ITALY
OpenInfra Days

OPENINFRA DAYS ITALY

Danilo Ardagna

MIlan, October 2, 2019

Optimal Resource Allocation of Cloud-Based Spark Applications

Organized by

IRIDEOS

Binario Etico

Under the patronage of

AGID Agenzia per l'Italia Digitale

Sponsored by

Mellanox TECHNOLOGIES

MESOSPHERE

gci SERVICE FACTORY

POLITECNICO MILANO 1863

Outline

- Motivations
- Target scenario and goals
- Performance models and Resource optimization
- Experimental Results
- Conclusions & Future Works

Motivations

- The big data paradigm is consolidating its central position in the industry, as well as in society at large
- Market growth from $130 billion in 2016 to $203 billion in 2020, with a CAGR of 11.9 %
- Cloud computing is an enabling cost-effective technology for big data
  - IDC estimates that by 2020 nearly 40% of big data analyses will be supported by public clouds

Target Scenario

- Big data applications: heterogeneous and irregular data access and computational patterns

+

- Cloud computing: offers flexibility, dynamically adjusting resources as needed

Develop intelligent resource management systems providing QoS guarantees to end-users and efficient use of resources

Target Problem

- Virtualization technologies provide means to setup a wide number of possible configurations that can be allocated for an application
  - Type of processing node, # cores, etc.

Which configuration should we choose to avoid under and overestimating resources?

Our Goals

How can we predict the execution time of an application running on a target configuration?

Predict execution time given an amount of resources available

Optimize computational resources given target deadline

Our Goals

How can we predict the execution time of an application running on a target configuration?

Predict execution time given an amount of resources available

Optimize computational resources given target deadline

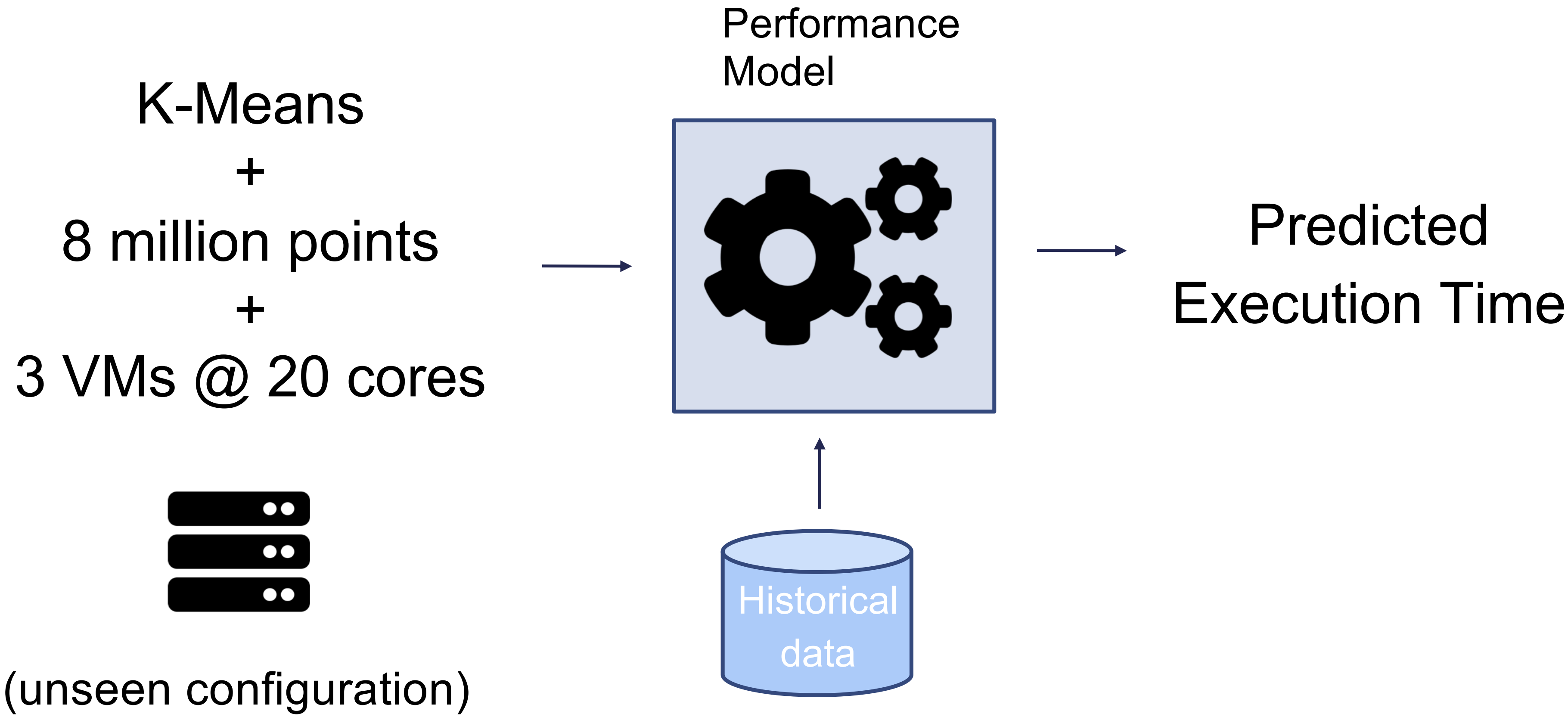
Our Goals

How can we predict the execution time of an application running on a target configuration?

