



Performance Comparison Between BPMN 2.0 Workflow Management Systems Versions

Vincenzo Ferme, Ana Ivanchikj, Cesare Pautasso

USI Lugano, Switzerland

Marigianna Skouradaki, Frank Leymann

University of Stuttgart, Germany



Semantic Versioning

1 . 3 . 1

BREAKING . FEATURE . FIX

incompatible
API changes

**breaking
change**

add backwards-
compatible
functionality

**new
feature**

make backwards-
compatible bug fix

**bug
fix**



Semantic Versioning

1 . 3 . 1

BREAKING . FEATURE . FIX

incompatible
API changes

**breaking
change**

add backwards-
compatible
functionality

**new
feature**

make backwards-
compatible bug fix

**bug
fix**



Semantic Versioning

1 . 3 . 1

BREAKING . FEATURE . FIX

incompatible
API changes

**breaking
change**

add backwards-
compatible
functionality

**new
feature**

make backwards-
compatible bug fix

**bug
fix**





2 . 5 . 0

BREAKING . FEATURE . FIX

incompatible
API changes

breaking
change

add backwards-
compatible
functionality

new
feature

make backwards-
compatible bug fix

bug
fix

1 . 3 . 1

1 . 4 . 1

New Functionality



1 . 4 . 1

New Functionality



Performance?
Resource Consumption?

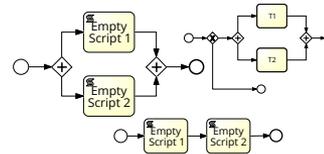


Before testing:

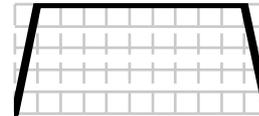
WfMS (versions)?



Workload mix?



Load function?



Metrics?



Representative of the company

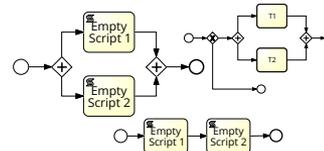


While testing:

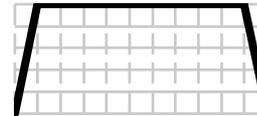
WfMS (versions)?



Workload mix?



Load function?



Metrics?



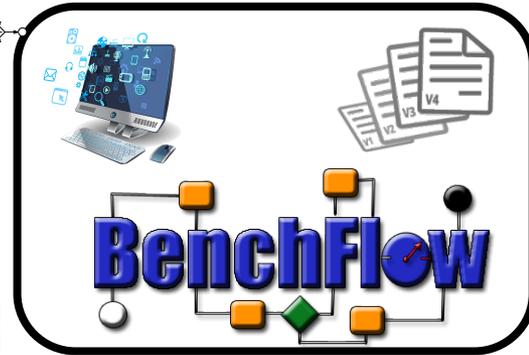
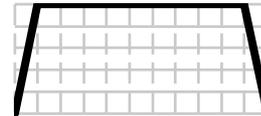
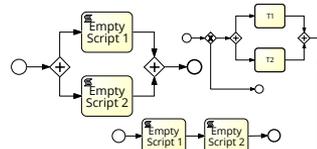
While testing:

WfMS (versions)?

Workload mix?

Load function?

Metrics?



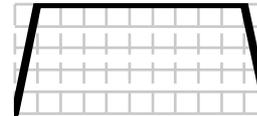
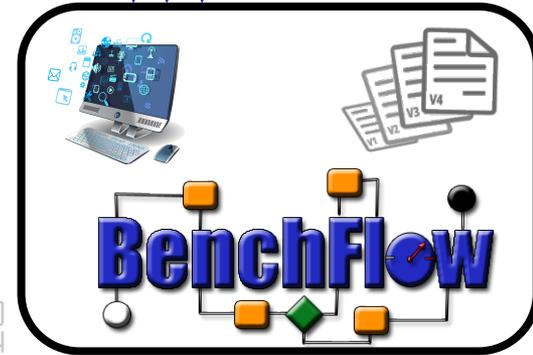
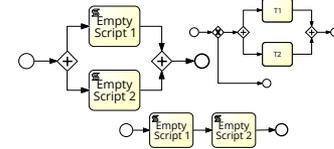
While testing:

WfMS (versions)?

Workload mix?

Load function?

Metrics?



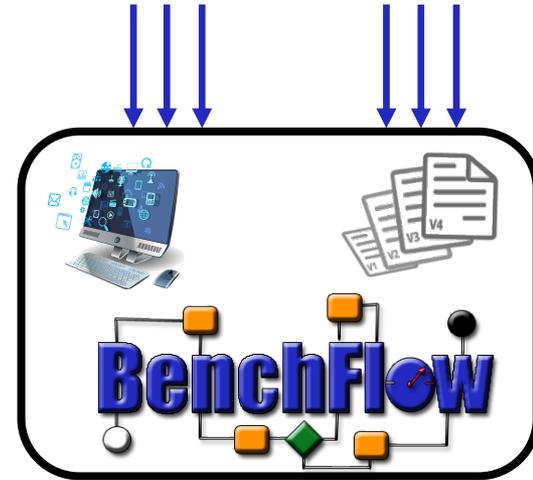
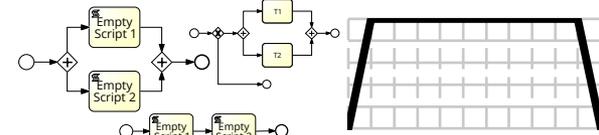
While testing:

WfMS (versions)?

Workload mix?

Load function?

Metrics?



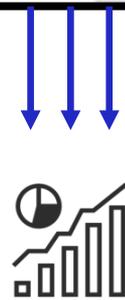
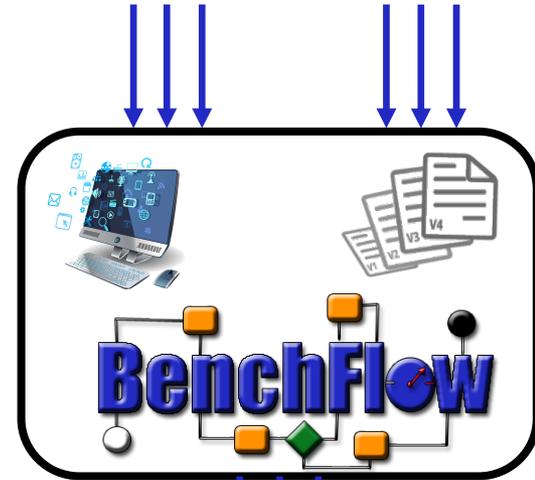
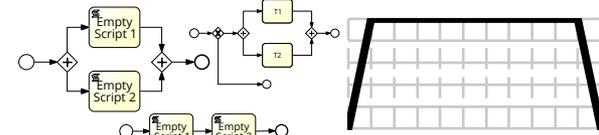
While testing:

WfMS (versions)?

Workload mix?

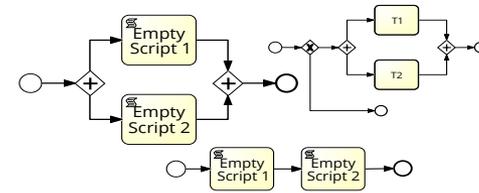
Load function?

Metrics?



Paper Contributions

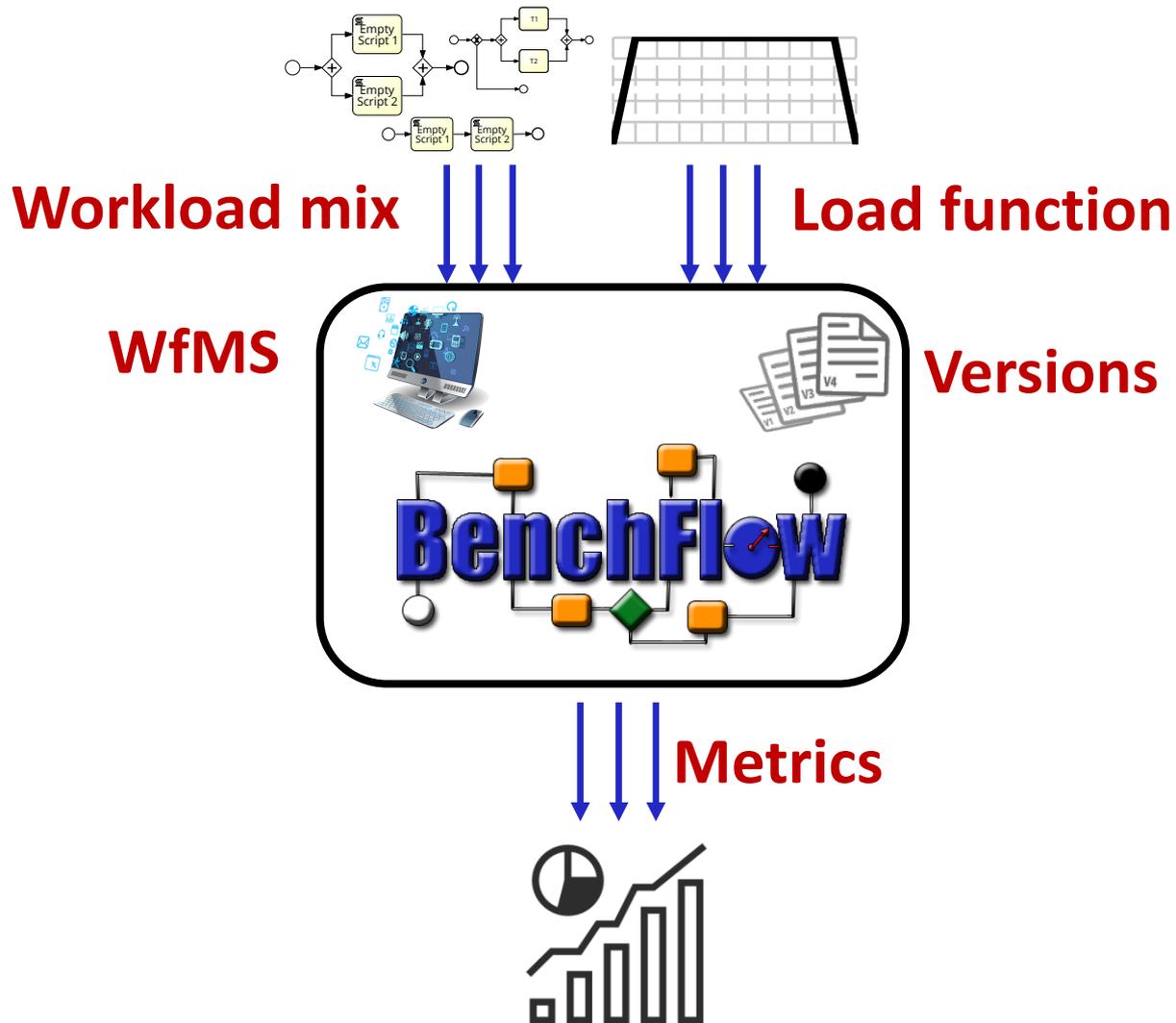
A novel method for deriving a synthetic workload mix from a given BP models collection



