TRANSFERRING PARETO FRONTIERS ACROSS HETEROGENEOUS HARDWARE ENVIRONMENTS

4/20/2020

Authors: Pavel Valov, Jianmei Guo, Krzysztof Czarnecki

Presented by: Pavel Valov

David R. Cheriton School of Computer Science

UNIVERSITY OF WATERLOO

BASIC TERMINOLOGY

- Software systems provide configuration options
- User-relevant configuration options are called features
- Features influence functional properties (algorithms, behavior, etc.)
- Features influence non-functional properties (memory, performance)
- Unique feature selection defines a particular system configuration
- Each configuration has corresponding property values (e.g. memory)
- Actual property values vary across configurations
- Property values of configurations vary across hardware

SAMPLE OF MEASURED CONFIGURATIONS OF X264

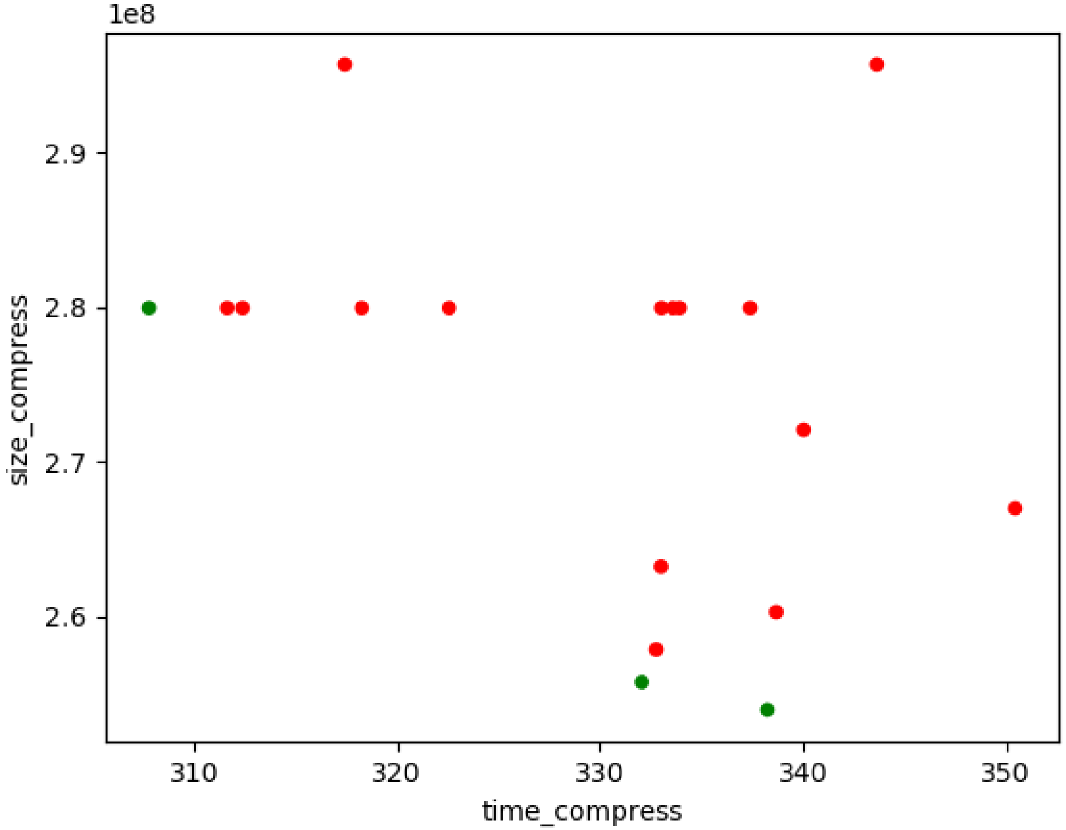
	Configuration Features								Properties on Server BscA0-2660		Properties on Server StdF1-2673v3	
	b-adapt	me	no-mbtree	no-scenecut	rc-lookahead	ref	subme	trellis	Compression Time (s)	Compression Size (B)	Compression Time (s)	Compression Size (B)
Randomly sampled configurations	2	0	1	1	0	16	9	2	515.38	4169610	88.83	4169610
	0	0	1	0	0	1	9	0	189.51	4007476	29.20	4007476
	2	0	1	1	60	1	0	2	63.17	5790676	10.50	5790676
	2	1	1	0	60	1	0	2	82.08	5911678	14.95	5911678
	2	1	0	1	60	1	0	0	66.84	3657516	11.82	3657516
	2	0	1	1	0	16	0	2	143.13	5739339	24.20	5739339
	2	0	1	0	0	1	0	0	57.13	5683517	9.25	5683517
	0	0	0	1	60	1	9	2	187.24	3067933	33.26	3068040
	2	0	0	1	0	1	0	2	62.75	5790676	10.58	5790676
	2	0	1	1	60	1	9	2	239.67	4194374	43.31	4194374

PARETO OPTIMALITY

- Configuration is Pareto-optimal if:
  - No other configuration can improve any system property (e.g. system performance) …
  - … without degrading some other system property (e.g. memory consumption)
- Each Pareto configuration is optimal in its own specific way
- Each Pareto configuration provides trade-off between properties