Machine Learning	Analytical Models	Simulation

Optimize computational resources given target deadline

Our Approach: Performance profiling

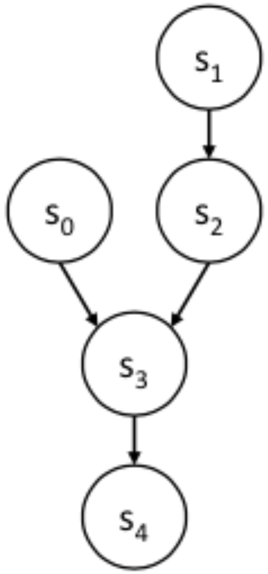


POLITECNICO
MILANO 1863

Focus: Apache Spark Applications

- Distributed general-purpose cluster-computing framework
  - Has support in the biggest cloud services (Amazon AWS, Azure, Google Cloud)

- Application execution represented by a DAG
  - Parallel stages
  - Parallel tasks execution in each stage



What is Machine Learning?

- Humans learn from **past experiences**
- A computer does not have "experiences"
  - A computer system learns from **data**, which represent some "**past experiences**" of an application domain
- Goal: learn a **target function** that can be used to **predict** the values of
  - a discrete class attribute, e.g., approve or not-approved, and high-risk or low risk (discrete world)
  - a continuous value, e.g., flight delays, cash at a bank branch/ATM (continuous setting)

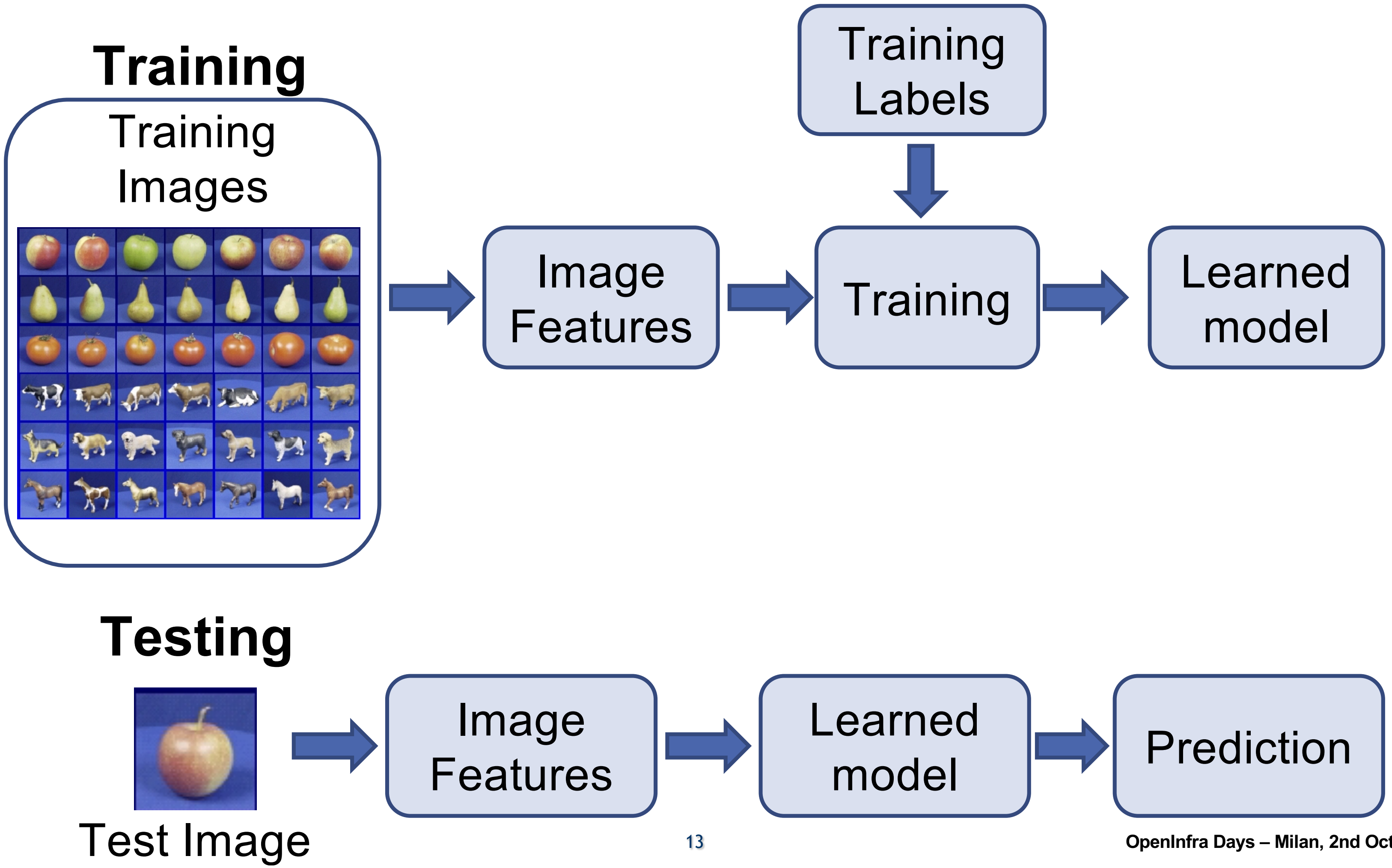
What is Machine Learning?