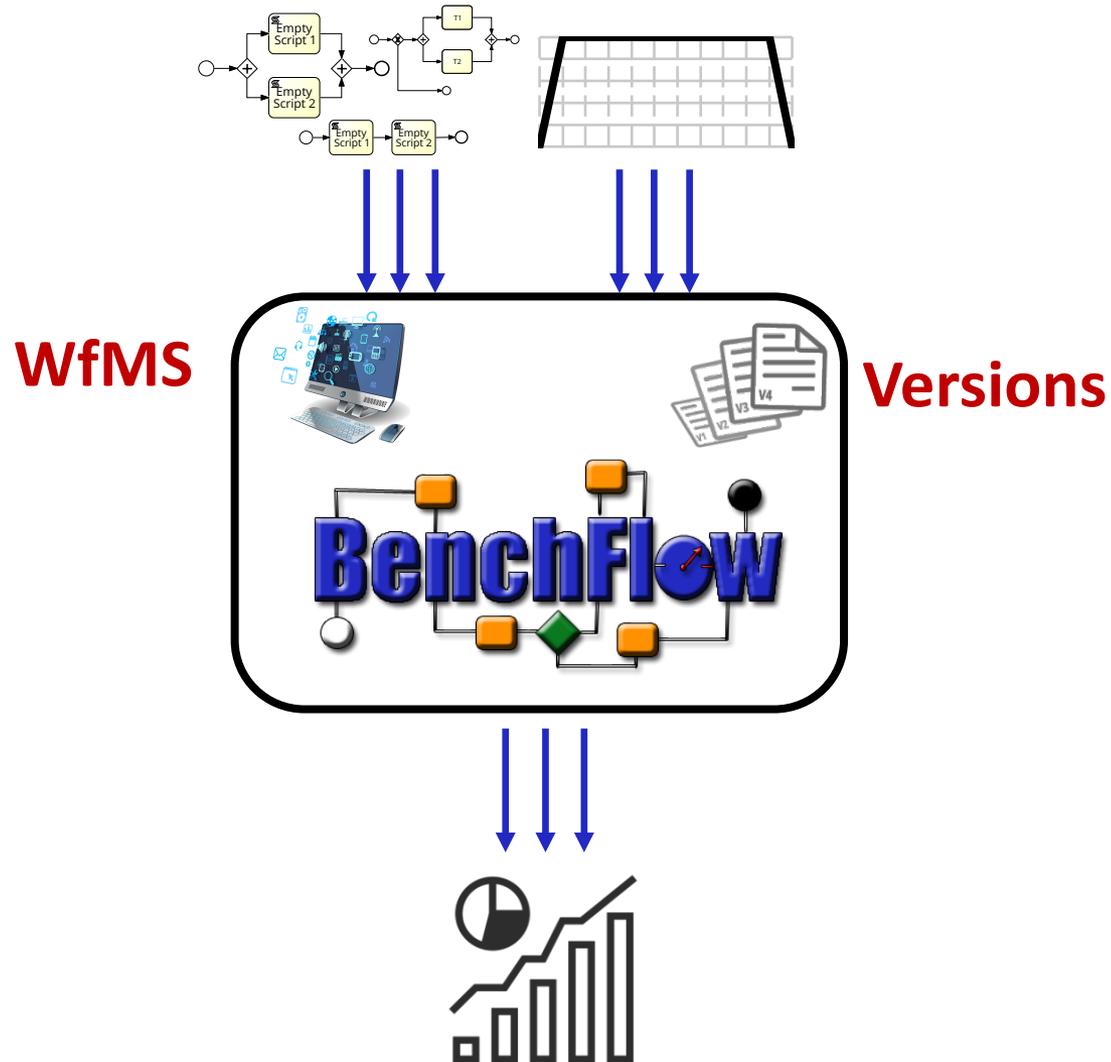
Insights on the evolution of two WfMSs in their last 4 minor versions in terms of performance and resource consumption



Performance testing



Performance testing



WfMSs and Versions



WfMS A

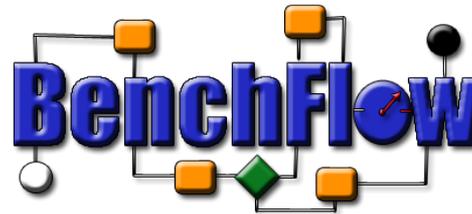
- official Docker image
- default config.

WfMS B

- popular Docker image
- vendor config.

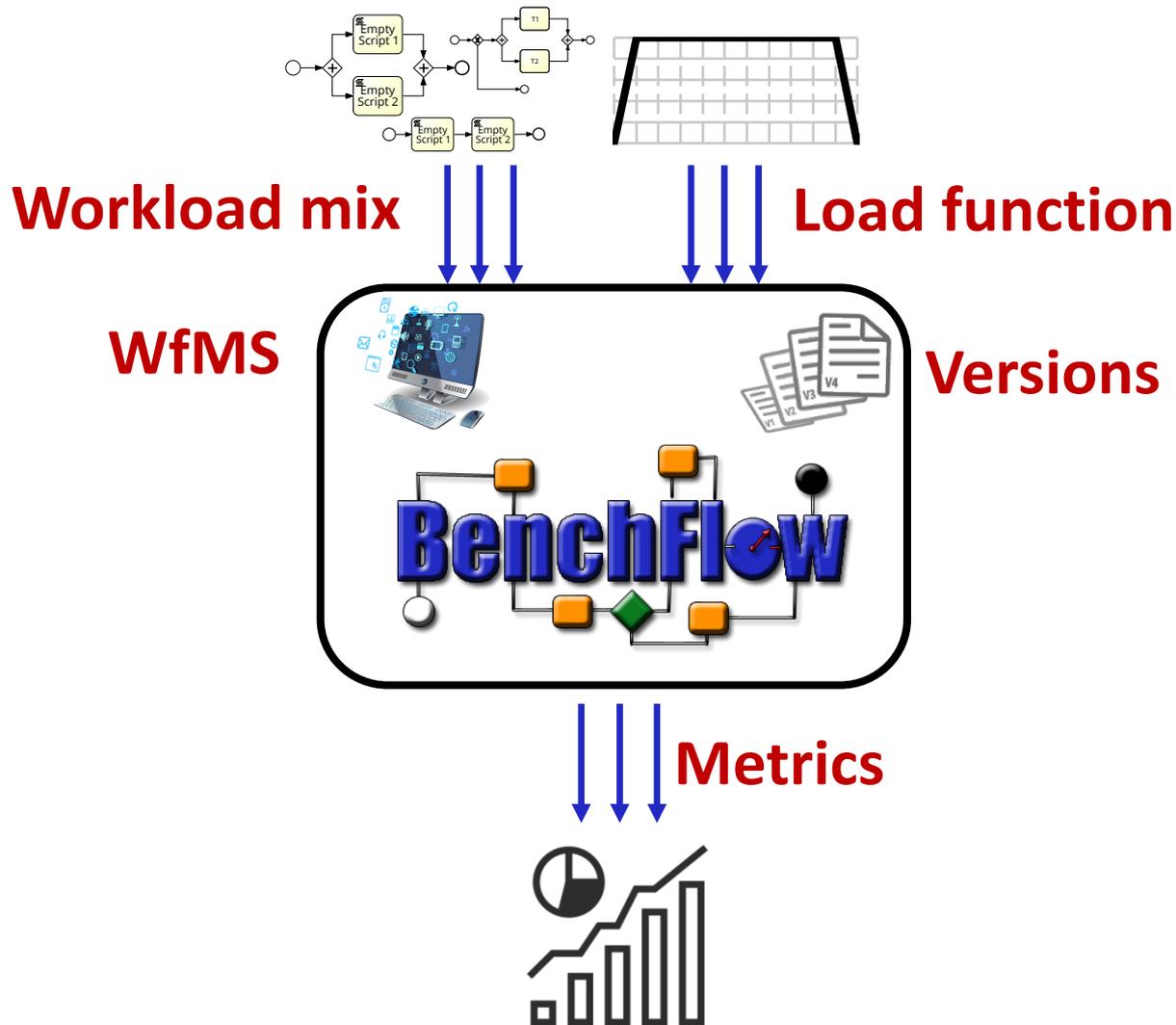
(2014-2016)

- v 7.2.0 - v 7.3.0
- v 7.4.0 - v 7.5.0

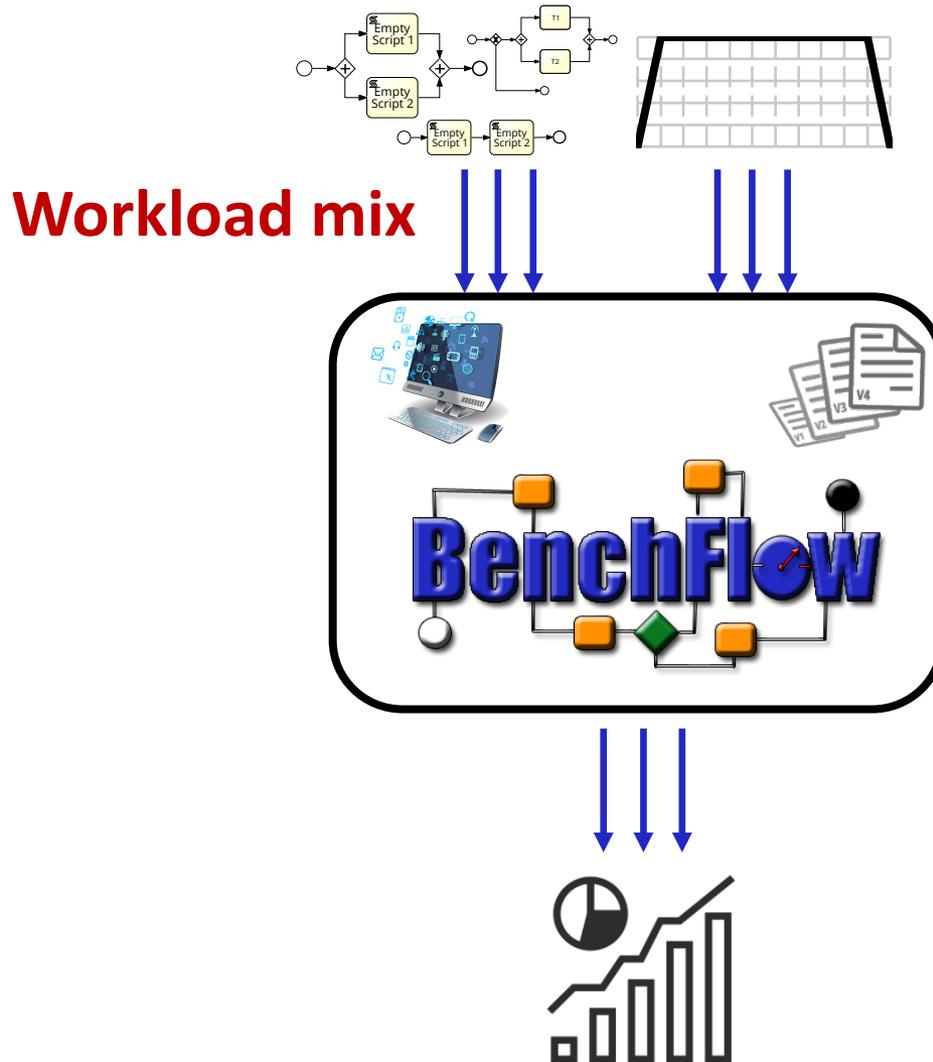


- v 5.18.0 - v 5.19.0.2
- v 5.20.0 - v 5.21.0

Performance testing



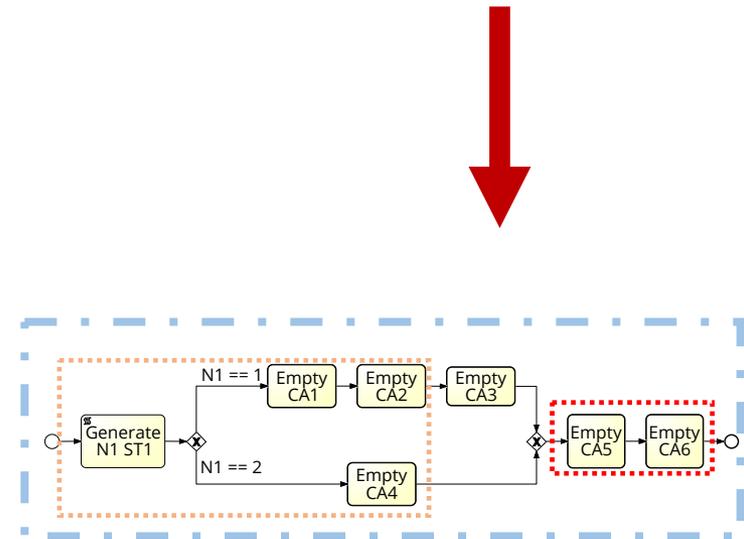
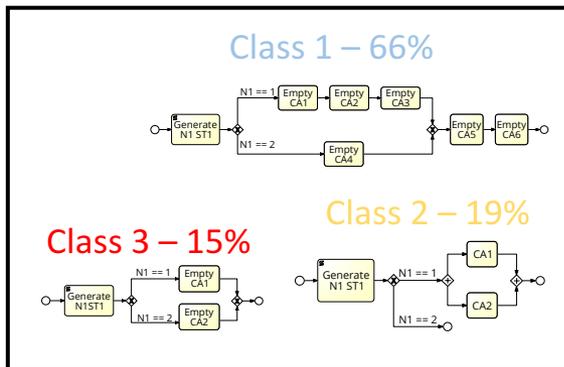
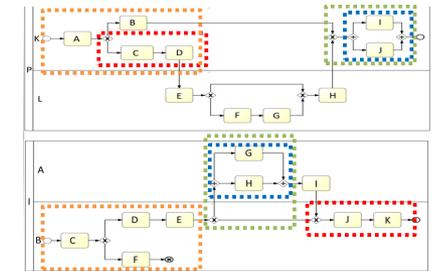
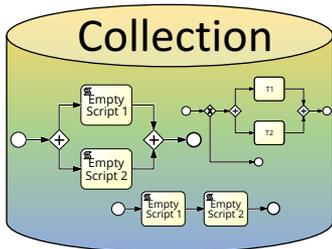
Performance testing