PARETO FRONTIER

bzip2 configuration space,
measured on Microsoft Azure cloud server



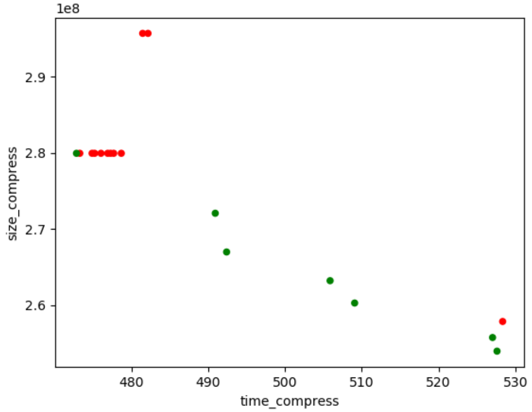
Compressed size of benchmarking file

Compression time of benchmarking file

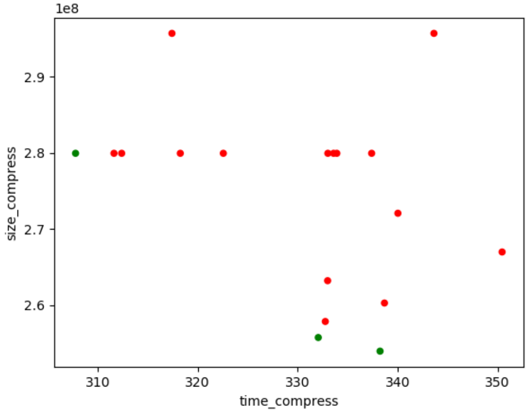
UNIVERSITY OF WATERLOO

PARETO FRONTIERS OF BZIP2

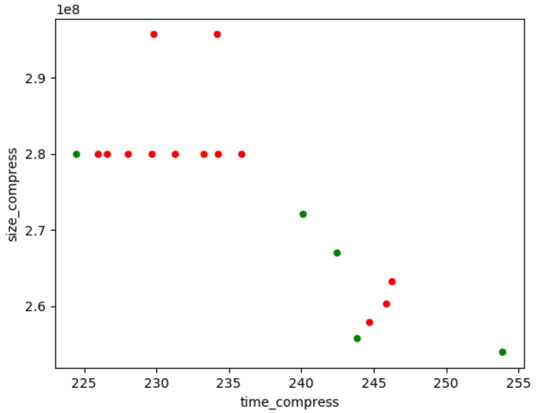
BscA0-2660



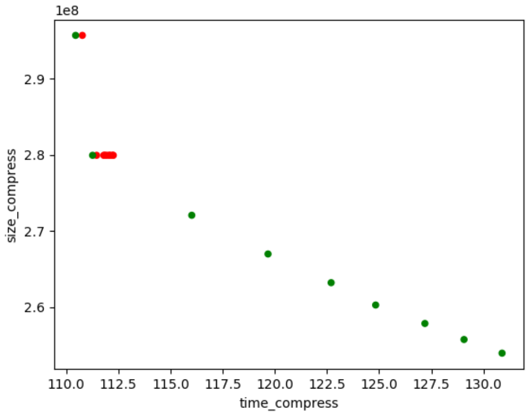
BscA2-2673v3



StdA1-2660



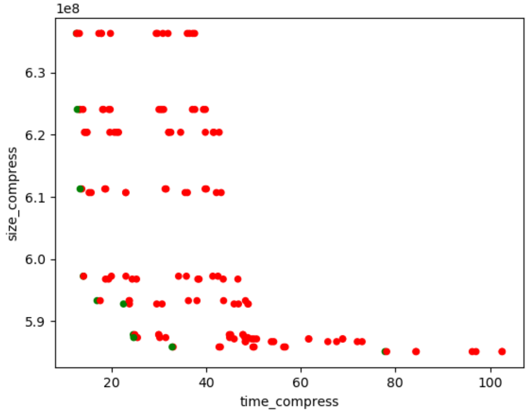
StdD2v3-2673v3



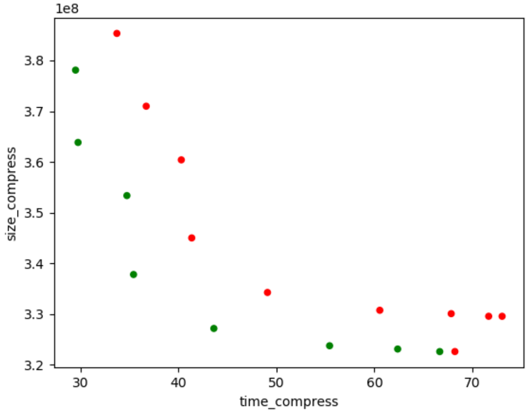
UNIVERSITY OF WATERLOO

PARETO FRONTIERS OF OTHER SYSTEMS

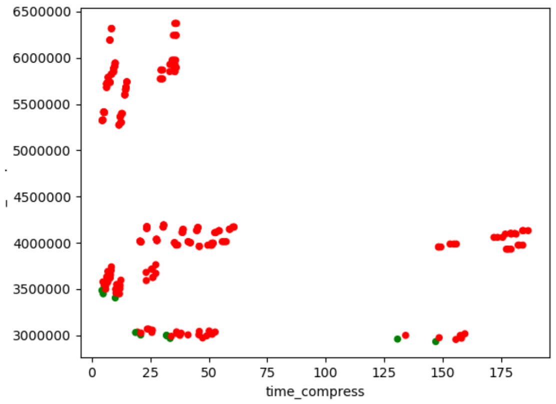
FLAC



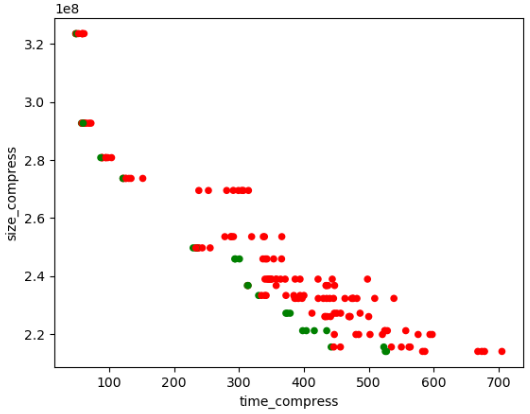
GZIP



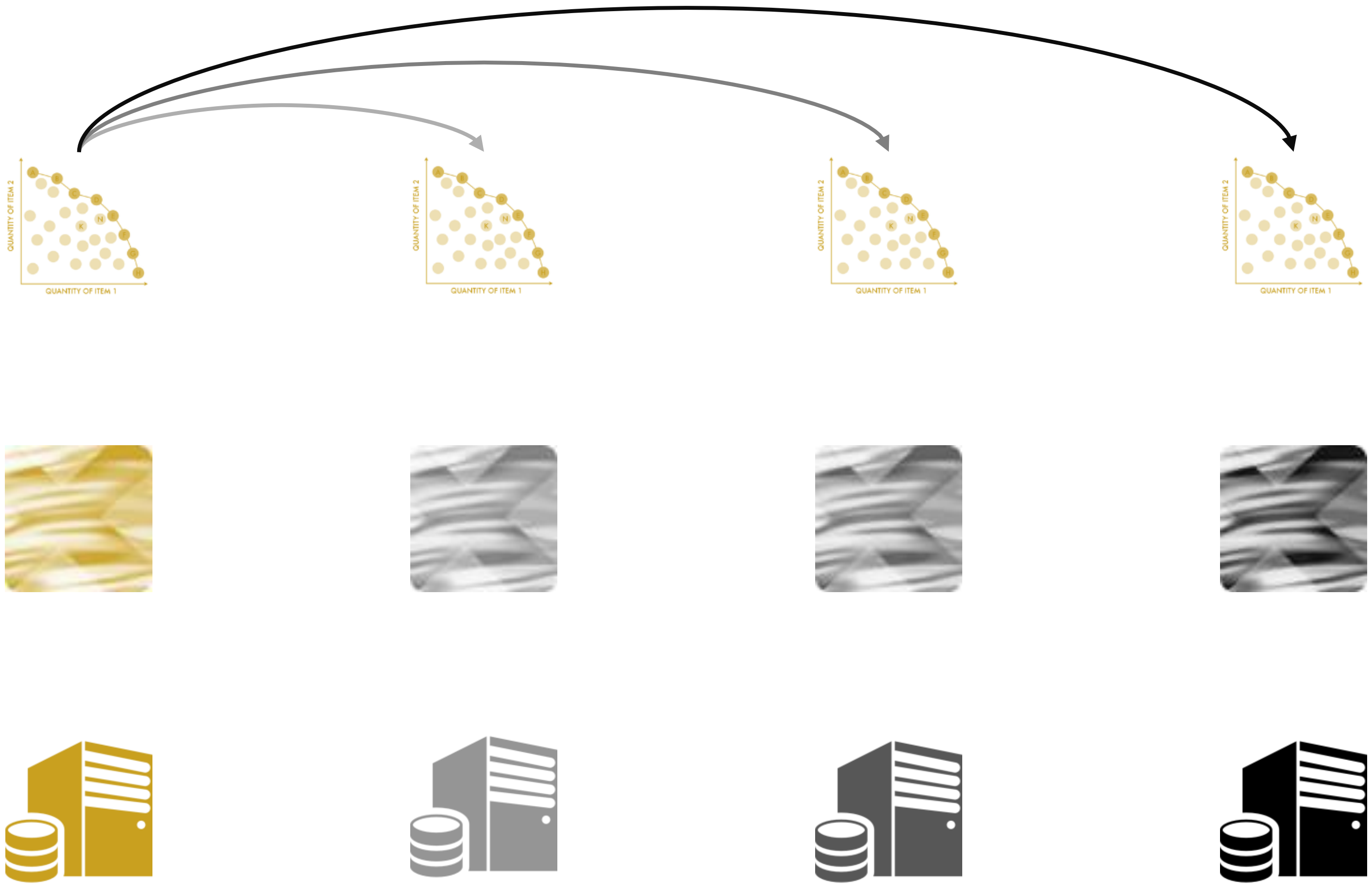
x264



XZ



PROBLEM STATEMENT



Transfer Pareto Frontiers across heterogeneous hardware platforms?

Pareto Frontiers
(of optimal system configurations)