$$y = f(\mathbf{x})$$

output | prediction function | Image feature

- **Training:** given a *training set* of labeled examples $\{(\mathbf{x}_1,y_1), \ldots, (\mathbf{x}_N,y_N)\}$, estimate the prediction function f by minimizing the prediction error on the training set
- **Testing:** apply f to a never before seen *test example* $\mathbf{x}$ and output the predicted value $y = f(\mathbf{x})$

ML Steps


Training
Training Images
Image Features
Training Labels
Training
Learned model
Testing
Test Image
Image Features
Learned model
Prediction

Reference Solution

- ERNEST model (proposed by Spark creators)
  - Linear Regression with Non Negative Least Squares
  - Input features based on # cores and data size

Can we do better by exploring other regression techniques and/or input features?

Can we do better with Analytical techniques?

S. Venkataraman, Z. Yang, M. J. Franklin, B. Recht, and I. Stoica, "Ernest: Efficient performance prediction for large-scale advanced analytics." in *NSDI*, 2016, pp. 363–378.

Our Performance Approaches

- ML models compute very fast estimations with good approximations
  - Number of cores, tasks time and number per stage
  - Black and gray box models
  - Used by optimization methods to compute quickly initial solutions

- Approximate analytical techniques (*Lundstrom*) provide more accurate results but are slower
  - Distribution of the tasks execution time within individual stages
  - Task execution overlaps
  - Used at runtime under heavy load

- Discrete event simulator (*dagSim*) produces accurate results at the cost of longer execution times
  - Used for initial deployment, offline

D. Ardagna, E. Barbierato, A. Evangelinou, E. Gianniti, M. Gribaudo, T. B. M. Pinto, A. Guimarães, A. P. Couto da Silva, J. M. Almeida. Performance Prediction of Cloud-Based Big Data Applications. ICPE 2018 Proceedings. 192-199. Berlin, Germany.

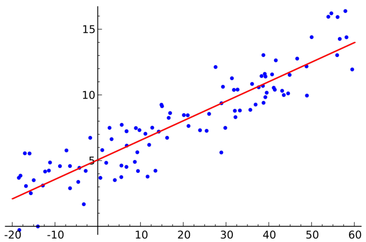
Our ML Approaches

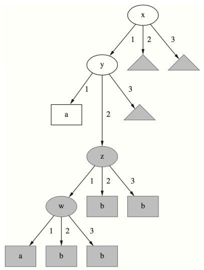
- Four algorithms with different properties:
  - LR: L1-regularized linear regression
  - DT: Decision tree
  - RF: Random forests
  - NN: Neural networks

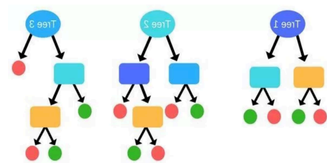
Linearity

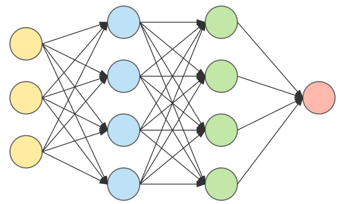
Interpretability

Non-linear and more complex relationships









Input Features

Model	Features
Ernest	- Ratio of data size to number of cores - Log of number of cores - Square root of ratio of data size to number of cores - Ratio of squared data size to number of cores

Input Features

Model	Features
Ernest	- Ratio of data size to number of cores - Log of number of cores - Square root of ratio of data size to number of cores - Ratio of squared data size to number of cores
Black box models	- Ratio of data size to number of cores - Log of number of cores - Data size - Number of cores - Number of TensorFlow cores (SparkDL only)

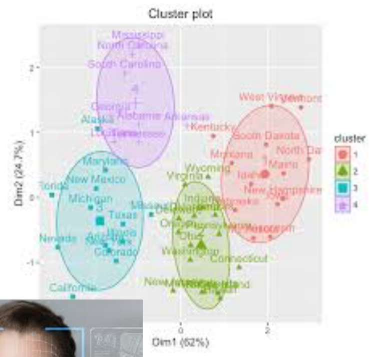
Information available *a priori*

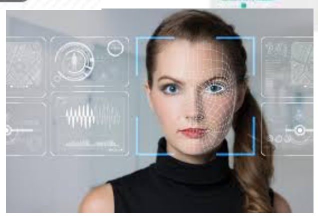
Input Features

Model	Features
Ernest	- Ratio of data size to number of cores - Log of number of cores - Square root of ratio of data size to number of cores - Ratio of squared data size to number of cores
Black box models	- Ratio of data size to number of cores - Log of number of cores - Data size - Number of cores - Number of TensorFlow cores (SparkDL only)
Gray box models	All black box models features and: - Number of tasks - Max/avg time over tasks - Max/avg shuffle time - Max/avg number of bytes transmitted between stages - Inverse of number of TensorFlow cores (SparkDL only)

Information available *only a posteriori:*
Use average values of training examples

Experimental Settings





- TPC-DS benchmark
- Sparkbench
- SparkDL: deep learning application based on DL pipelines

- Cluster configurations:
  - Public Microsoft Azure cloud
  - Private IBM Power8 cluster

- ML scenarios:
  - Core interpolation
  - Core interpolation + data extrapolation