Workload Mix Generation

I Analyze

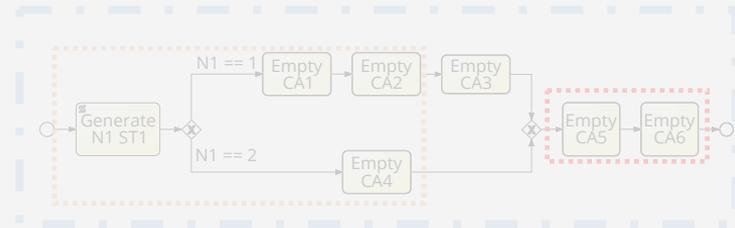
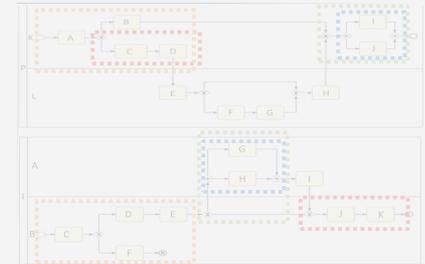
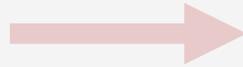
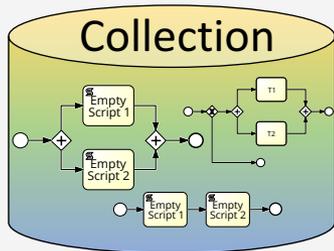
II Extract Reoccurring Structures



III Synthesize Representative BPs

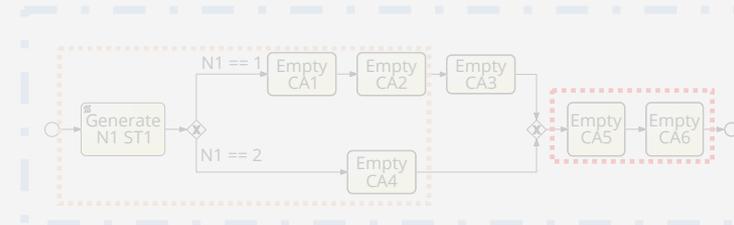
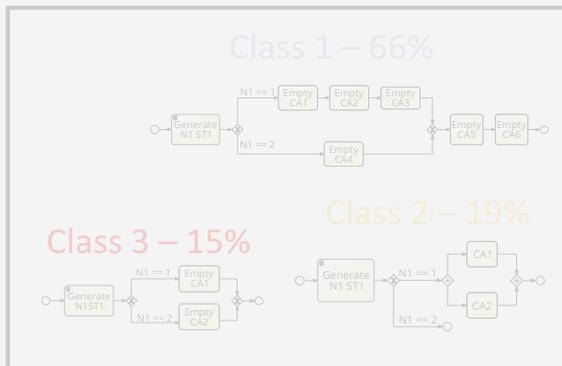
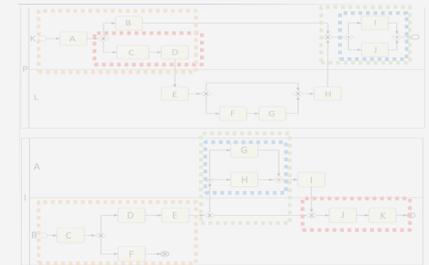
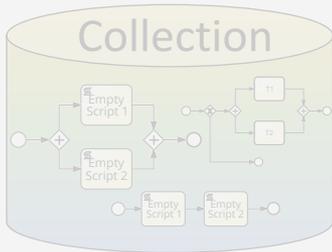
IV Define the workload mix (set of classes)

Workload Mix Generation



Workload Mix Generation

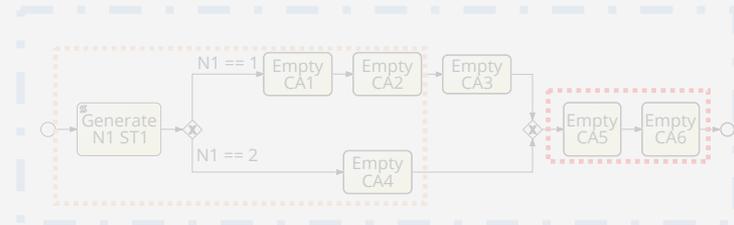
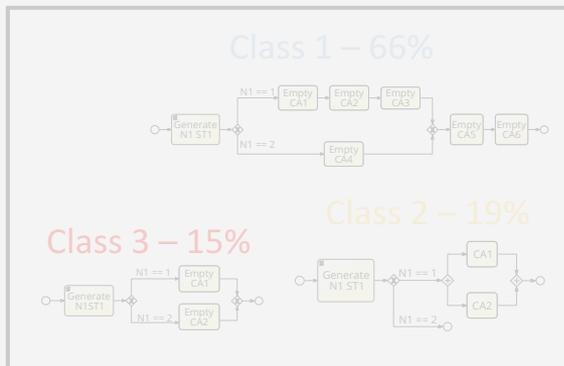
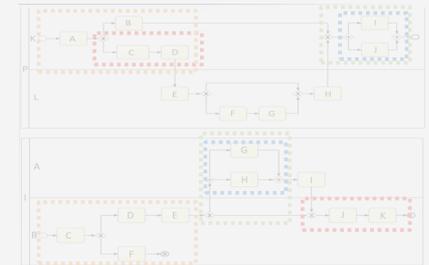
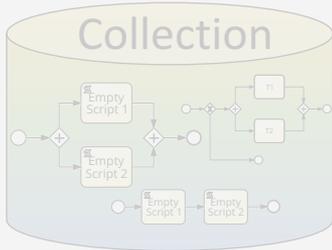
I Analyze



Workload Mix Generation

I Analyze

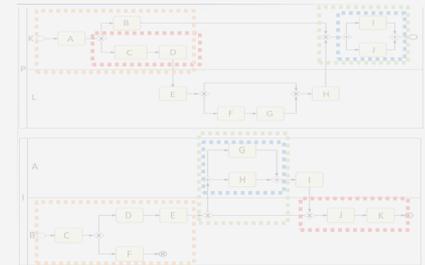
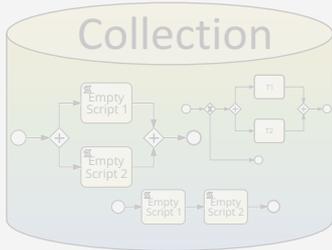
#Nodes
#Gateways...



Workload Mix Generation

I Analyze

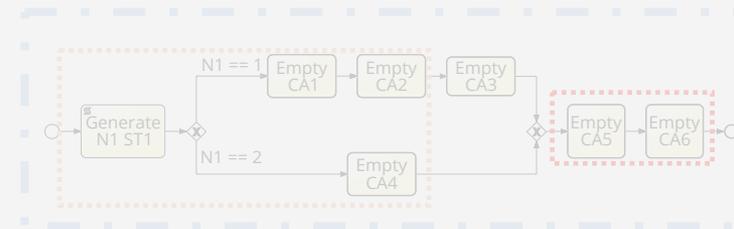
#Nodes
#Gateways...



Cluster 1:
7 tasks
2 XOR gateways

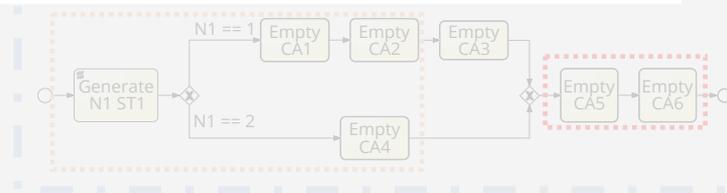
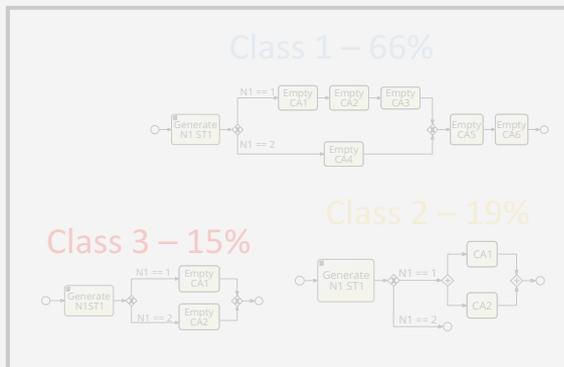
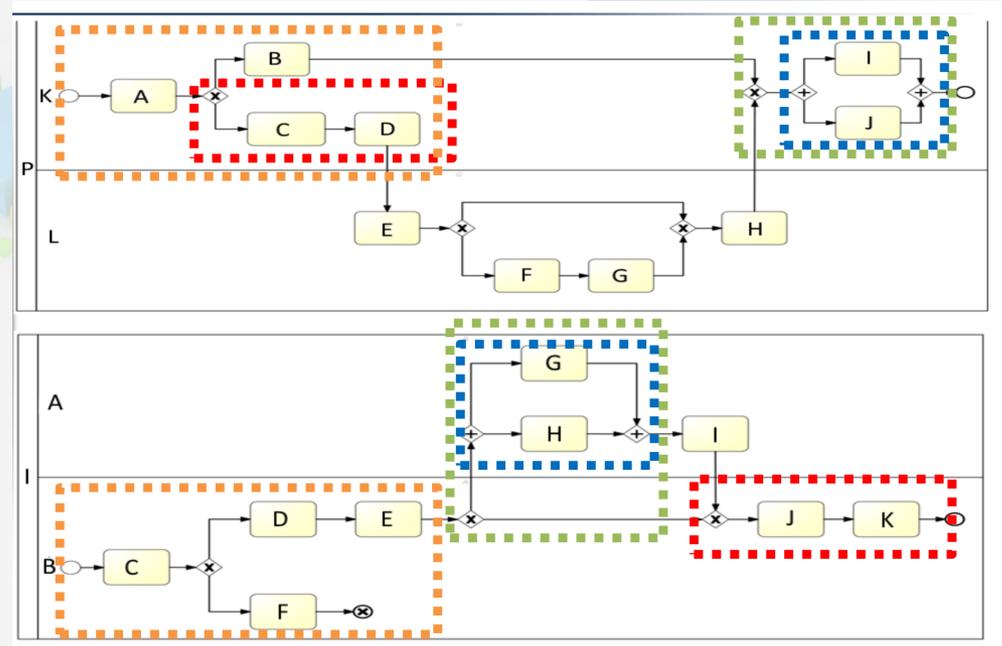
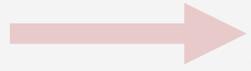
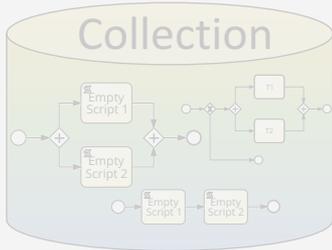
Cluster 3...

Cluster 2...



Workload Mix Generation

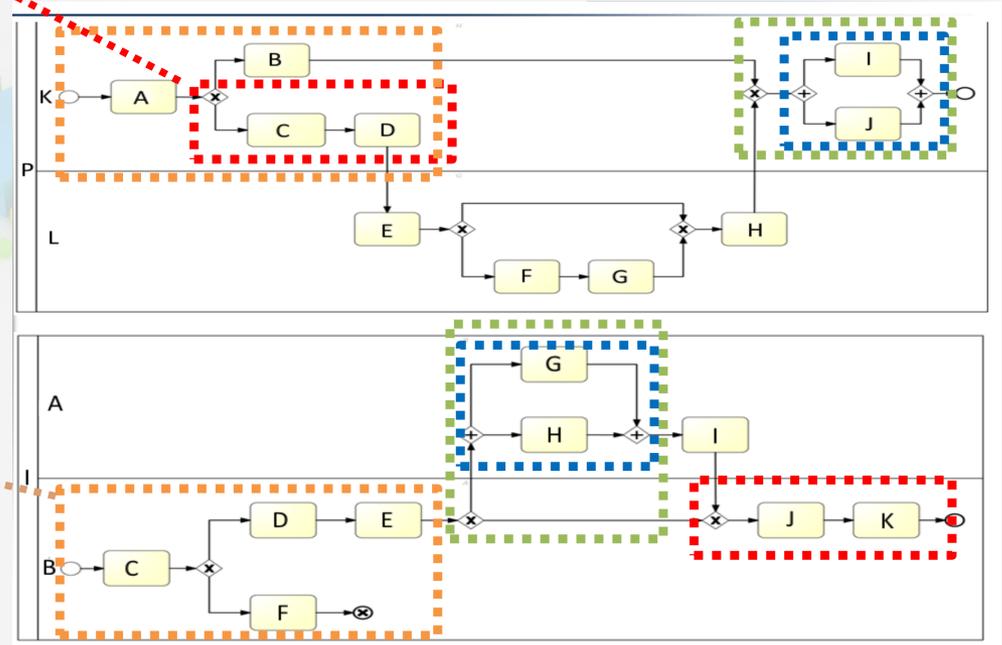
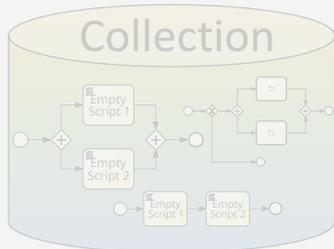
II Extract Reoccurring Structures