x264
(configurable software system)

Microsoft Azure Servers
(heterogeneous hardware cluster)

CHALLENGES

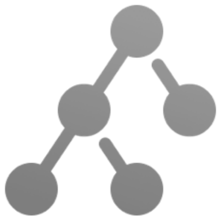
- Problem of limited information:
    - Configuration space grows exponentially
    - Benchmarking might have a limited budget
    - Benchmarking of configurations is time-consuming
- Problem of heterogeneous hardware:
    - Benchmarking results are irrelevant across heterogeneous hardware
    - Additional cross-platform benchmarking is required

PROBLEM OF LIMITED INFORMATION

Features				Prop
#1	#2	…	#n	
0	0	…	0	?
1	1	…	1	?
…	…	…	…	…
1	1	…	0	?

Unmeasured configurations

Property to be approximated



Features				Prop
#1	#2	…	#n	
1	0	…	0	35
0	1	…	1	30
…	…	…	…	…
0	0	…	1	40

Measured configurations

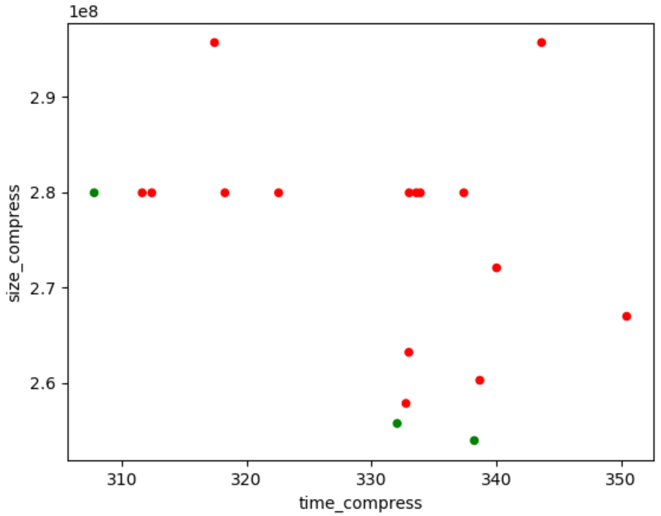
Features				Prop
#1	#2	…	#n	
0	0	…	0	25
1	1	…	1	50
…	…	…	…	…
1	1	…	0	45