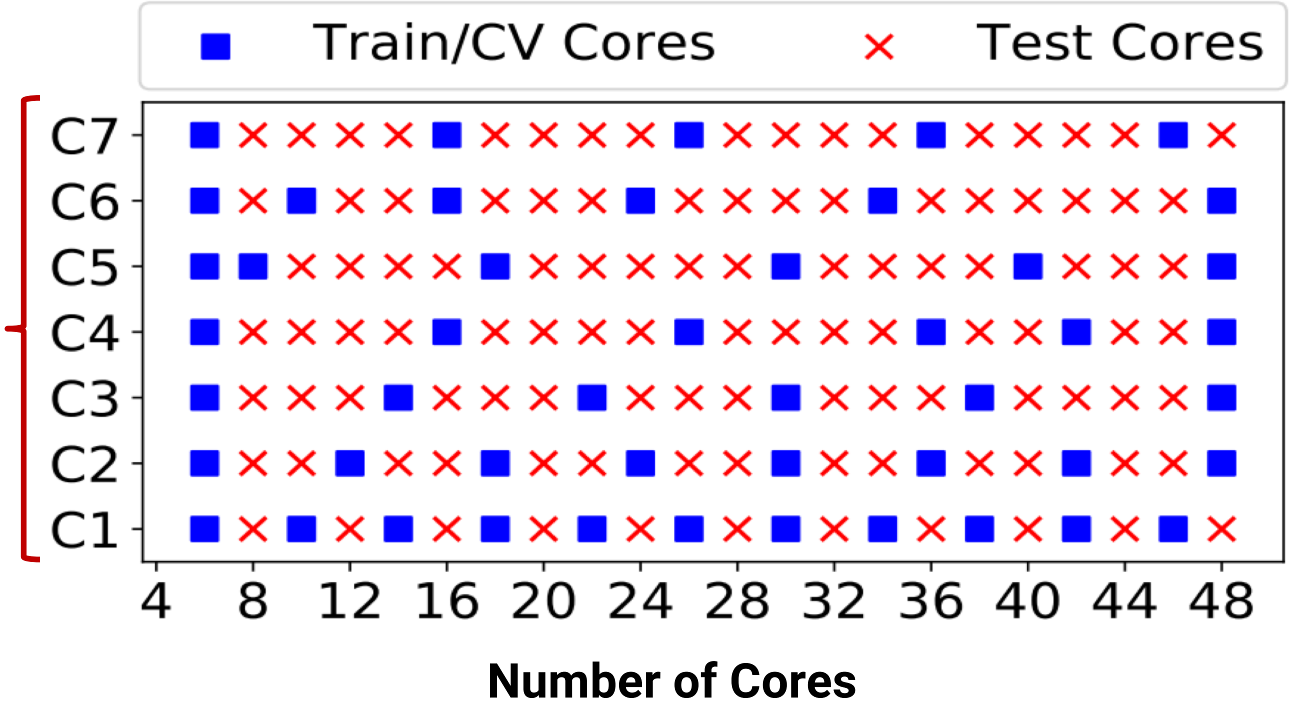
Microsoft Azure

A. Maros, F. Murai, A. P. Couto da Silva, J. M. Almeida, M. Lattuada, E. Gianniti, M. Hosseini, *D. Ardagna*. Machine Learning for Performance Prediction of Spark Cloud Applications. IEEE Cloud 2019 Proceedings. 99-106. Milan, Italy.

ML Core Interpolation: Training and Test Sets

K-means

7 cases defined by different splits of training and test sets



ML Core Interpolation and Data Extrapolation

Data sizes in training and test sets

Workload	Core Interpolation		Data Extrapolation	
	Training	Test	Training	Test
Query 26 [GB]	750	750	250, 750	1000
K-means [Rows]	15	15	5, 10, 15	20
SparkDL [Images]	1500	1500	1000, 1500	2000

ML Core Interpolation and Data Extrapolation

Data sizes in training and test sets

Workload	Core Interpolation		Data Extrapolation	
	Training	Test	Training	Test
Query 26 [GB]	750	750	250, 750	1000
K-means [Rows]	15	15	5, 10, 15	20
SparkDL [Images]	1500	1500	1000, 1500	2000

How accurate are predictions if we use different number of cores for training and and testing (but same data set size) ?

ML Core Interpolation and Data Extrapolation

POLITECNICO MILANO 1863

Data sizes in training and test sets

Workload	Core Interpolation		Data Extrapolation	
	Training	Test	Training	Test
Query 26 [GB]	750	750	250, 750	1000
K-means [Rows]	15	15	5, 10, 15	20
SparkDL [Images]	1500	1500	1000, 1500	2000

How accurate are predictions if we use different data set sizes and number of cores for training and and testing?

