently
- Sorting is *O(N* log *N)*, but can be done once and cached

Preliminary Analysis



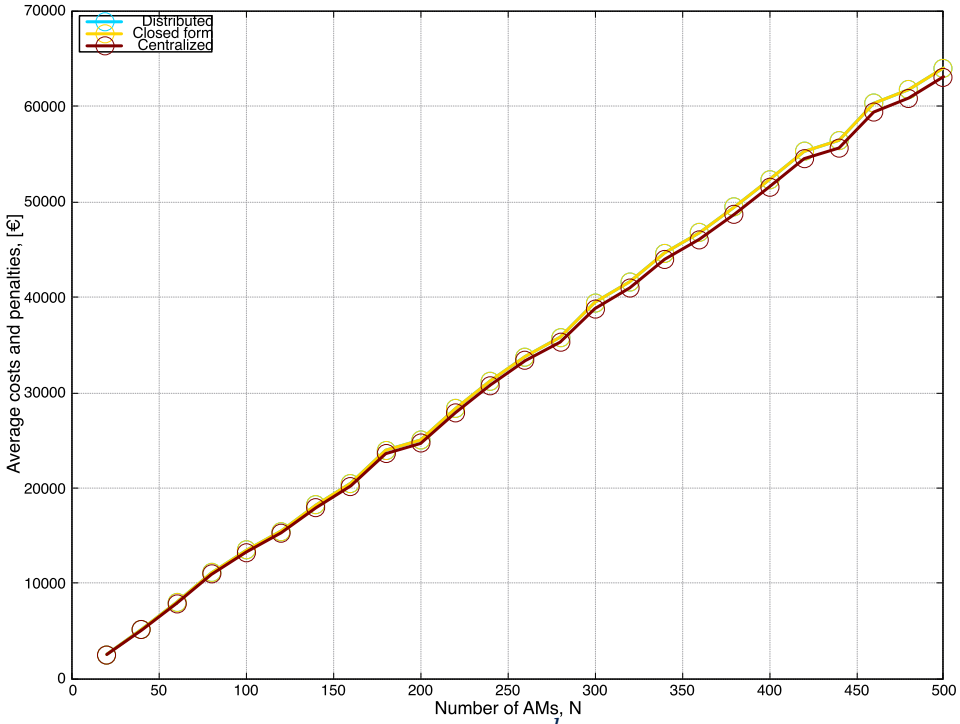
100 AMs



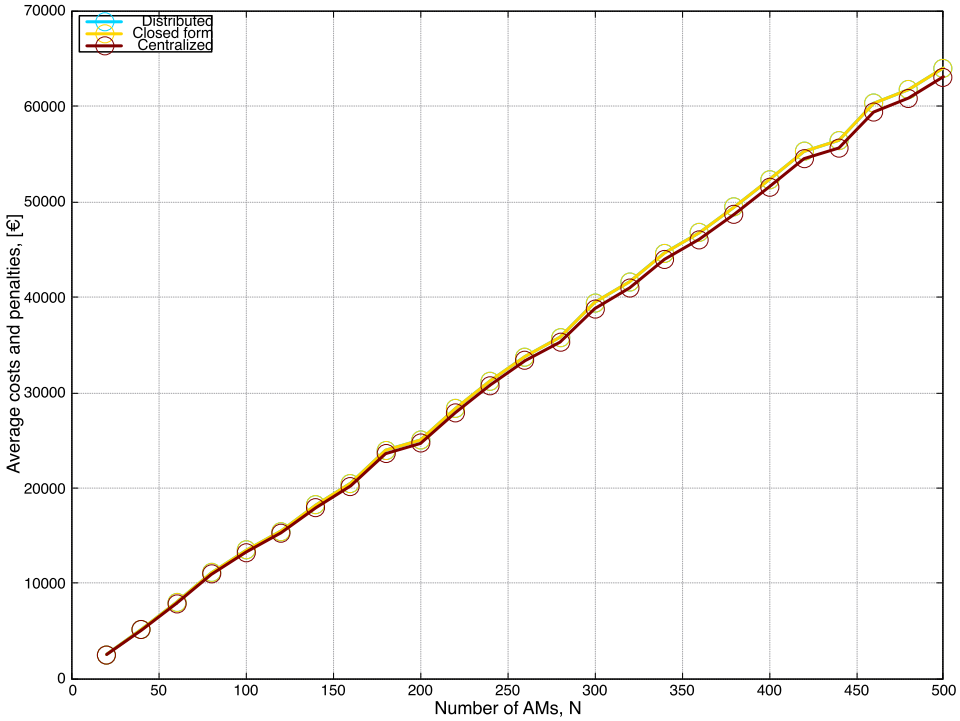
1,000 AMs

Decreasing deadlines experiment, $\tilde{r}_i = r_i - r_i^{low}$

Scalability Analysis