Approximated configurations

Property is approximated

MULTIPLE PROPERTIES

bzip2 Pareto frontier



Compressed size predictor

Compression time predictor

Approximate Pareto frontier by individually predicting system properties

PARETO FRONTIER APPROXIMATION

Predict all properties &
combine approximated Pareto frontiers

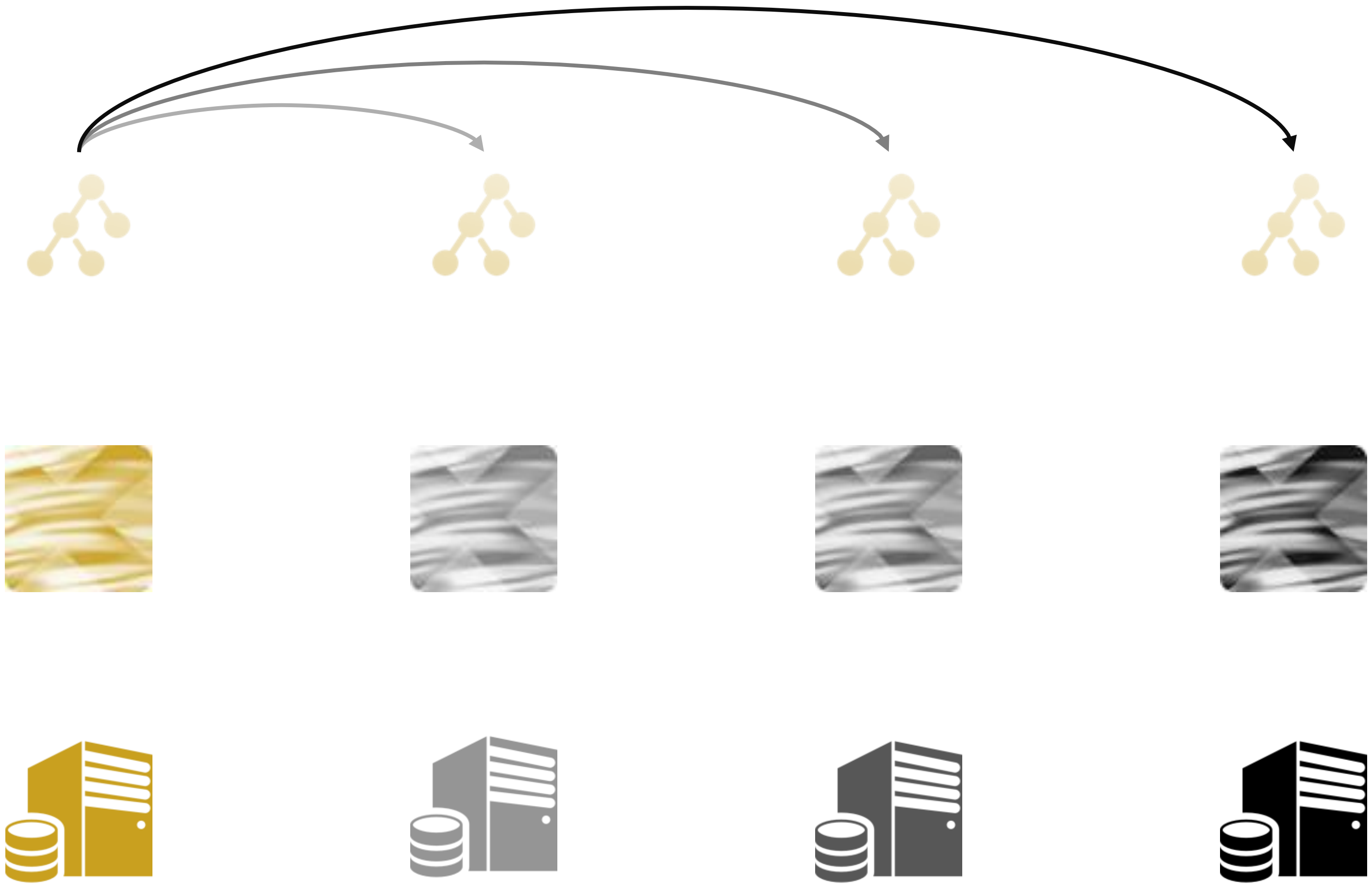
Compression time predictor
(Property-B predictor)

Compression time predictor
(Property-A predictor)

x264
(configurable software system)

Microsoft Azure Servers
(heterogeneous hardware cluster)

PROBLEM OF HETEROGENEOUS HARDWARE



1. Train property transferrers
2. Transfer property predictors

Regression Trees
(property prediction models)

x264
(configurable software system)

Microsoft Azure Servers
(heterogeneous hardware cluster)

TRAINING PROPERTY TRANSFERRERS

Features				Servers	
#1	#2	…	#n	Source	Destin.
0	0	…	0	20	?
1	1	…	1	15	?
…	…	…	…	…	…
1	1	…	0	25	?

Configurations:

- measured on source server
- unmeasured on destination serve

Features				Servers	
#1	#2	…	#n	Source	Destin.
1	0	…	0	20	30
0	1	…	0	15	25
…	…	…	…	…	…
0	0	…	1	25	35

Training Configurations:

- measured on source server
- measured on destination server

Features				Servers	
#1	#2	…	#n	Source	Destin.
0	0	…	0	20	30
1	1	…	1	15	25
…	…	…	…	…	…
1	1	…	0	25	35

Configurations:

- measured on source server
- predicted on destination server

UNIVERSITY OF WATERLOO

PARETO FRONTIER TRANSFERRING

Predict all properties &
combine approximated Pareto frontiers

Compression time predictor
(Property-B predictor)

Compression time predictor
(Property-A predictor)

x264
(configurable software system)

Microsoft Azure Servers
(heterogeneous hardware cluster)

REQUIREMENTS

- Pragmatic methodology for real-world scenarios
- Practitioner has:
  1. No control over sampling of configurations (pseudo-random sampling)
  2. Limited benchmarking of configurations (different sample sizes)
  3. No source code of configurable system (black-box approach)
- Approach should:
  1. Produce valid models with minimal amount of data
  2. Work in a completely automatic fashion
  3. Visualize models and results

HARDWARE ENVIRONMENTS

Table 1: Summary of all Azure-based virtual machines; CRS – number of CPU cores that are specifically allocated to the virtual machine, RAM – amount of RAM allocated to the VM (GiB), RPC – amount of RAM allocated per CPU core of the VM (GiB), STR – amount of storage allocated to the VM (GiB), STP – storage type allocated to the VM