Q-26 Core Interpolation
(750 GB for training and test sets)

	Gray Box Models				Black Box Models				Ernest
	DT	LR	NN	RF	DT	LR	NN	RF	
C1	20.6	63.4	12.3	18.8	8.4	**1.0**	6.7	2.4	1.5
C2	16.7	72.6	16.1	19.9	7.9	**1.2**	20.1	6.0	1.6
C3	18.0	98.5	36.3	18.9	11.2	**1.2**	3.1	8.0	1.7
C4	21.7	300.6	18.9	27.2	12.9	**1.1**	4.4	9.7	1.6
C5	35.7	229.7	30.8	35.0	12.5	**1.2**	23.0	12.0	1.6
C6	27.0	414.1	26.3	32.3	8.9	**1.2**	5.4	6.9	1.7

Splits of training and test sets

New approaches

Reference model

POLITECNICO MILANO 1863

Q-26 Core Interpolation

(750 GB for training and test sets)

	Gray Box Models				Black Box Models			
	DT	LR	NN	RF	DT	LR	NN	RF
C1	20.6	63.4	12.3	18.8	8.4	**1.0**	6.7	2.4
C2	16.7	72.6	16.1	19.9	7.9	**1.2**	20.1	6.0
C3	18.0	98.5	36.3	18.9	11.2	**1.2**	3.1	8.0
C4	21.7	300.6	18.9	27.2	12.9	**1.1**	4.4	9.7
C5	35.7	229.7	30.8	35.0	12.5	**1.2**	23.0	12.0
C6	27.0	414.1	26.3	32.3	8.9	**1.2**	5.4	6.9

- Black box models outperform gray box solutions

Q-26 Core Interpolation
(750 GB for training and test sets)

	Gray Box Models				Black Box Models			
	DT	LR	NN	RF	DT	LR	NN	RF
C1	20.6	63.4	12.3	18.8	8.4	**1.0**	6.7	2.4
C2	16.7	72.6	16.1	19.9	7.9	**1.2**	20.1	6.0
C3	18.0	98.5	36.3	18.9	11.2	**1.2**	3.1	8.0
C4	21.7	300.6	18.9	27.2	12.9	**1.1**	4.4	9.7
C5	35.7	229.7	30.8	35.0	12.5	**1.2**	23.0	12.0
C6	27.0	414.1	26.3	32.3	8.9	**1.2**	5.4	6.9

- Simple black box LR approach is the best approach

POLITECNICO MILANO 1863

Q-26 Core Interpolation
(750 GB for training and test sets)

	Gray Box Models				Black Box Models				Ernest
	DT	LR	NN	RF	DT	LR	NN	RF	
C1	20.6	63.4	12.3	18.8	8.4	**1.0**	6.7	2.4	1.5
C2	16.7	72.6	16.1	19.9	7.9	**1.2**	20.1	6.0	1.6
C3	18.0	98.5	36.3	18.9	11.2	**1.2**	3.1	8.0	1.7
C4	21.7	300.6	18.9	27.2	12.9	**1.1**	4.4	9.7	1.6
C5	35.7	229.7	30.8	35.0	12.5	**1.2**	23.0	12.0	1.6
C6	27.0	414.1	26.3	32.3	8.9	**1.2**	5.4	6.9	1.7

- Simple black box LR approach is the best approach (a bit better than Ernest)

POLITECNICO
MILANO 1863

Q-26 Data Extrapolation
(250 and 750 GB for training; 1000 GB for testing)

	Gray Box Models				Black Box Models				Ernest
	DT	LR	NN	RF	DT	LR	NN	RF	
C1	38.2	28.6	7.0	39.6	15.6	**3.9**	9.9	19.4	7.5
C2	42.5	23.5	24.8	33.6	16.0	**4.0**	10.3	16.6	7.4
C3	39.0	36.7	11.2	37.2	21.1	**3.7**	7.8	19.0	7.3
C4	42.2	33.3	13.5	35.4	25.5	**4.0**	25.4	23.5	7.3
C5	32.5	12.4	9.8	35.1	19.0	**4.4**	31.8	17.3	7.6
C6	37.0	24.8	24.8	37.3	17.3	**4.9**	18.1	19.5	8.0

- Similar conclusions

K-means Core Interpolation
(15 Million points for training and test sets)

	Gray Box Models				Black Box Models				Ernest
	DT	LR	NN	RF	DT	LR	NN	RF	
C1	27.7	184.3	77.9	24.8	16.0	50.8	38.3	**5.0**	126.7
C2	28.7	225.0	109.6	54.7	16.9	46.3	18.1	**13.7**	148.1
C3	22.9	278.3	435.7	26.0	18.5	42.1	10.3	**14.0**	161.3
C4	33.8	300.6	445.1	26.3	21.4	41.9	23.6	**14.2**	176.5
C5	27.1	543.4	1146.1	22.5	22.9	42.3	31.3	**19.3**	187.0
C6	33.9	414.1	170.8	91.3	15.2	48.2	**10.6**	12.0	159.9
C7	22.6	363.1	626.0	31.3	17.4	41.8	33.2	**14.8**	178.1

K-means Core Interpolation
(15 Million points for training and test sets)

	Gray Box Models				Black Box Models				Ernest
	DT	LR	NN	RF	DT	LR	NN	RF	
C1	27.7	184.3	77.9	24.8	16.0	50.8	38.3	**5.0**	126.7
C2	28.7	225.0	109.6	54.7	16.9	46.3	18.1	**13.7**	148.1
C3	22.9	278.3	435.7	26.0	18.5	42.1	10.3	**14.0**	161.3
C4	33.8	300.6	445.1	26.3	21.4	41.9	23.6	**14.2**	176.5
C5	27.1	543.4	1146.1	22.5	22.9	42.3	31.3	**19.3**	187.0
C6	33.9	414.1	170.8	91.3	15.2	48.2	**10.6**	12.0	159.9
C7	22.6	363.1	626.0	31.3	17.4	41.8	33.2	**14.8**	178.1