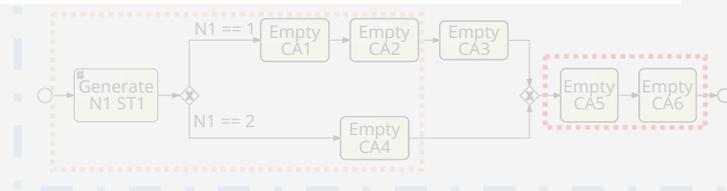
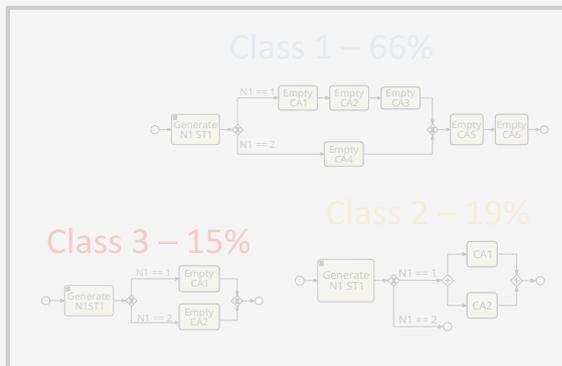
Workload Mix Generation

Metadata: 2 tasks
Fr. of appearance: 50
#BPs: 48

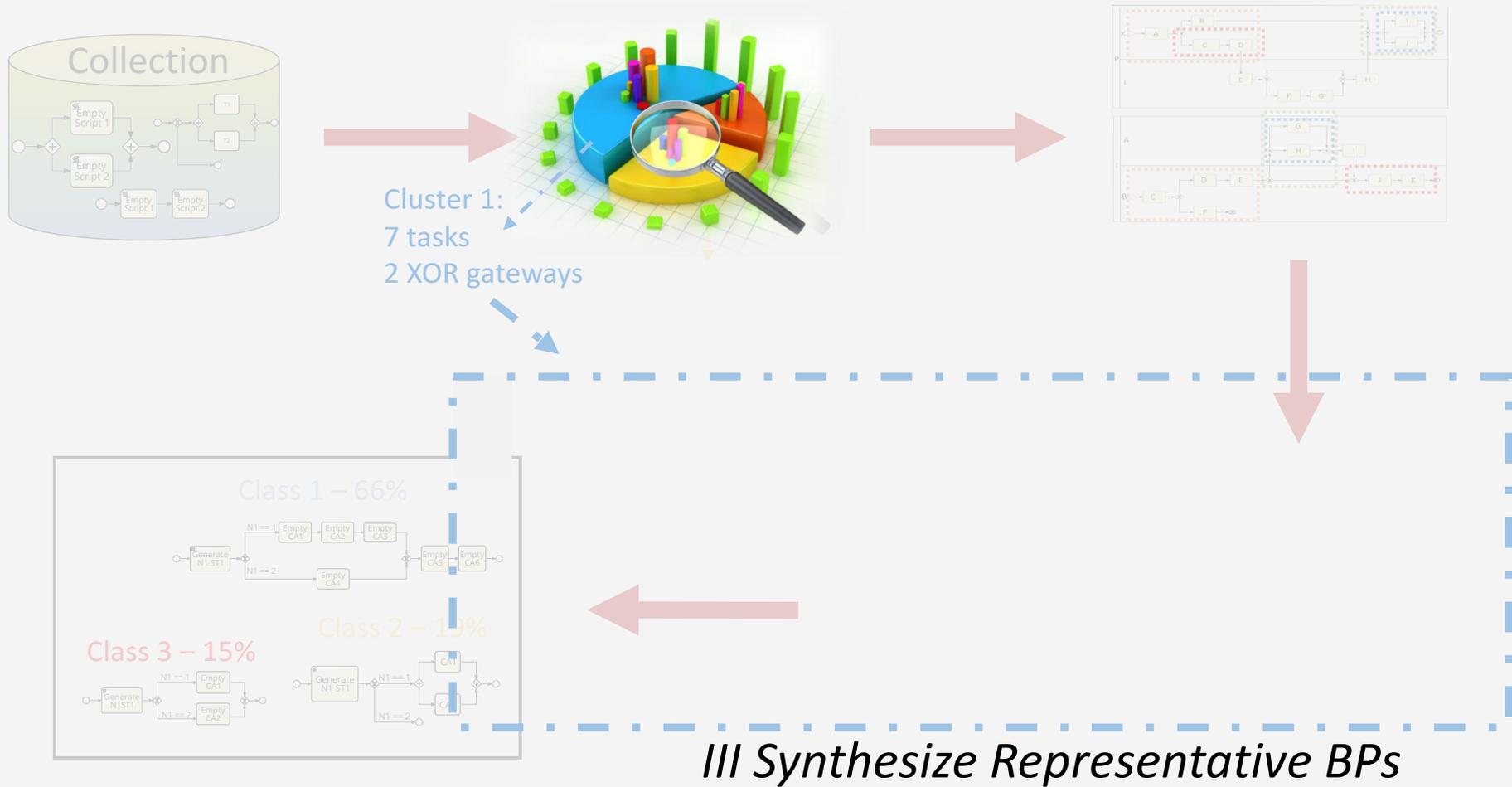
II Extract Reoccurring Structures



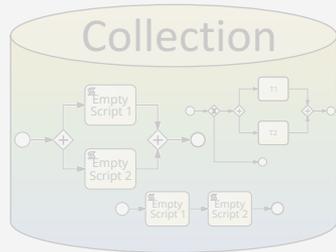
Metadata: 3 tasks, 1 XOR gateway
Fr. of appearance: 20
#BPs: 16



Workload Mix Generation

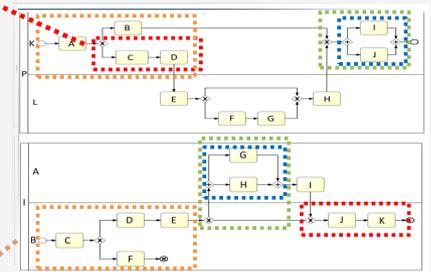


Workload Mix Generation

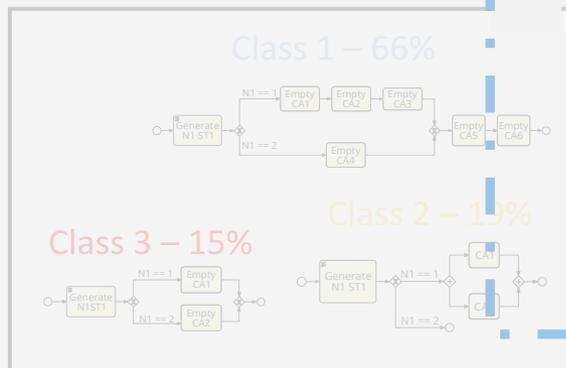


Cluster 1:
7 tasks
2 XOR gateways

Metadata: 2 tasks
Fr. of appearance: 50
#BPs: 48

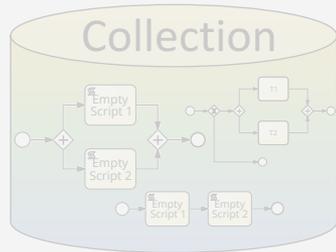


Metadata: 3 tasks, 1 XOR gateway
Fr. of appearance: 20
#BPs: 16



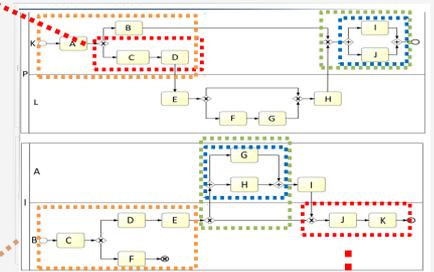
III Synthesize Representative BPs

Workload Mix Generation

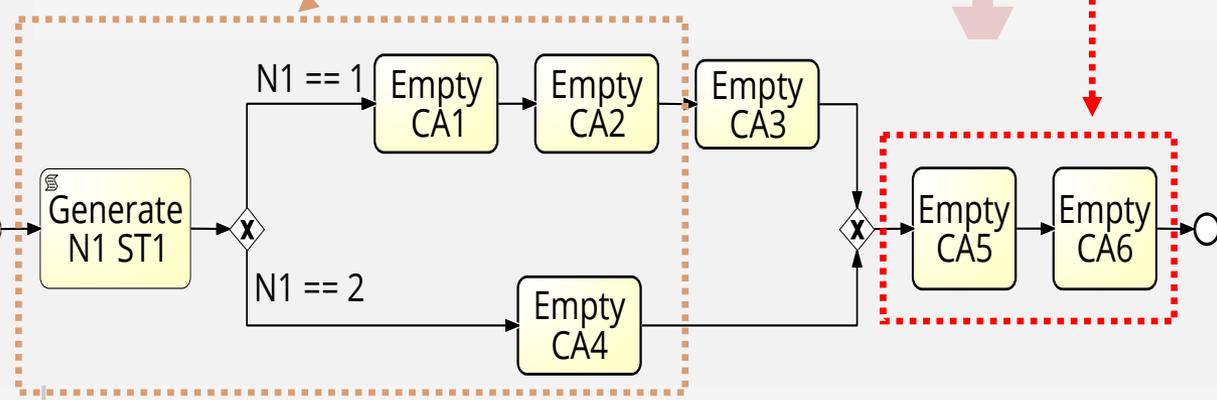
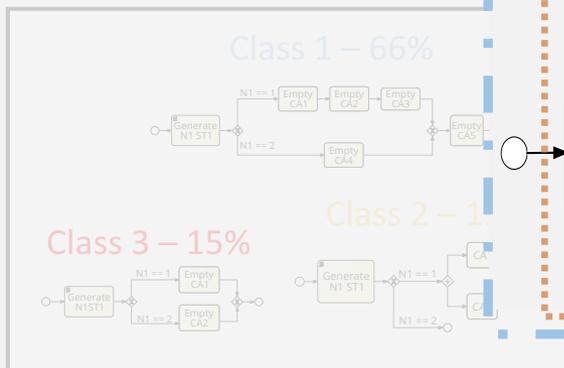