General Server Info			Server Architecture						Benchmarked Systems				
Name	Azure Type	Deployment Region	CPU Model Name	CRS	RAM	RPC	STR	STP	BZIP2	GZIP	XZ	FLAC	X264
BscA0-4171HE	BasicA0	South Brazil	AMD Opteron 4171 HE	1	0.75	0.75	20	HDD		✓			✓
BscA0-2660	BasicA0	East Japan	Intel Xeon E5-2660	1	0.75	0.75	20	HDD	✓	✓		✓	✓
BscA0-2673v3	BasicA0	West US	Intel Xeon E5-2673 v3	1	0.75	0.75	20	HDD	✓	✓		✓	✓
BscA1-4171HE	BasicA1	South Brazil	AMD Opteron 4171 HE	1	1.75	1.75	40	HDD	✓	✓		✓	✓
BscA1-2660	BasicA1	East Japan	Intel Xeon E5-2660	1	1.75	1.75	40	HDD	✓	✓	✓	✓	✓
BscA1-2673v3	BasicA1	West US	Intel Xeon E5-2673 v3	1	1.75	1.75	40	HDD	✓	✓	✓	✓	✓
BscA2-4171HE	BasicA2	South Brazil	AMD Opteron 4171 HE	2	3.5	1.75	60	HDD	✓	✓		✓	✓
BscA2-2660	BasicA2	East Japan	Intel Xeon E5-2660	2	3.5	1.75	60	HDD	✓	✓	✓	✓	✓
BscA2-2673v3	BasicA2	West US	Intel Xeon E5-2673 v3	2	3.5	1.75	60	HDD	✓	✓	✓	✓	✓
StdA0-2673v3	StandardA0	Central Canada	Intel Xeon E5-2673 v3	1	0.75	0.75	20	SSD	✓	✓		✓	✓
StdA0-2660	StandardA0	East Japan	Intel Xeon E5-2660	1	0.75	0.75	20	SSD	✓	✓		✓	✓
StdA1-2673v3	StandardA1	Central Canada	Intel Xeon E5-2673 v3	1	1.75	1.75	70	SSD	✓	✓	✓	✓	✓
StdA1-2660	StandardA1	East Japan	Intel Xeon E5-2660	1	1.75	1.75	70	SSD	✓	✓		✓	✓
StdA1v2-2660	StandardA1 v2	South Central US	Intel Xeon E5-2660	1	2	2	10	SSD	✓		✓	✓	✓
StdA1v2-2673v3	StandardA1 v2	West Central US	Intel Xeon E5-2673 v3	1	2	2	10	SSD	✓	✓	✓	✓	✓
StdA2-2673v3	StandardA2	Central Canada	Intel Xeon E5-2673 v3	2	3.5	1.75	135	SSD	✓	✓	✓	✓	✓
StdA2-2660	StandardA2	East Japan	Intel Xeon E5-2660	2	3.5	1.75	135	SSD	✓	✓	✓	✓	✓
StdA2v2-2660	StandardA2 v2	South Central US	Intel Xeon E5-2673 v3	2	4	2	20	SSD	✓		✓	✓	✓
StdA2v2-2673v3	StandardA2 v2	West Central US	Intel Xeon E5-2673 v3	2	4	2	20	SSD	✓	✓	✓	✓	✓
StdD1-2660	StandardD1	South East Australia	Intel Xeon E5-2660	1	3.5	3.5	50	SSD	✓	✓	✓	✓	✓
StdD1v2-2673v3	StandardD1 v2	Central US	Intel Xeon E5-2673 v3	1	3.5	3.5	50	SSD	✓	✓	✓	✓	✓
StdD1v2-2673v4	StandardD1 v2	South India	Intel Xeon E5-2673 v4	1	3.5	3.5	50	SSD	✓				
StdD2-2660	StandardD2	South East Australia	Intel Xeon E5-2660	2	7	3.5	100	SSD	✓	✓	✓	✓	✓
StdD2v2-2673v3	StandardD2 v2	Central US	Intel Xeon E5-2673 v3	2	7	3.5	100	SSD	✓	✓	✓	✓	✓
StdD2v2-2673v4	StandardD2 v2	South India	Intel Xeon E5-2673 v4	2	7	3.5	100	SSD	✓				
StdD2v3-2673v4	StandardD2 v3	South East Australia	Intel Xeon E5-2673 v4	2	8	4	50	SSD	✓	✓	✓	✓	✓
StdD2v3-2673v3	StandardD2 v3	South East Asia	Intel Xeon E5-2673 v3	2	8	4	50	SSD	✓	✓	✓	✓	✓
StdE2v3-2673v4	StandardE2 v3	West Europe	Intel Xeon E5-2673 v4	2	16	8	50	SSD	✓	✓	✓	✓	✓
StdF1-2673v3	StandardF1	East US	Intel Xeon E5-2673 v3	1	2	2	16	SSD	✓	✓	✓	✓	✓
StdF1-2673v4	StandardF1	South India	Intel Xeon E5-2673 v4	1	2	2	16	SSD	✓				
StdF2-2673v3	StandardF2	East US	Intel Xeon E5-2673 v3	2	4	2	32	SSD	✓	✓	✓	✓	✓
StdF2-2673v4	StandardF2	South India	Intel Xeon E5-2673 v4	2	4	2	32	SSD	✓				
StdF2sv2-8168	StandardF2 v2	West US 2	Intel Xeon Platinum 8168	2	4	2	16	SSD	✓	✓	✓	✓	✓
StdG1-2698Bv3	StandardG1	East US 2	Intel Xeon E5-2698B v3	2	28	14	384	SSD	✓	✓	✓	✓	✓

HARDWARE CPUS

Table 2: Summary of all CPUs used in the experiment; TCH – technology node (nm), FRQ – CPU core frequency (MHz), FBS – front-side bus frequency (MHz), CM – clock multiplier, CRS – number of CPU cores, TRD – number of threads, L1i – Level 1 instruction cache size (KB), L1d – Level 1 data cache size (KB), L2 – Level 2 cache size, L3 – Level 3 cache size, PMC – PassMark CPU benchmark score (higher is better), PMT – PassMark single thread benchmark score (higher is better), PMR – PassMark overall CPU rank in PassMark database (lower is better)