- Ernest is not able to capture greater complexity of the workload

K-means Core Interpolation

(15 Million points for training and test sets)

	Gray Box Models				Black Box Models			
	DT	LR	NN	RF	DT	LR	NN	RF
C1	27.7	184.3	77.9	24.8	16.0	50.8	38.3	**5.0**
C2	28.7	225.0	109.6	54.7	16.9	46.3	18.1	**13.7**
C3	22.9	278.3	435.7	26.0	18.5	42.1	10.3	**14.0**
C4	33.8	300.6	445.1	26.3	21.4	41.9	23.6	**14.2**
C5	27.1	543.4	1146.1	22.5	22.9	42.3	31.3	**19.3**
C6	33.9	414.1	170.8	91.3	15.2	48.2	**10.6**	12.0
C7	22.6	363.1	626.0	31.3	17.4	41.8	33.2	**14.8**

- Once again, black box models outperform gray box solutions

POLITECNICO
MILANO 1863

K-means Core Interpolation

(15 Million points for training and test sets)

	Gray Box Models				Black Box Models				Ernest
	DT	LR	NN	RF	DT	LR	NN	RF	
C1	27.7	184.3	77.9	24.8	16.0	50.8	38.3	**5.0**	126.7
C2	28.7	225.0	109.6	54.7	16.9	46.3	18.1	**13.7**	148.1
C3	22.9	278.3	435.7	26.0	18.5	42.1	10.3	**14.0**	161.3
C4	33.8	300.6	445.1	26.3	21.4	41.9	23.6	**14.2**	176.5
C5	27.1	543.4	1146.1	22.5	22.9	42.3	31.3	**19.3**	187.0
C6	33.9	414.1	170.8	91.3	15.2	48.2	**10.6**	12.0	159.9
C7	22.6	363.1	626.0	31.3	17.4	41.8	33.2	**14.8**	178.1

- Black box RF is the best approach, capturing non-linear relationships

K-means Data Extrapolation
(5, 10, 15 Million points for training; 20 Million for testing)

	Gray Box Models				Black Box Models				Ernest
	DT	LR	NN	RF	DT	LR	NN	RF	
C1	24.0	20.2	22.9	27.1	19.8	40.0	32.5	**12.5**	93.4
C2	35.6	16.0	121.3	20.7	**14.5**	39.6	29.1	16.5	107.8
C3	66.7	31.9	134.1	32.6	**14.4**	41.4	28.0	16.5	119.6
C4	28.8	24.2	118.9	25.8	**13.3**	31.8	68.4	16.0	127.9
C5	42.0	34.0	27.5	26.7	**15.3**	25.7	60.0	19.5	121.4
C6	75.7	37.7	37.1	24.2	**11.0**	52.6	26.7	14.9	109.8
C7	32.6	43.1	148.1	42.6	20.8	34.5	96.4	**17.9**	129.5

- Ernest performs poorly again

K-means Data Extrapolation

(5, 10, 15 Million points for training; 20 Million for testing)

	Gray Box Models				Black Box Models				Ernest
	DT	LR	NN	RF	DT	LR	NN	RF	
C1	24.0	20.2	22.9	27.1	19.8	40.0	32.5	**12.5**	93.4
C2	35.6	16.0	121.3	20.7	**14.5**	39.6	29.1	16.5	107.8
C3	66.7	31.9	134.1	32.6	**14.4**	41.4	28.0	16.5	119.6
C4	28.8	24.2	118.9	25.8	**13.3**	31.8	68.4	16.0	127.9
C5	42.0	34.0	27.5	26.7	**15.3**	25.7	60.0	19.5	121.4
C6	75.7	37.7	37.1	24.2	**11.0**	52.6	26.7	14.9	109.8
C7	32.6	43.1	148.1	42.6	20.8	34.5	96.4	**17.9**	129.5

- Overall: black box DT and RF best approaches

POLITECNICO
MILANO 1863

SparkDL Core Interpolation
(1500 images for training and test sets)

	Gray Box Models				Black Box Models				Ernest
	DT	LR	NN	RF	DT	LR	NN	RF	
C1	5.2	28.7	4.7	4.5	5.1	7.3	**4.6**	5.1	10.5
C2	5.8	5.7	13.3	4.8	**5.5**	6.2	8.6	5.7	6.3
C3	8.9	7.5	5.4	6.0	5.5	5.5	5.7	**4.9**	5.7

- Black box models are usually better than gray box approaches

SparkDL Data Extrapolation

(1000 and 1500 images for training; 2500 for testing)

	Gray Box Models				Black Box Models				Ernest
	DT	LR	NN	RF	DT	LR	NN	RF	
C1	37.0	10.7	25.7	34.7	35.9	**7.5**	34.1	36.7	43.5
C2	36.3	10.0	31.4	37.0	41.5	**7.6**	15.3	41.9	37.4
C3	36.9	14.7	9.9	34.5	41.0	**7.8**	33.3	41.1	36.8

- Black box LR is the overall best approach

Lundstrom and dagSim results

Q-26

Cores	Lundstrom	dagSim
12	**4.4**	5.5
16	**6.7**	9.7
20	**9.1**	11.8
24	**11.0**	11.9
28	**11.7**	16.2
32	**13.5**	14.7
36	16.3	**5.2**
40	17.6	**6.0**
44	19.3	**3.5**
48	20.7	**-0.1**
52	17.6	**-0.4**