Cluster 1:
7 tasks
2 XOR gateways

Metadata: 2 tasks
Fr. of appearance: 50
#BPs: 48

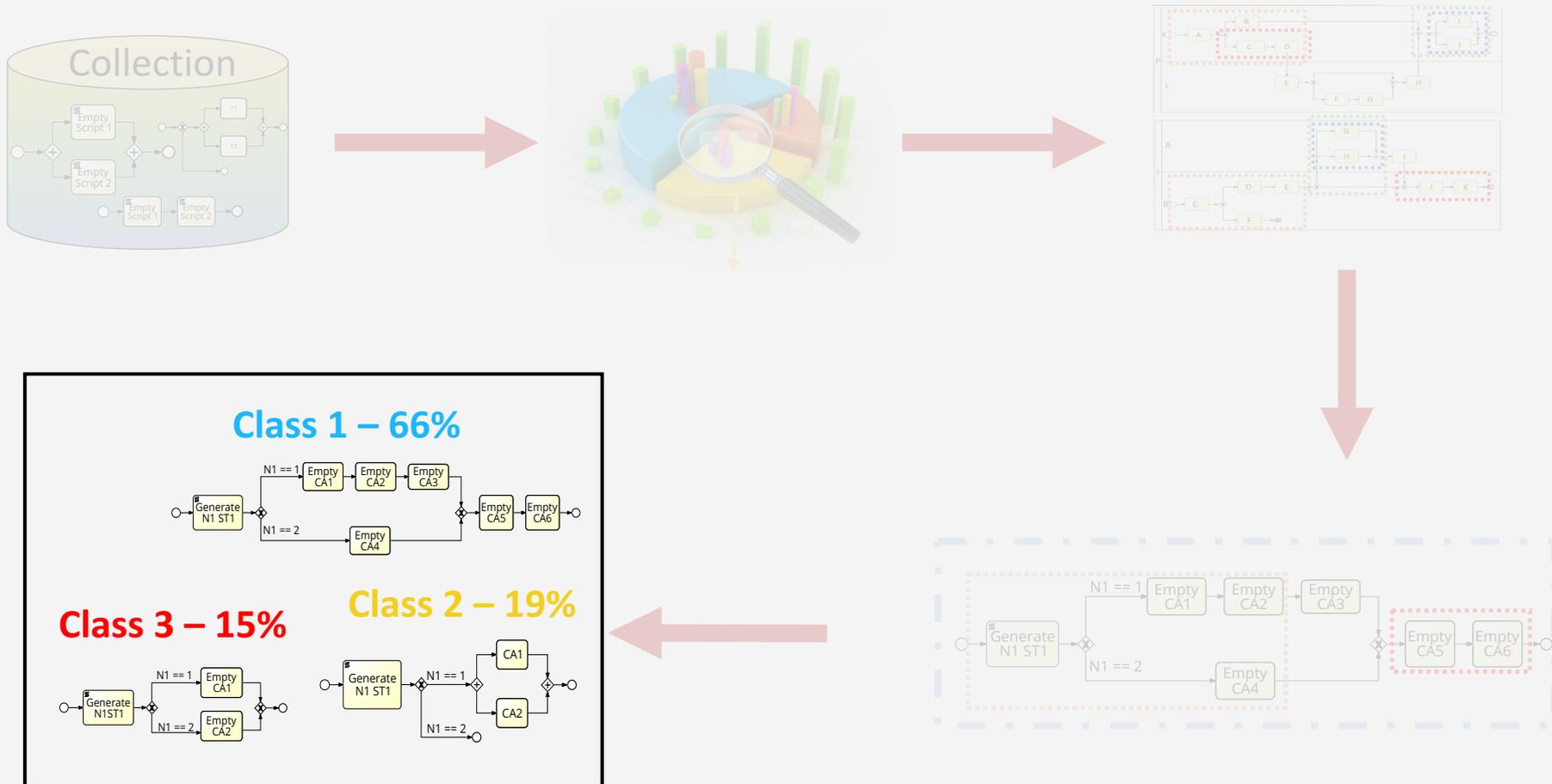


Metadata: 3 tasks, 1 XOR gateway
Fr. of appearance: 20
#BPs: 16



III Synthesize Representative BPs

Workload Mix Generation

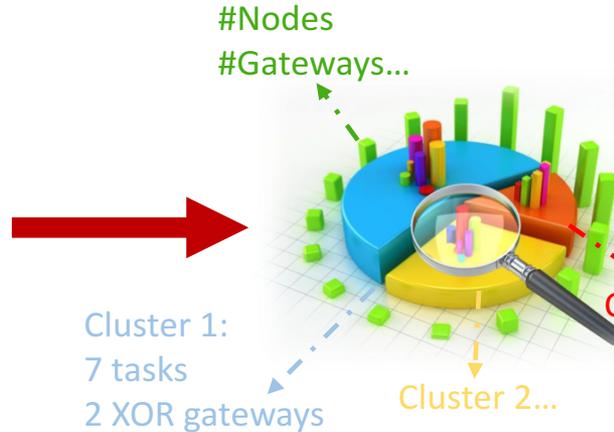
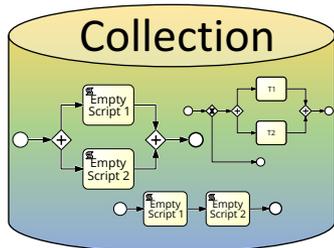


IV Define the workload mix (set of classes)

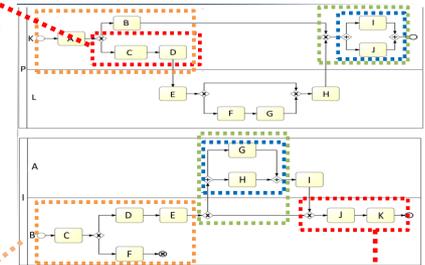
Workload Mix Generation

I Analyze

II Extract Reoccurring Structures



Metadata: 2 tasks
Fr. of appearance: 50
#BPs: 48

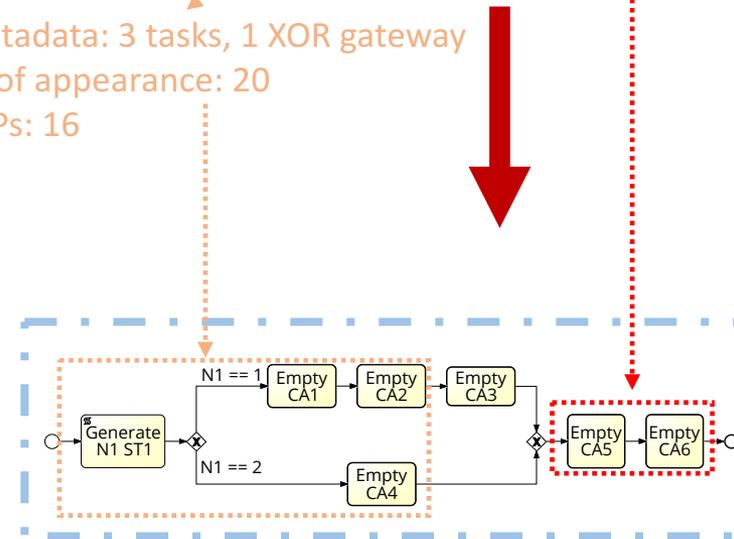
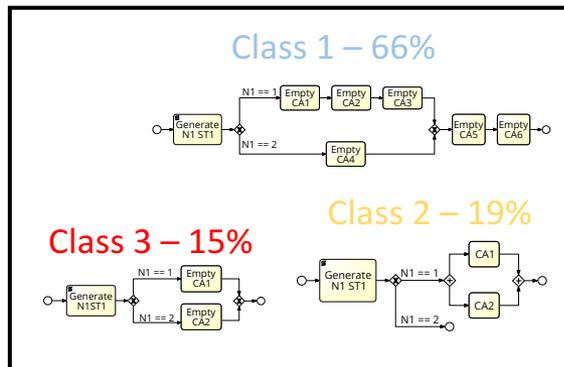


Cluster 1:
7 tasks
2 XOR gateways

Cluster 2...

Cluster 3...

Metadata: 3 tasks, 1 XOR gateway
Fr. of appearance: 20
#BPs: 16



III Synthesize Representative BPs

IV Define the workload mix (set of classes)

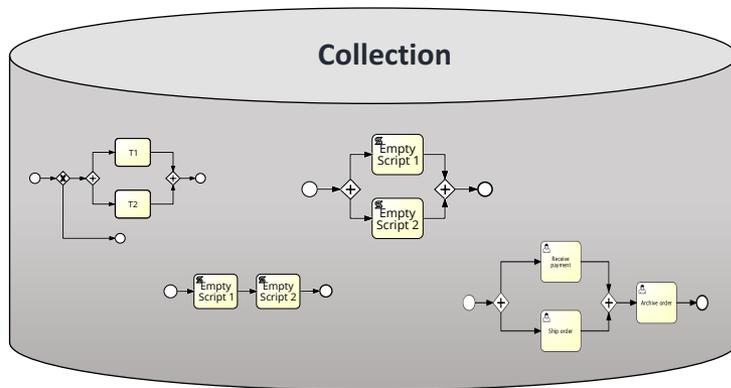
Process Models Collection

- Workload Mix Generation Applied -

Total
14,167 Process Models

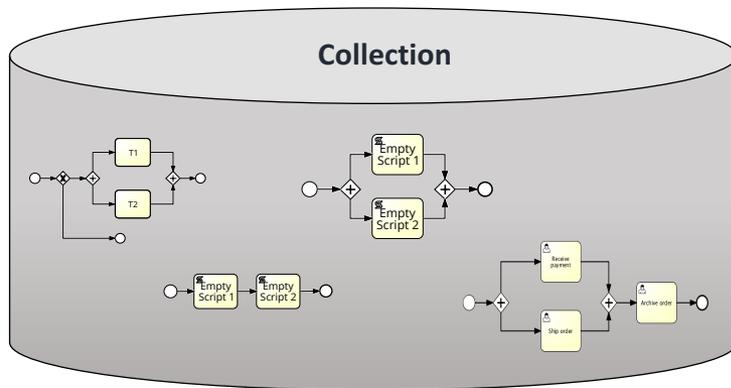
Including:

- Invalid process models
- Incomplete process models
- Mock-up process models
- *Anonymized* process models



Process Models Collection

- Workload Mix Generation Applied -

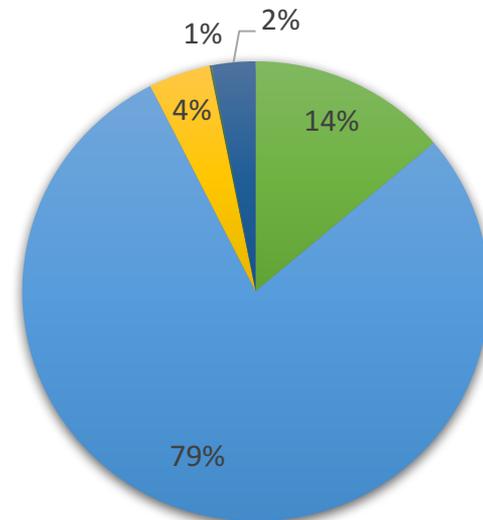


Total
14,167 Process Models

Including:

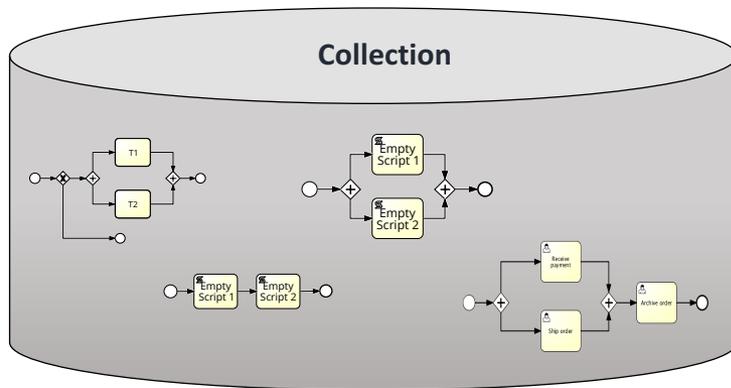
- Invalid process models
- Incomplete process models
- Mock-up process models
- *Anonymized* process models

■ .pnml ■ .bpmn ■ .yawl ■ .bpel ■ .EPC



Process Models Collection

- Workload Mix Generation Applied -

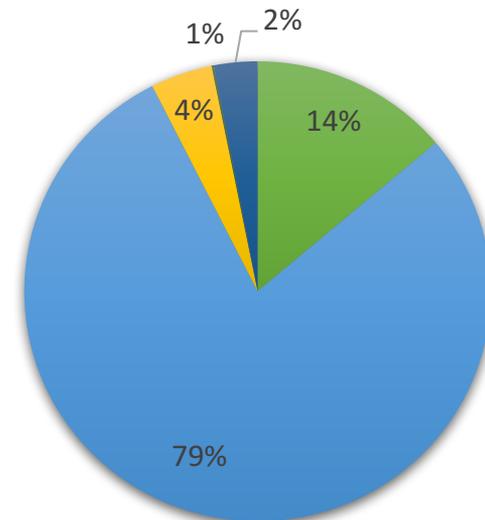


Total
14,167 Process Models

Including:

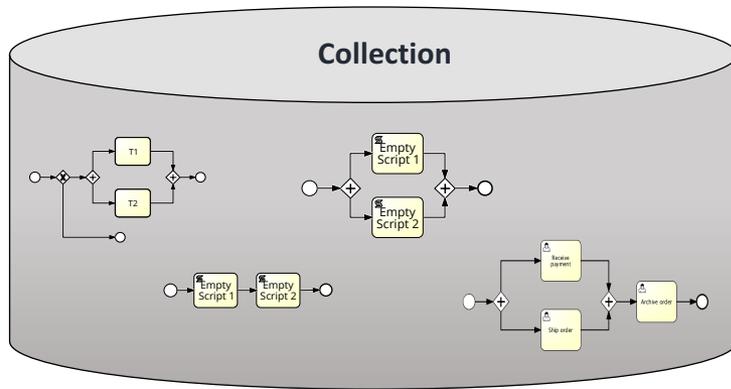
- Invalid process models
- Incomplete process models
- Mock-up process models
- *Anonymized* process models

■ .pnml ■ .bpmn ■ .yawl ■ .bpel ■ .EPC



Process Models Collection

- Workload Mix Generation Applied -

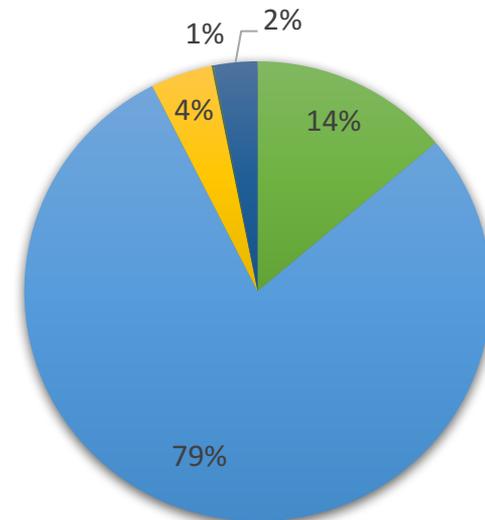


Total
14,167 Process Models

Including:

- Invalid process models
- Incomplete process models
- Mock-up process models
- *Anonymized* process models

■ .pnml ■ .bpmn ■ .yawl ■ .bpel ■ .EPC



Process Models Collection

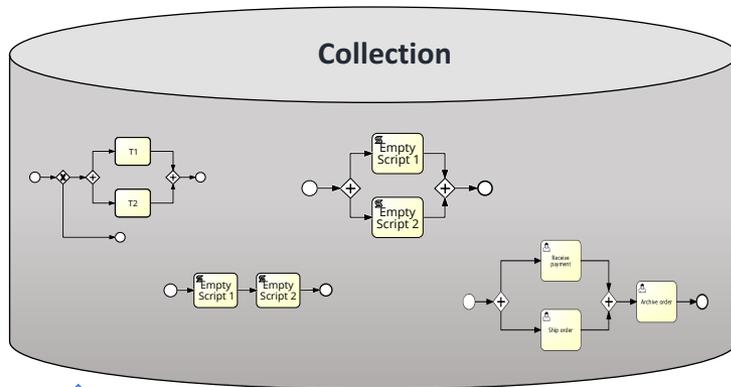
- Workload Mix Generation Applied -

Total

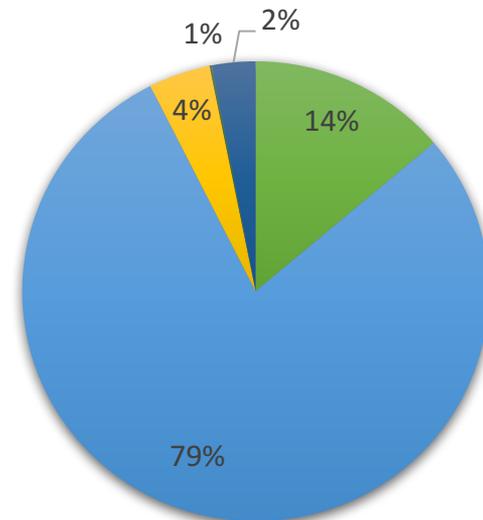
14,167 Process Models

Including:

- Invalid process models
- Incomplete process models
- Mock-up process models
- *Anonymized* process models



■ .pnml ■ .bpmn ■ .yawl ■ .bpel ■ .EPC



Process Models Collection

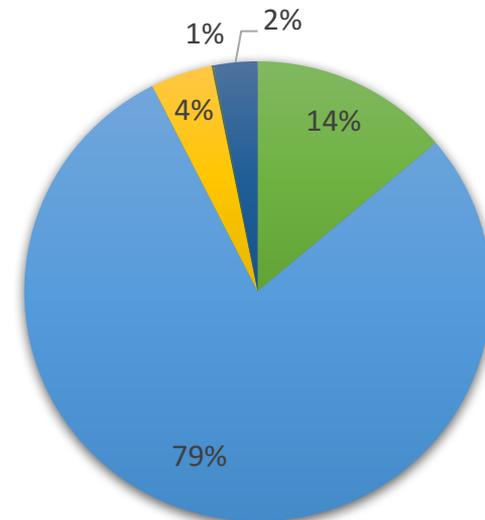
- Workload Mix Generation Applied -

Total
14,167 Process Models

Including:

- Invalid process models
- Incomplete process models
- Mock-up process models
- *Anonymized* process models

■ .pnml ■ .bpmn ■ .yawl ■ .bpel ■ .EPC



Statistical & Clustering Analysis

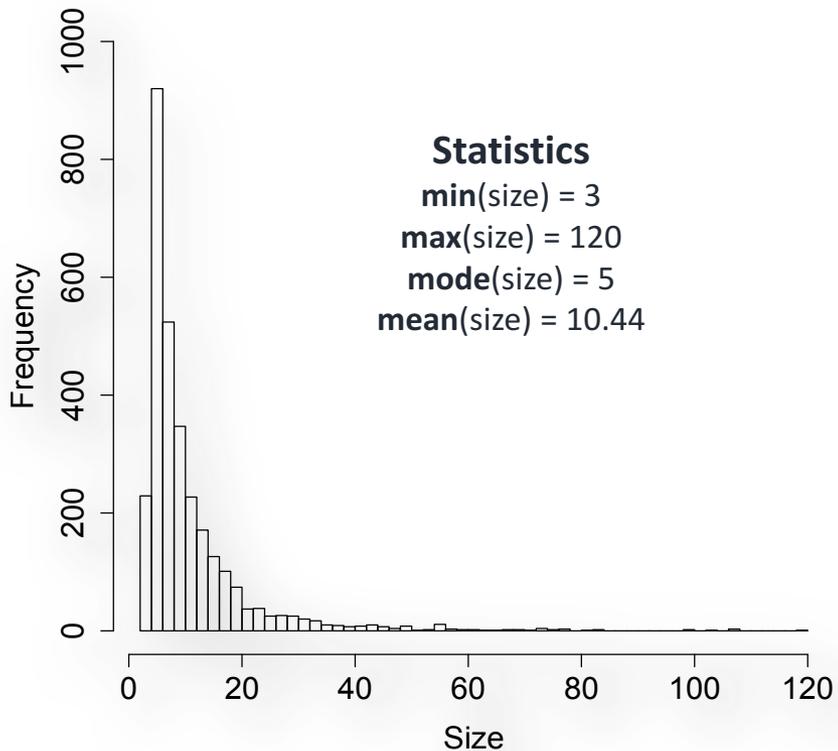
- *Workload Mix Generation Applied* -



Size of Representative Process Models:

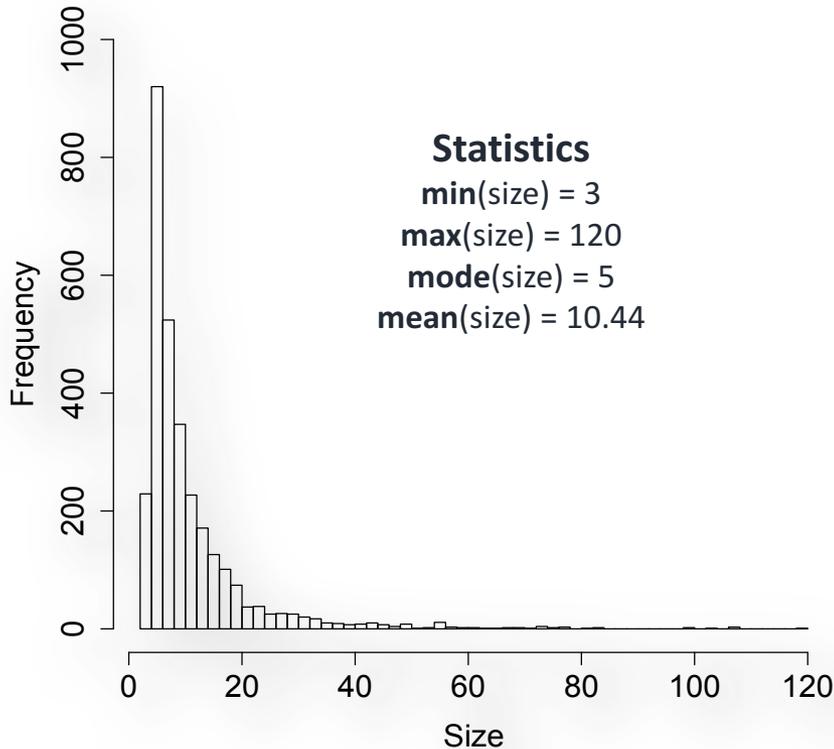
$$5 \leq \text{size} \leq 32$$

covers **82%** of the collection



Statistical & Clustering Analysis

- *Workload Mix Generation Applied* -



Size of Representative Process Models:

$$5 \leq \text{size} \leq 32$$

covers **82%** of the collection

k-means Clustering Analysis

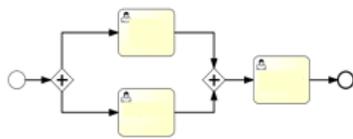
| | BPMN2.0 Elements |
|--------------------------|--|
| Cluster 1 (35.2%) | 1 Start Event, 2 End Events, 4 Tasks/Activities, 1 Exclusive Gateway |
| Cluster 2 (28.7%) | 1 Start Event, 2 End Events, 6 Tasks/Activities, 2 Exclusive Gateways |
| Cluster 3 (19.4%) | 1 Start Event, 3 End Events, 11 Tasks/Activities, 4 Exclusive Gateways, 1 Parallel Gateway |
| Cluster 4 (10.6%) | 1 Start Event, 3 End Events, 16 Tasks/Activities, 5 Exclusive Gateways, 1 Parallel Gateway |



Reoccurring Structures Extraction

- *Workload Mix Generation Applied* -

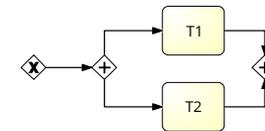
R_{SE} Algorithm Execution



3,247
BPMN 2.0
Process Models



14h for
5.229.769 Comparisons

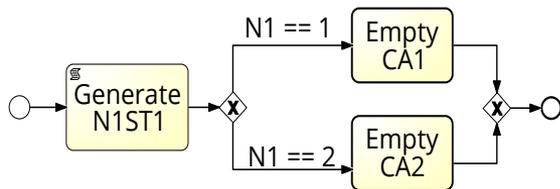


143 structures size ≥ 5

Workload Mix

- Workload Mix Generation Applied -

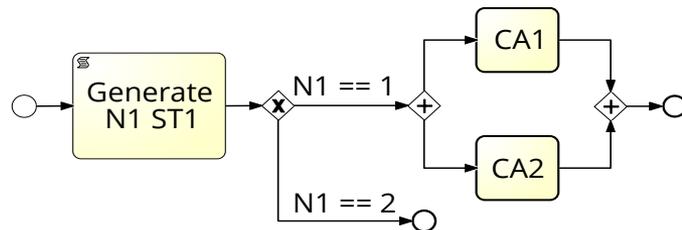
Class 1



Process Models: 1,602
Times: 1,731

38.93%

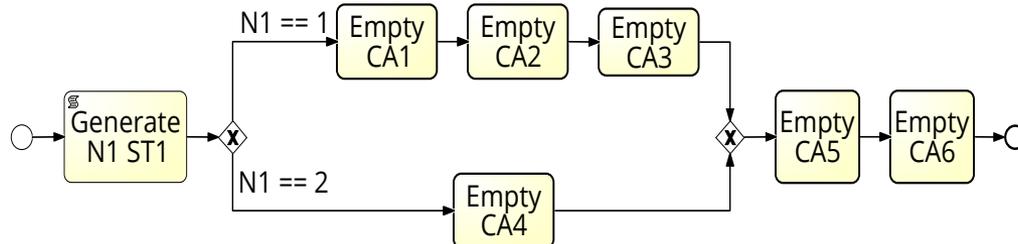
Class 2



Process Models: 953
Times: 2,303

26.92%

Class 3



Process Models: 640
Times: 1,710

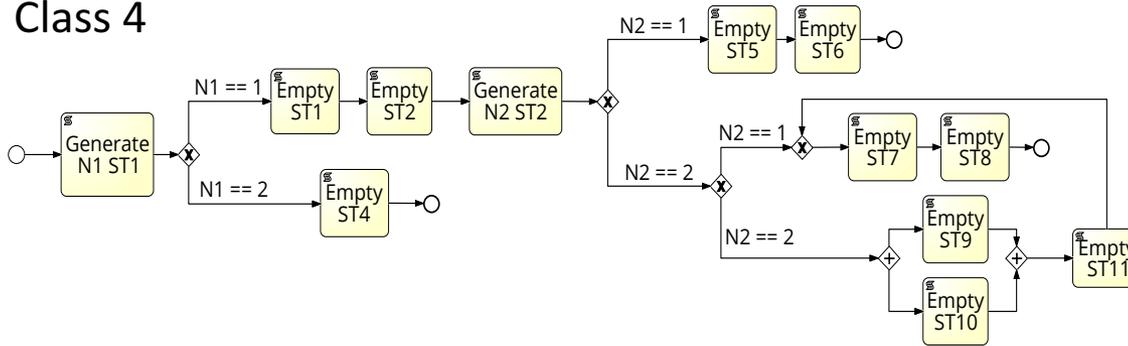
18.49%



Workload Mix

- Workload Mix Generation Applied -

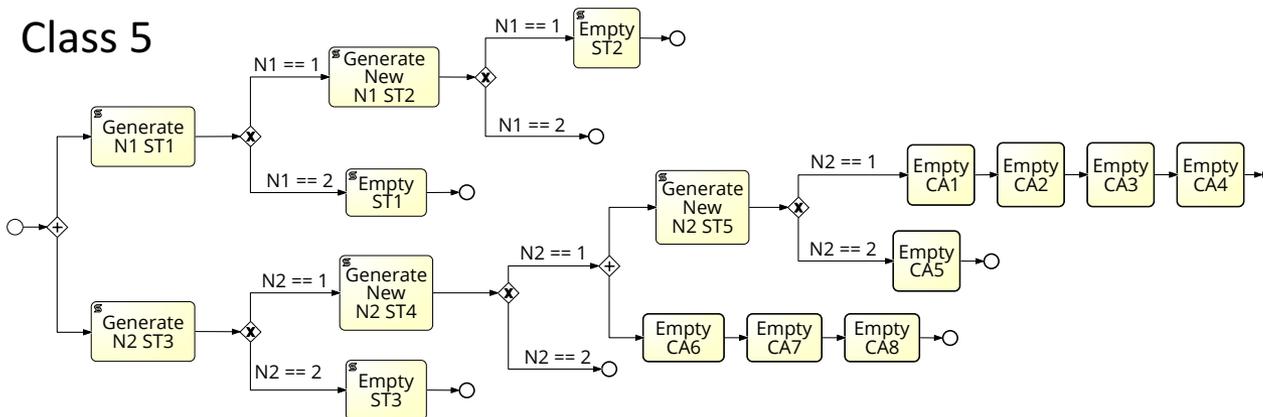
Class 4



| | |
|------------------|-------|
| <u>S1</u> | |
| Process Models : | 640 |
| Times : | 1,710 |
| <u>S2</u> | |
| Process Models : | 309 |
| Times : | 635 |