General		Architecture							Caches per CPU				Caches per core			PassMark Scores		
CPU Model Name	**First Seen**	**Type**	**TCH**	**FRQ**	**FBS**	**CM**	**CRS**	**TRD**	**L1i**	**L1d**	**L2**	**L3**	**L1i**	**L1d**	**L2**	**PMC**	**PMT**	**PMR**
AMD Opteron 4171 HE	Q4 2010	K10	45	2100	3200		6	6	384	384	3072	6144	64	64	512	**3664**[1]	**732**[1]	**1147**[1]
Intel Xeon E5-2660	Q2 2012	Sandy Bridge	32	2200	4000	22	8	16	256	256	2048	20480	32	32	256	**11048**	**1387**	**265**
Intel Xeon E5-2673 v3	Q2 2015	Haswell	22	2400	4800	24	12	24	384	384	3072	30720	32	32	256	**16383**	**1666**	**116**
Intel Xeon E5-2698B v3	Q2 2014	Haswell	22	2000	4800	20	16	32	512	512	4096	40960	32	32	256	**21042**[2]	**1846**[2]	**51**[2]
Intel Xeon E5-2673 v4	Q4 2016	Broadwell	14	2300	4800	23	20	40	640	640	5120	51200	32	32	256	**21474**	**1792**	**46**
Intel Xeon Platinum 8168	Q4 2017	Skylake	14	2700	5200	27	24	48	768	768	24576	33792	32	32	1024	**29131**	**2073**	**3**

[1] PassMark Scores are provided for AMD Opteron 4170 HE
[2] PassMark Scores are provided for Intel Xeon E5-2698 v3

CONFIGURABLE SOFTWARE SYSTEMS

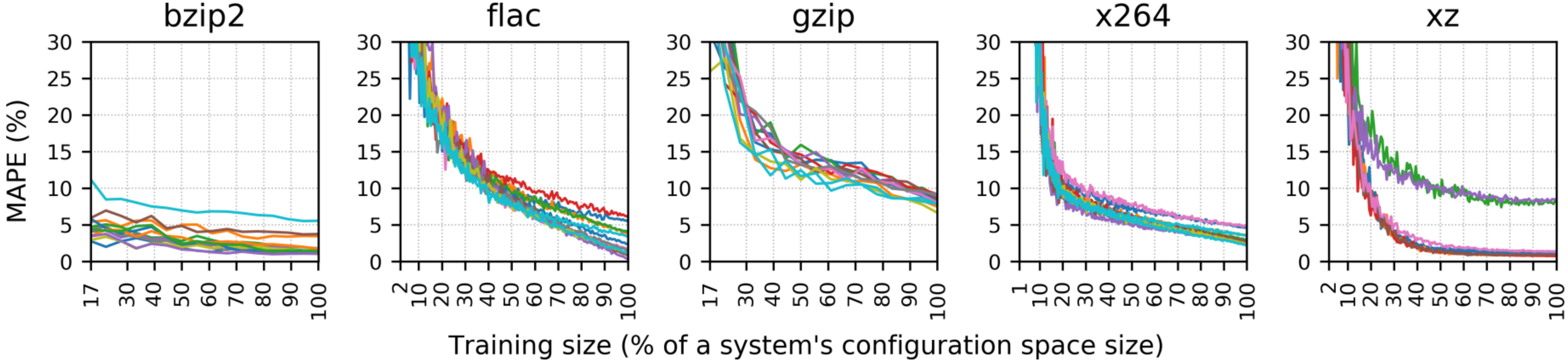
Table 3: Summary of benchmarked configurable software systems; N_f – Number of varied system features; NC – Number of benchmarked system configurations; NS – Number of servers on which systems were benchmarked.

Name	**N_f**	**NC**	**NS**
BZIP2	2	18	33
GZIP	3	36	28
XZ	4	160	27
FLAC	5	144	29
x264	8	256	30

TREE-BASED PREDICTORS ACCURACY

How accurate are property prediction models?

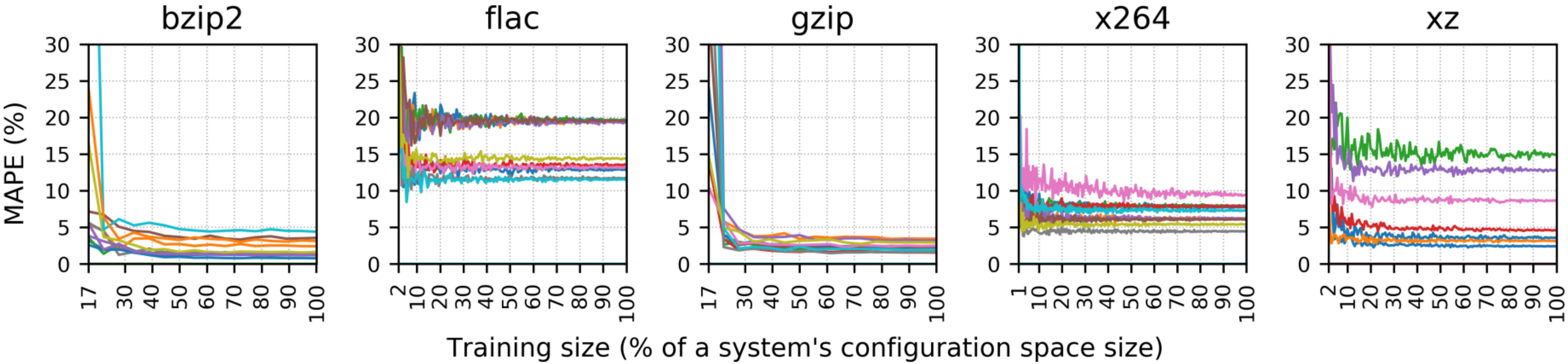
Figure 2: MAPE error distributions when predicting *compression time* metric using *regression trees* as predictors. Each line represents the error's distribution on a particular hardware.



LINEAR-BASED TRANSFERRERS ACCURACY

How accurate are linear-based property transferring models?