K-means

Cores	Data set size (GB)	Lundstrom	dagSim
24	8	**17.3**	23.6
24	48	**5.0**	-6.5
24	96	**1.9**	8.5
48	8	**18.1**	22.1
48	48	**8.3**	-12.4
48	96	**3.6**	-25.6

Query	Quartile	dagSim [s]	Real [s]	ε_r [%]
Q26	Q_1	492.496	515.449	4.66
Q26	Q_2	495.077	537.436	8.56
Q26	Q_3	497.800	597.302	19.99
Q52	Q_1	509.974	509.810	0.03
Q52	Q_2	511.676	515.547	0.76
Q52	Q_3	513.454	520.582	1.39

Our Goals

How can we predict the execution time of an application running on a target cloud configuration?

Predict execution time given an amount of resources available

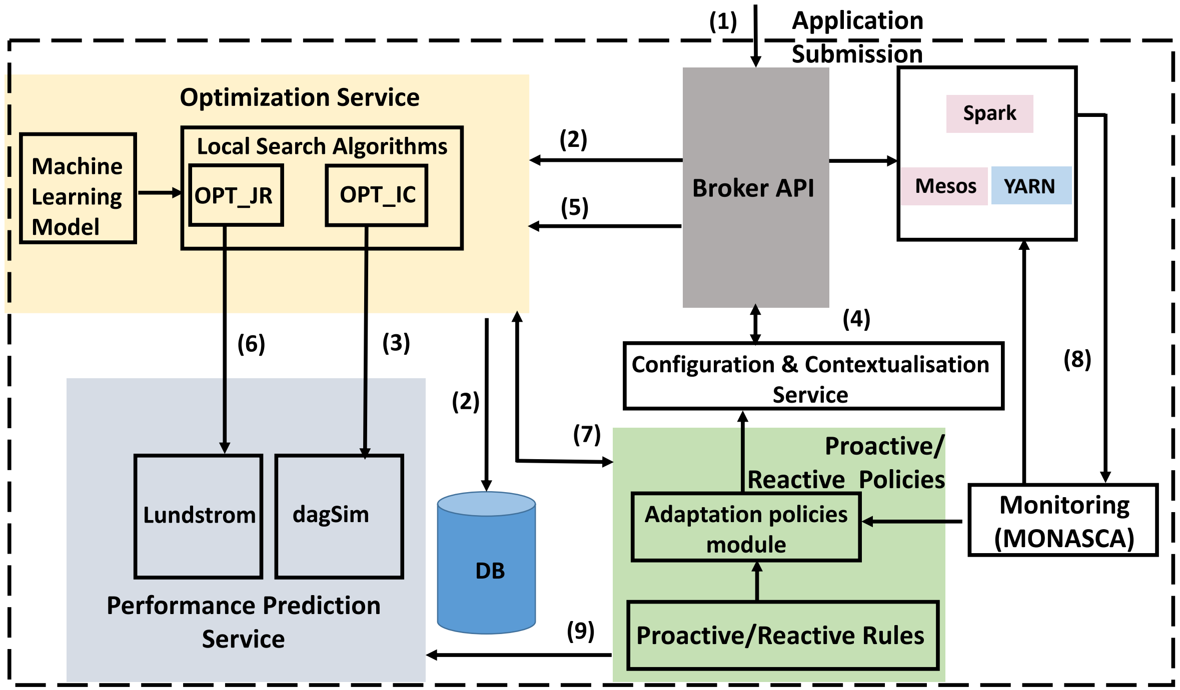
Optimal deployment

Rebalancing under heavy load

Optimal deployment

- *Identifying the Initial Configuration*
  1. Identify the minimum number of VMs for an application to fulfill its deadline
  2. Periodically estimate application capacity according to its progress

- *Resource Rebalancing under Heavy Load*
  1. Prioritize resources to hard deadline applications and reallocate the residual capacity among soft deadline ones
  2. Minimize weighted tardiness



- MINLP
- Local search

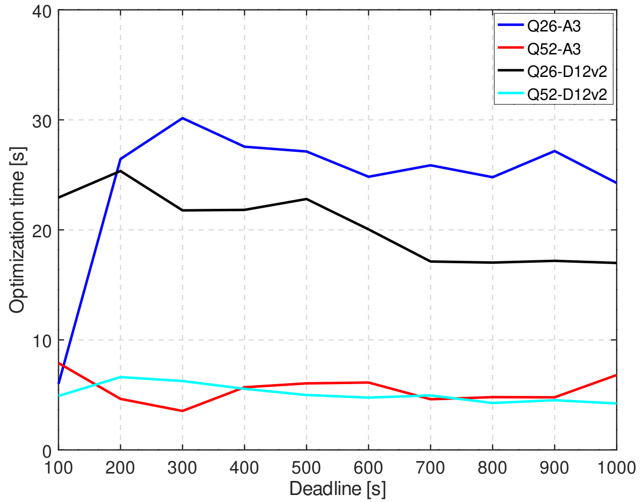
Experimental results – Initial configuration