13.46%

Class 5



| | |
|------------------|-----|
| <u>S1</u> | |
| Process Models : | 130 |
| Times : | 157 |
| <u>S2</u> | |
| Process Models : | 30 |
| Times : | 30 |

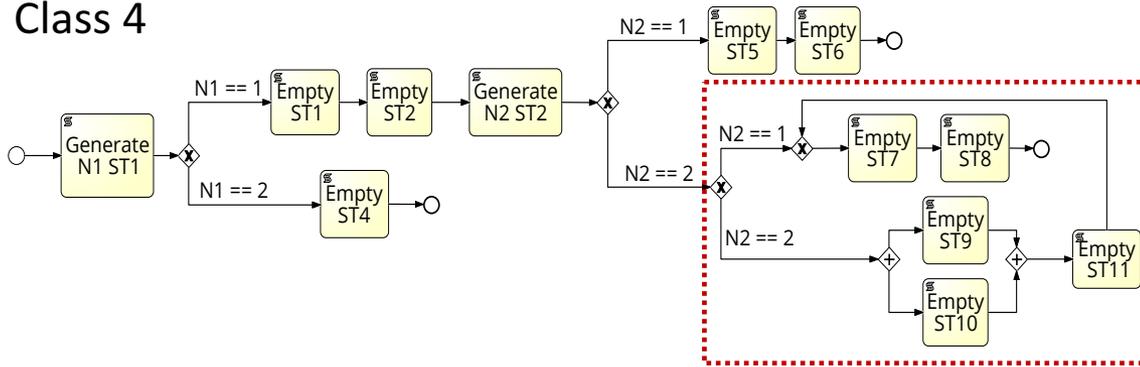
2.19%



Workload Mix

- Workload Mix Generation Applied -

Class 4

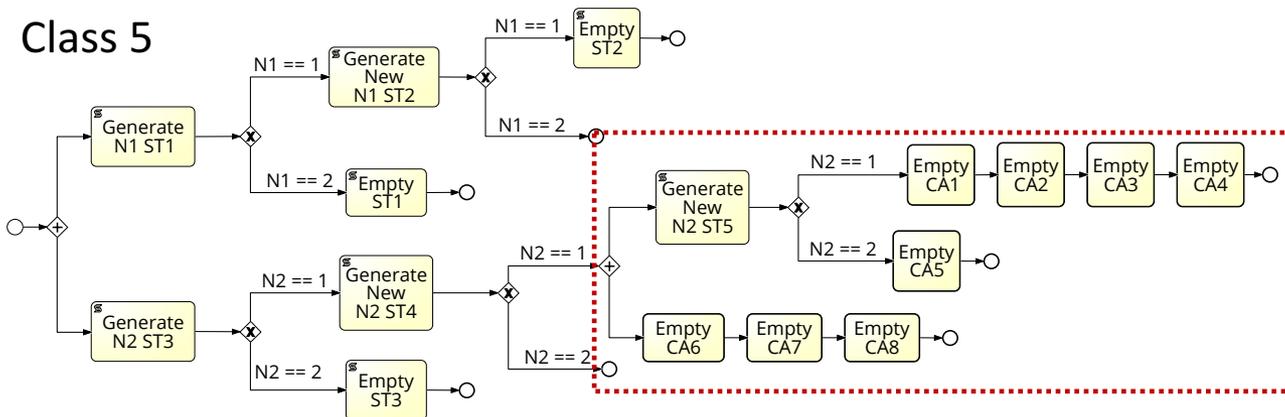


S1
 Process Models : 640
 Times: 1,710

S2
 Process Models: 309
 Times: 635

13.46%

Class 5



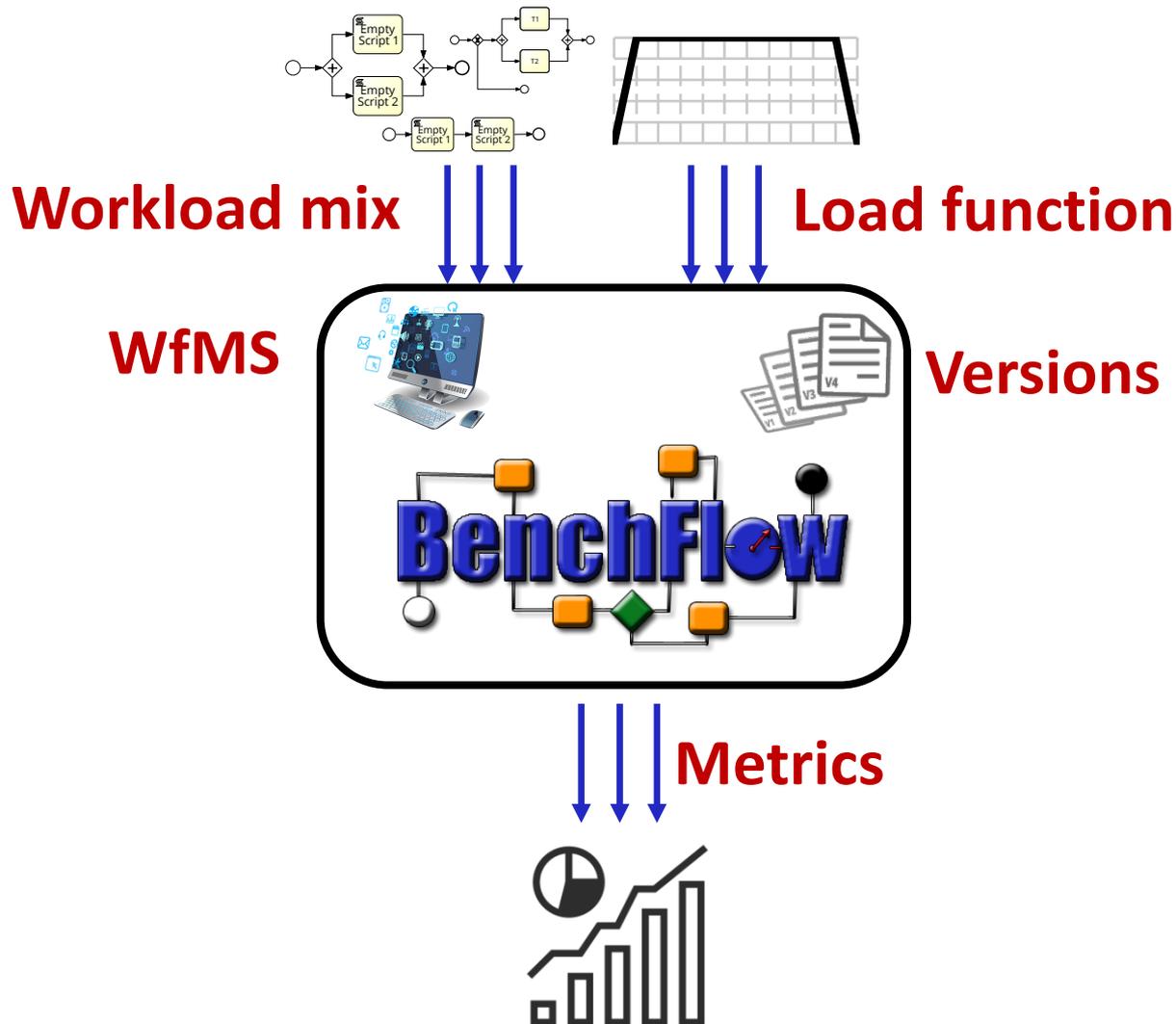
S1
 Process Models : 130
 Times: 157

S2
 Process Models: 30
 Times: 30

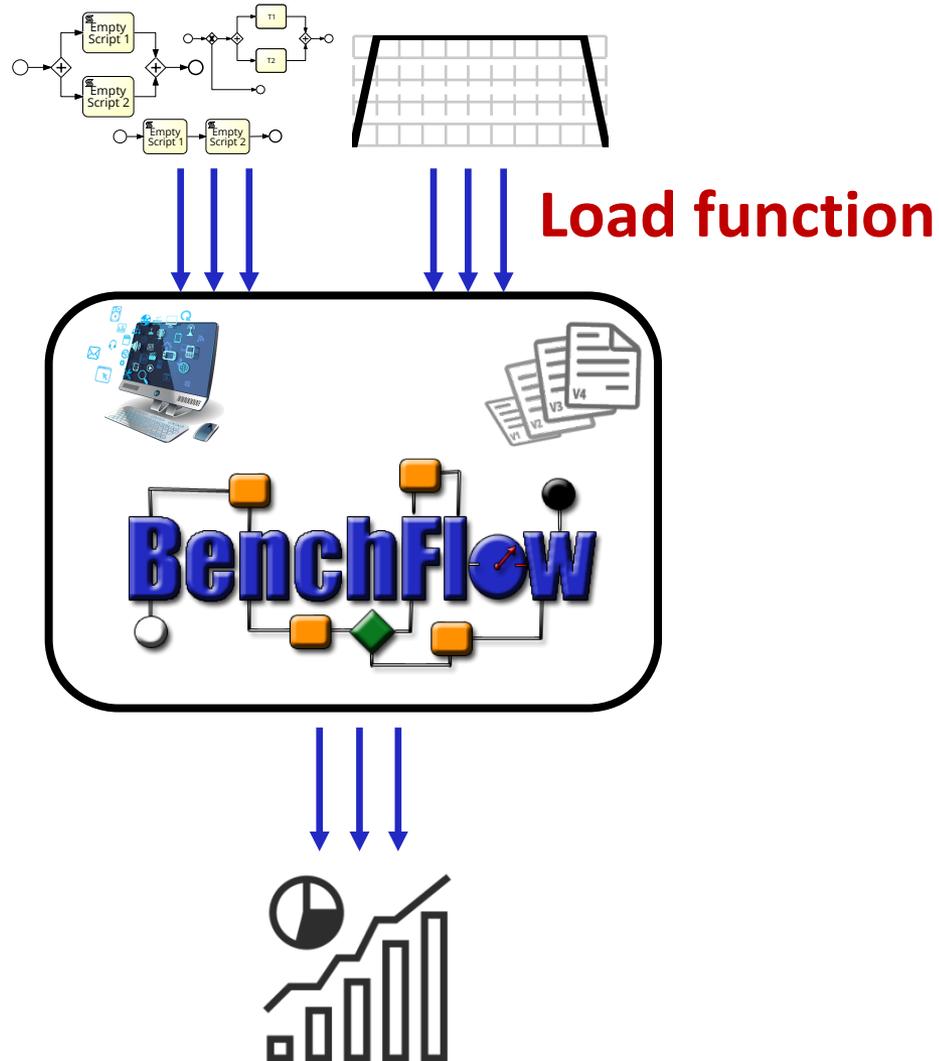
2.19%



Performance testing



Performance testing



Load function

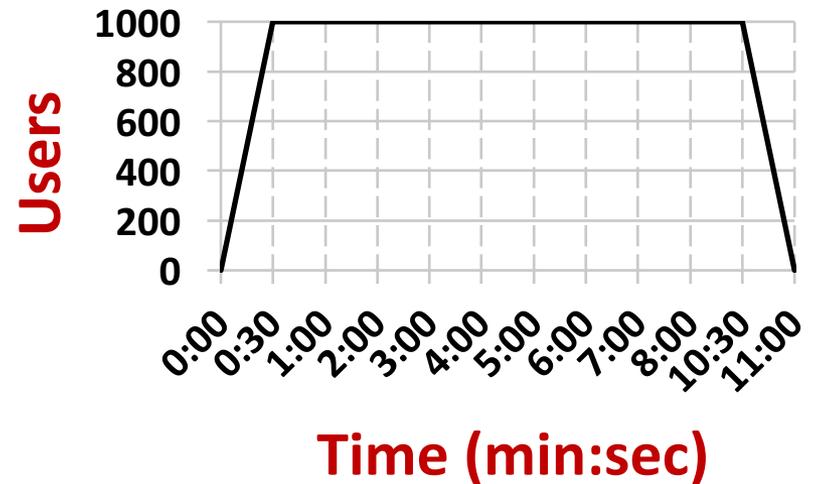


- 50
- 500 users
- 1000

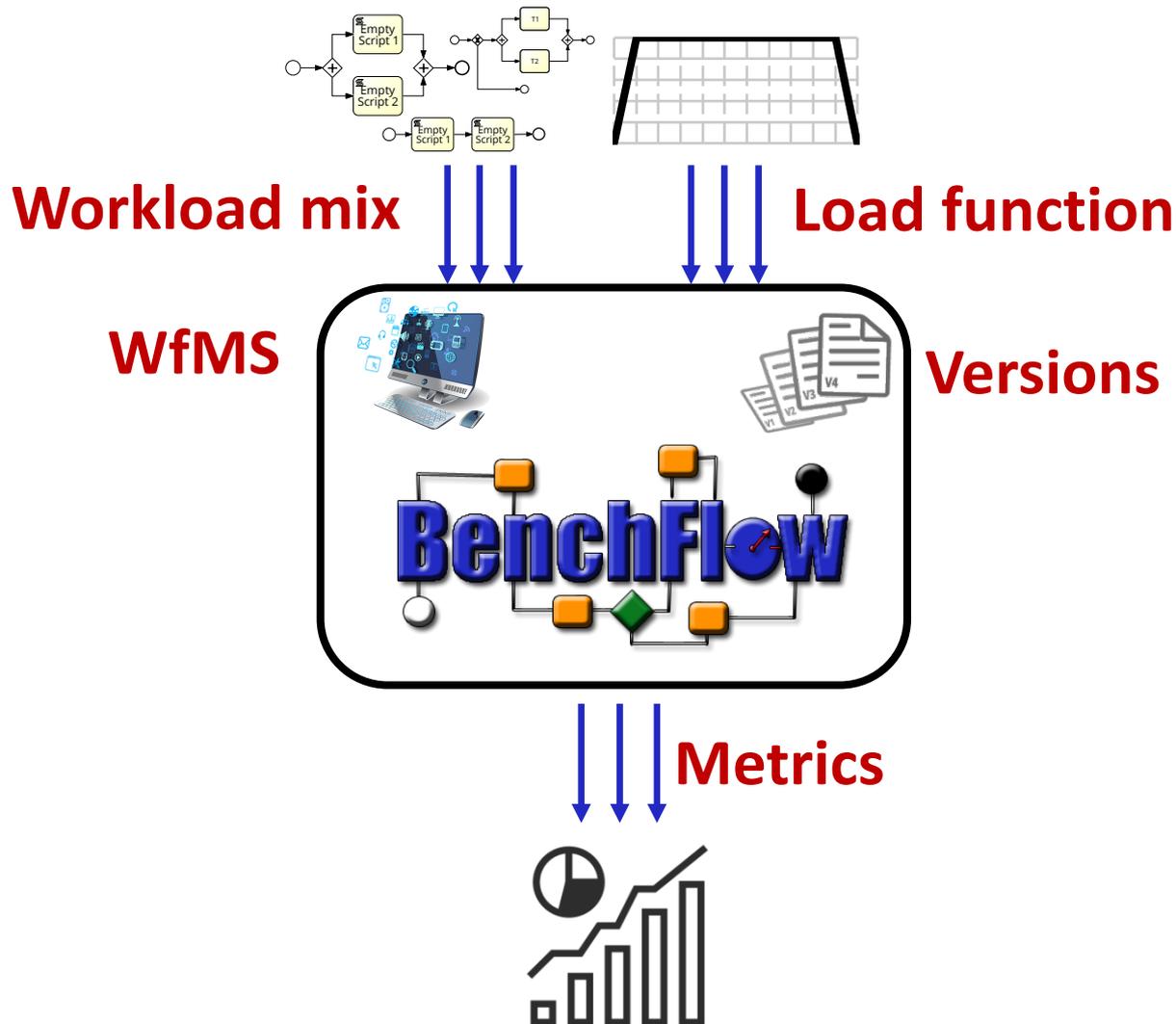


1 second think time

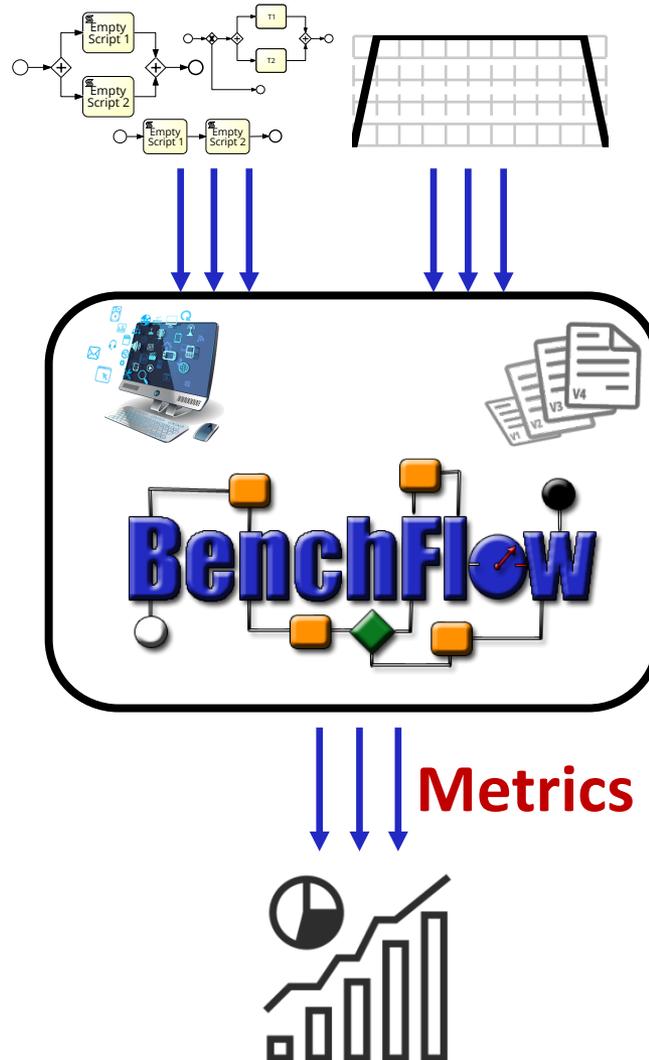
Example for
users = 1000



Performance testing



Performance testing



Metrics