Figure 3: MAPE error distributions when transferring *compression time* metric using *using linear models* as transferrers. Each line represents the error's distribution on a particular hardware.

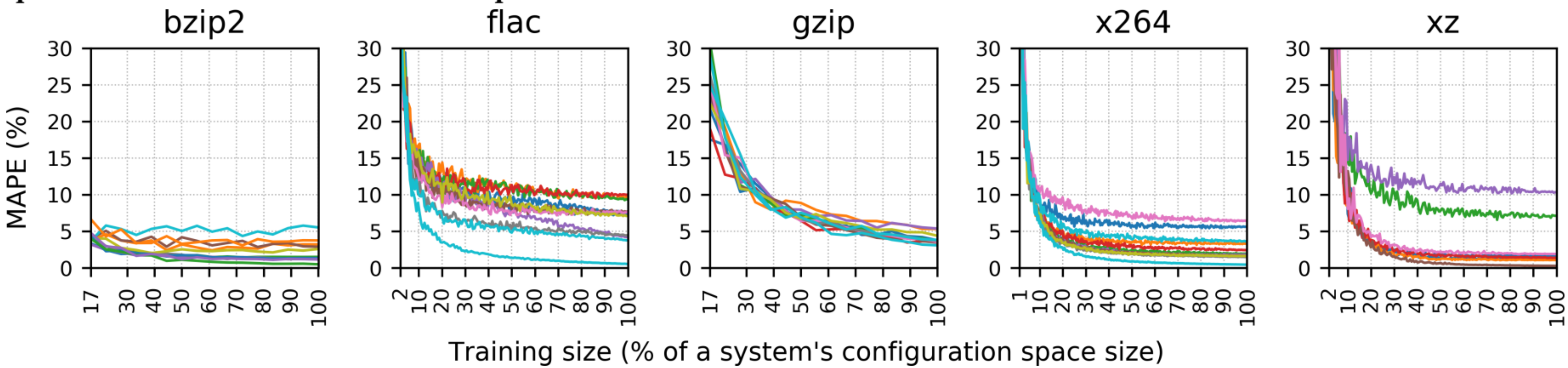


UNIVERSITY OF WATERLOO

TREE-BASED TRANSFERRERS ACCURACY

How accurate are tree-based property transferring models?

Figure 4: MAPE error distributions when transferring *compression time* metric using *regression trees* as transferrers. Each line represents the error's distribution on a particular hardware.



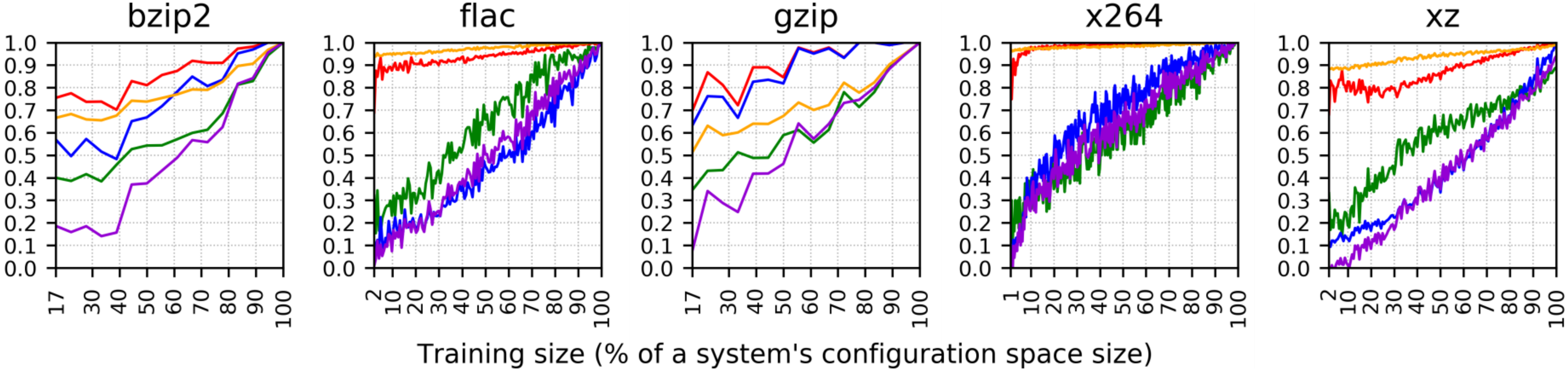
ASSESSING PARETO FRONTIERS

- Pareto frontier as a binary classifier
- Classical statistical measures for assessing binary classifiers
- True positive rate: frontier contains all optimal configs
- True negative rate: frontier left out all non-optimal configs
- Positive predictive value: classified config is truly optimal
- Negative predictive value: classified config is truly non-optimal
- Matthews c.c.: strong quantitative differences between classes

APPROXIMATED PARETO FRONTIERS QUALITY

How accurate are approximated Pareto frontiers?

Figure 5: Classification measures' distributions, applied to evaluation of Pareto frontiers, *approximated using regression trees*. Each line represents a particular measure: TPR (green), TNR (red), PPV (blue), NPV (orange), MCC (violet).