Q-26

D [s]	c^{r}	c^{e}	ϵ_e [%]	c^{p}	ϵ_p [%]
661	12	16	−33.33	16	−33.33
553	16	16	0.00	16	0.00
454	20	20	0.00	20	0.00
386	24	24	0.00	24	0.00
354	28	28	0.00	24	14.29
304	32	32	0.00	28	12.50
244	36	40	−11.11	36	0.00
225	40	44	−10.00	40	0.00
199	44	48	−9.09	44	0.00

K-means

D[s]	c^{r}	c^{e}	ϵ_e [%]	c^{p}	ϵ_p [%]
4,386	8	4	50.00	8	0.00
1,934	12	6	50.00	12	0.00
718	16	18	−12.50	18	−12.50
428	20	28	−40.00	22	−10.00
362	24	32	−33.33	24	0.00
317	28	38	−35.71	28	0.00
284	32	42	−31.25	42	−31.25
259	36	46	−27.78	46	−27.78
240	40	50	−25.00	48	−20.00
229	44	52	−18.18	48	−9.09



POLITECNICO
MILANO 1863

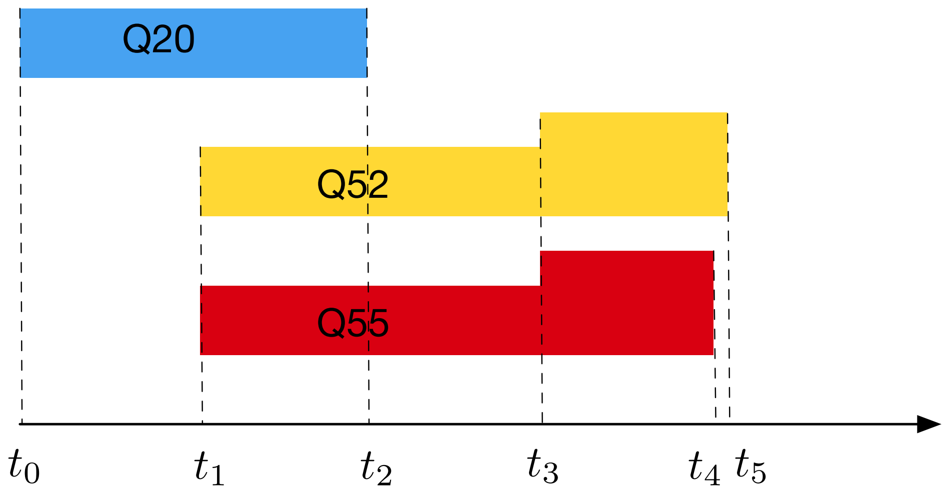
Experimental results – Resource rebalancing

Test	Size	Of_{real} [s]	Of_e [s]	ϵ_e [%]	Of_j [s]	ϵ_j [%]
Test1	small	666396	666396	0.00	666396	0.00
Test1	large	446025	494255	10.81	466654	4.62
Test2	small	0	0	0.00	0.00	0.00
Test2	large	0	214608	$+\infty$	0.00	0.00
Test3	small	3115221	3505957	12.54	115221	0.00
Test3	large	1891436	2188250	15.69	966587	3.97
Test4	small	1340295	1866892	39.29	340295	0.00
Test4	large	606460	1476138	143.40	756368	24.72
Test5	small	135637	253209	86.68	135637	0.00
Test5	large	820166	1263198	54.02	948275	15.62
Test6	-	885533	2699995	204.90	239210	39.94

with 4 threads.

		ST	MT(2)		MT(4)	
Test	Size	Time [s]	Time [s]	SU	Time [s]	SU
Test1	small	30.13	17.09	1.76	16.12	1.87
Test1	large	39.01	28.90	1.35	21.14	1.85
Test2	small	27.01	14.64	1.84	13.00	2.08
Test2	large	36.00	26.74	1.35	19.12	1.88
Test3	small	32.01	19.10	1.78	17.21	1.86
Test3	large	42.14	31.19	1.73	24.90	1.69
Test4	small	29.00	16.26	1.72	15.15	1.91
Test4	large	39.13	22.68	1.63	20.13	1.94
Test5	small	48.10	27.90	1.67	22.23	2.16
Test5	large	52.02	32.01	1.63	27.18	1.91
Test6	-	82.36	49.34	1.67	38.34	2.15

MS Azure deployment



Real system:
21% gap

POLITECNICO
MILANO 1863

Conclusions and Future Work

- Performance models and online resource allocation of Spark big data applications

- Average percentage error in computing the minimum capacity is around 7% while the average percentage error in re-balancing about 12%

- Resource provisioning of continuous applications

POLITECNICO
MILANO 1863

Acknowledgements

Colleagues at Polimi & Universidad Federal de Minas Gerais:
Marco Lattuada, Eugenio Gianniti, Marco Gribaudo, Enrico Barbierato,
Marjan Hosseini, Alexandre Maros, Fabrício Murai,
Ana Paula Couto da Silva, Jussara M. Almeida

EUBRA BIGSEA

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

ATMOSPHERE

Adaptive, Trustworthy, Manageable, Orchestrated, Secure Privacy-assuring Hybrid, Ecosystem for REsilient Cloud Computing

European and Brazilian Research Innovation Action projects, within the H2020 program, in the field of Cloud Computing

POLITECNICO
MILANO 1863

Thanks for your attention!