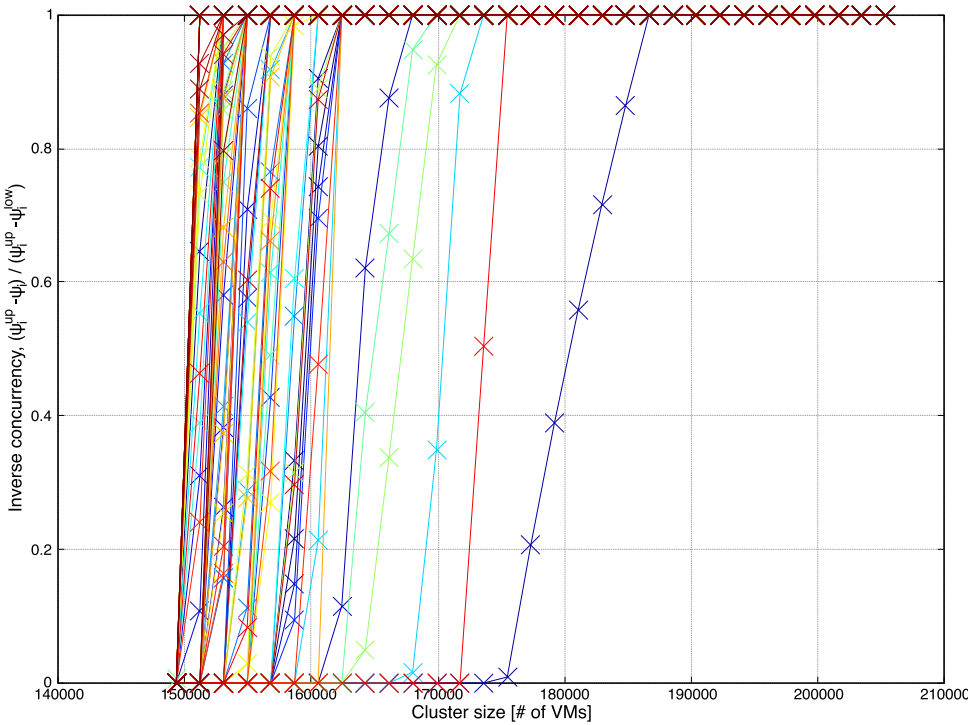
$\tilde{r}_i = r_i - r_i^{low}$



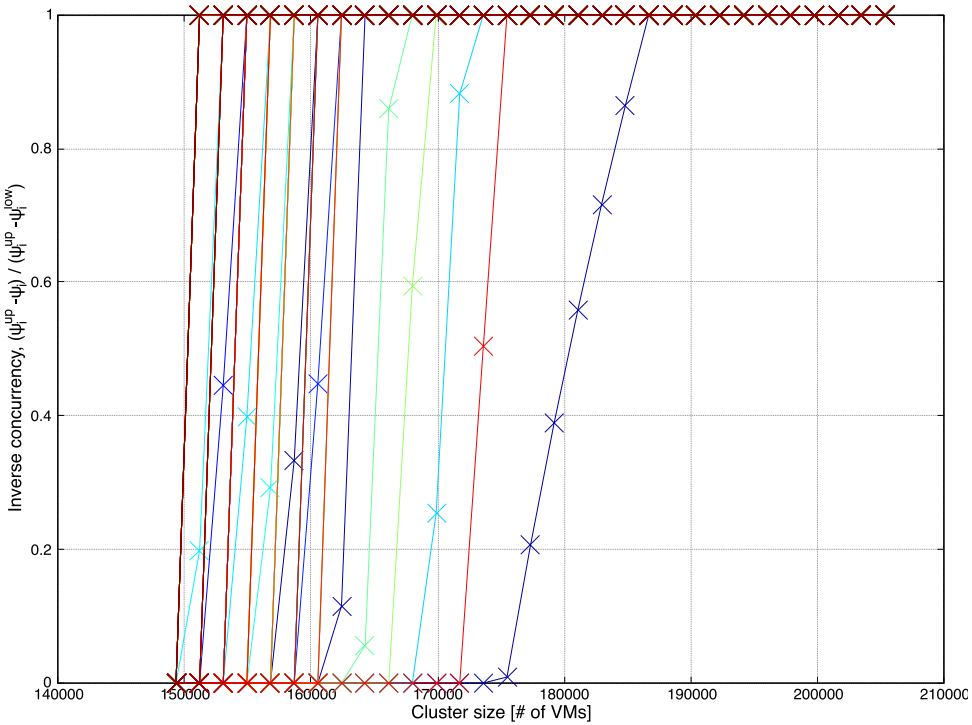
$\tilde{r}_i = r_i$

Total cost and penalties

Obtained Concurrency Levels



Centralized model



Closed form model

Decreasing cluster capacity experiment