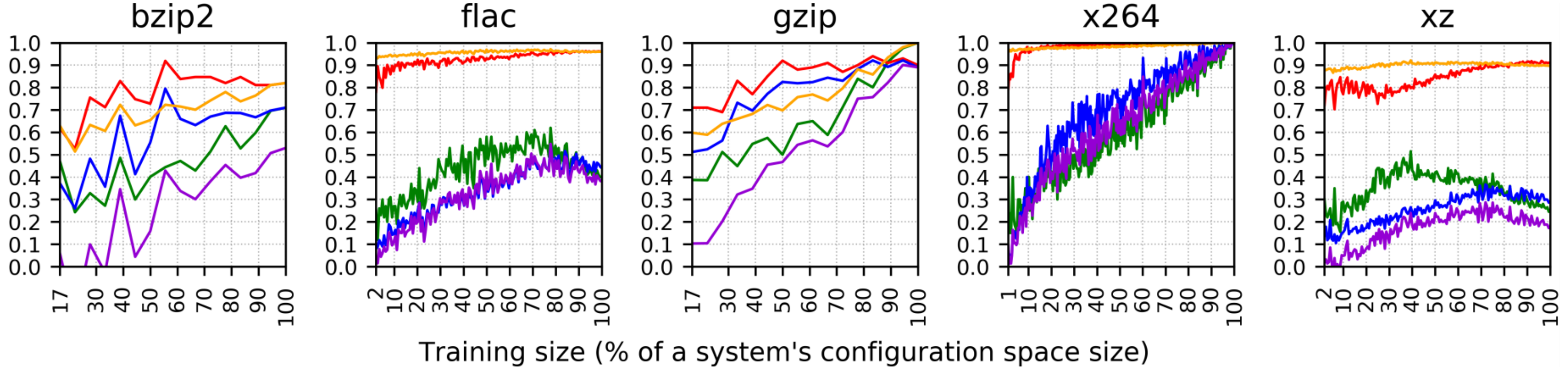


TRANSFERRED PARETO FRONTIERS QUALITY

How accurate are transferred Pareto frontiers?

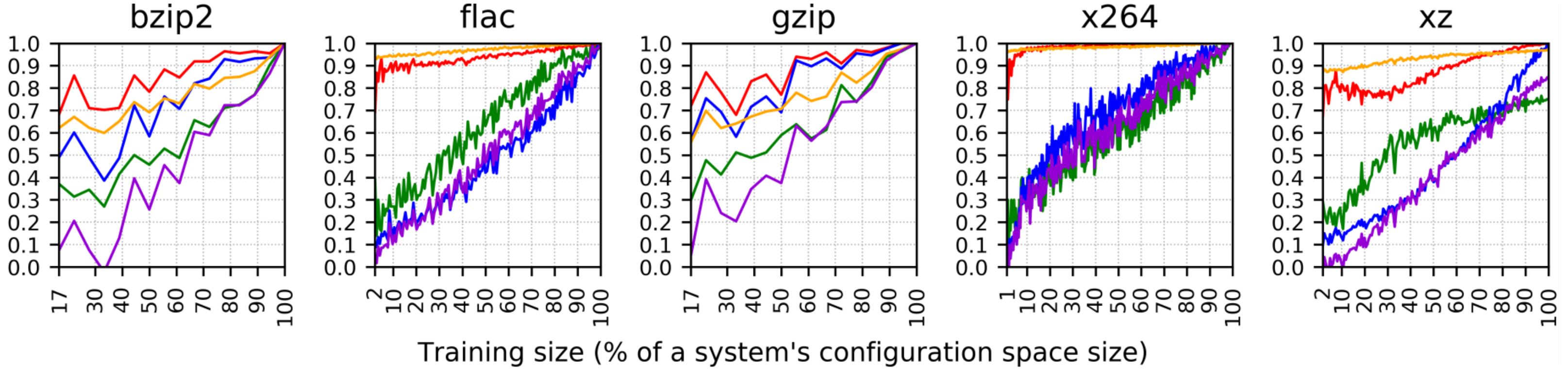
Linearly-Transferred

Figure 6: Classification measures' distributions, applied to evaluation of Pareto frontiers, *transferred using linear models*. Each line represents a particular measure: TPR (green), TNR (red), PPV (blue), NPV (orange), MCC (violet).



Tree-Transferred

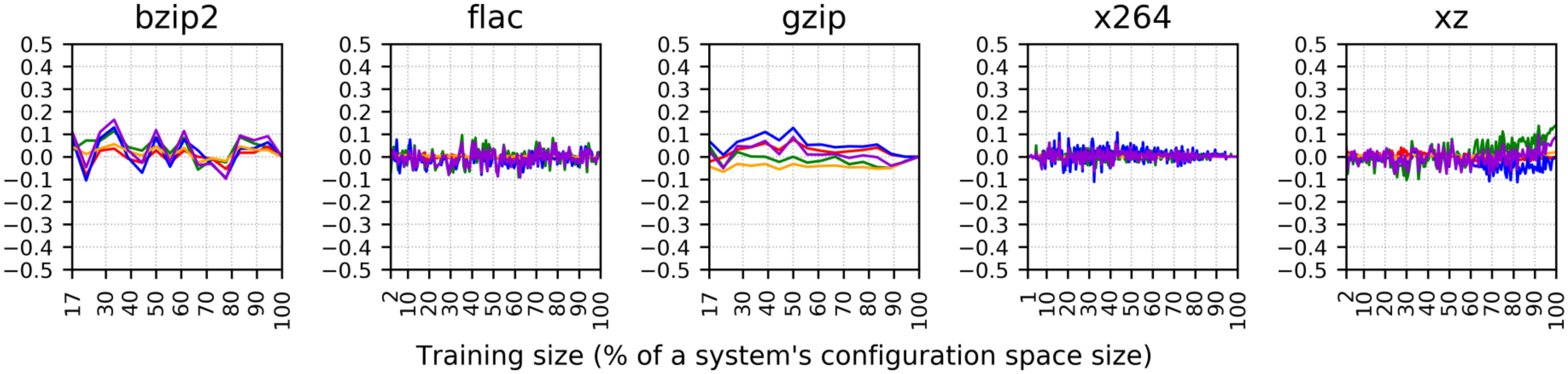
Figure 7: Classification measures' distributions, applied to evaluation of Pareto frontiers, *transferred using regression trees*. Each line represents a particular measure: TPR (green), TNR (red), PPV (blue), NPV (orange), MCC (violet).



TRANSFERRING PROCESS INACCURACY

How much error does the transferring process add?

Figure 8: Distributions of deltas between classification measures of approximated and transferred (using regression trees) Pareto frontiers. Each line represents a particular measure: TPR (green), TNR (red), PPV (blue), NPV (orange), MCC (violet).



CONCLUSION

- Approach for approximation and transferring of Pareto frontiers
- Evaluated property predictors and transferrers
- Demonstrated superiority of tree-based transferrers
- Evaluated approximated and transferred Pareto frontiers
- Frontiers improve linearly with increase of a sample size
- Transferring has a minor influence on a final frontier accuracy
- In future work, we will focus on approximation process improving