Client-side performance:

Server-side performance:

Resource consumption:



Metrics

Client-side performance:

- Number of requests per second (*#REQ/s*)

Server-side performance:

Resource consumption:



Metrics

Client-side performance:

- Number of requests per second ($\#REQ/s$)

Server-side performance:

- BP (bp_i) instance duration ($D[ms]$)
- Throughput ($T[\# bp_i/s]$)

Resource consumption:



Metrics

Client-side performance:

- Number of requests per second ($\#REQ/s$)

Server-side performance:

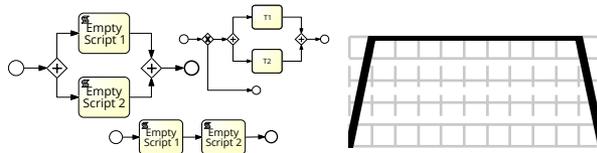
- BP (bp_i) instance duration ($D[ms]$)
- Throughput ($T[\# bp_i/s]$)

Resource consumption:

- CPU consumption ($CPU[\%]$)
- RAM consumption ($RAM[MB]$)



Performance testing

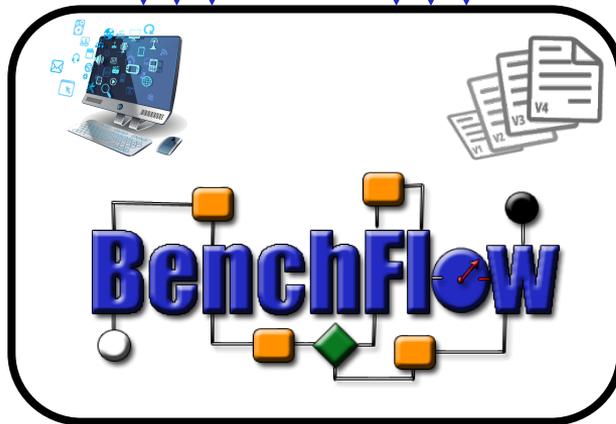


Workload mix

Load function

WfMS

Versions



Metrics

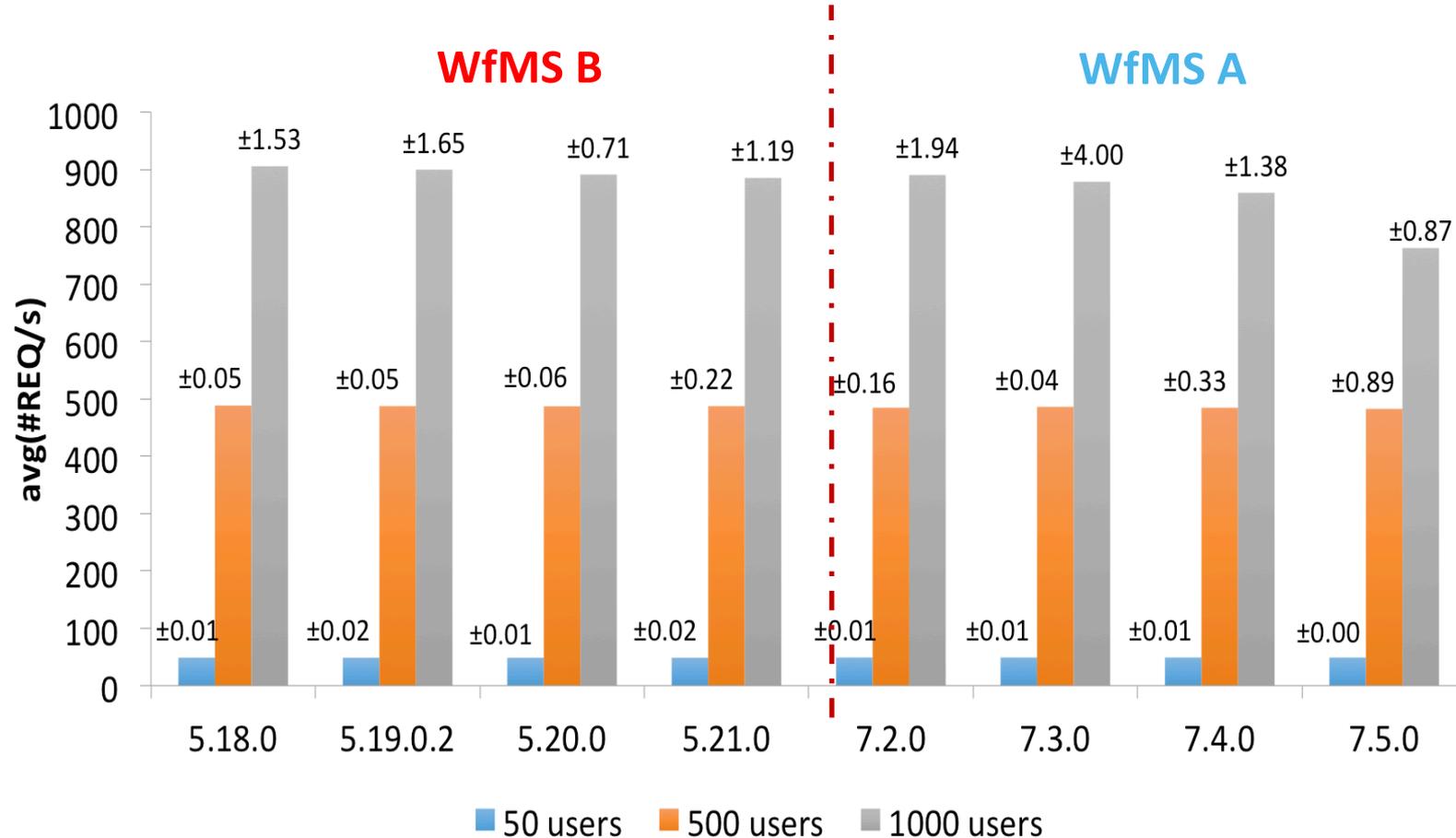
Execute tests and
analyze the results



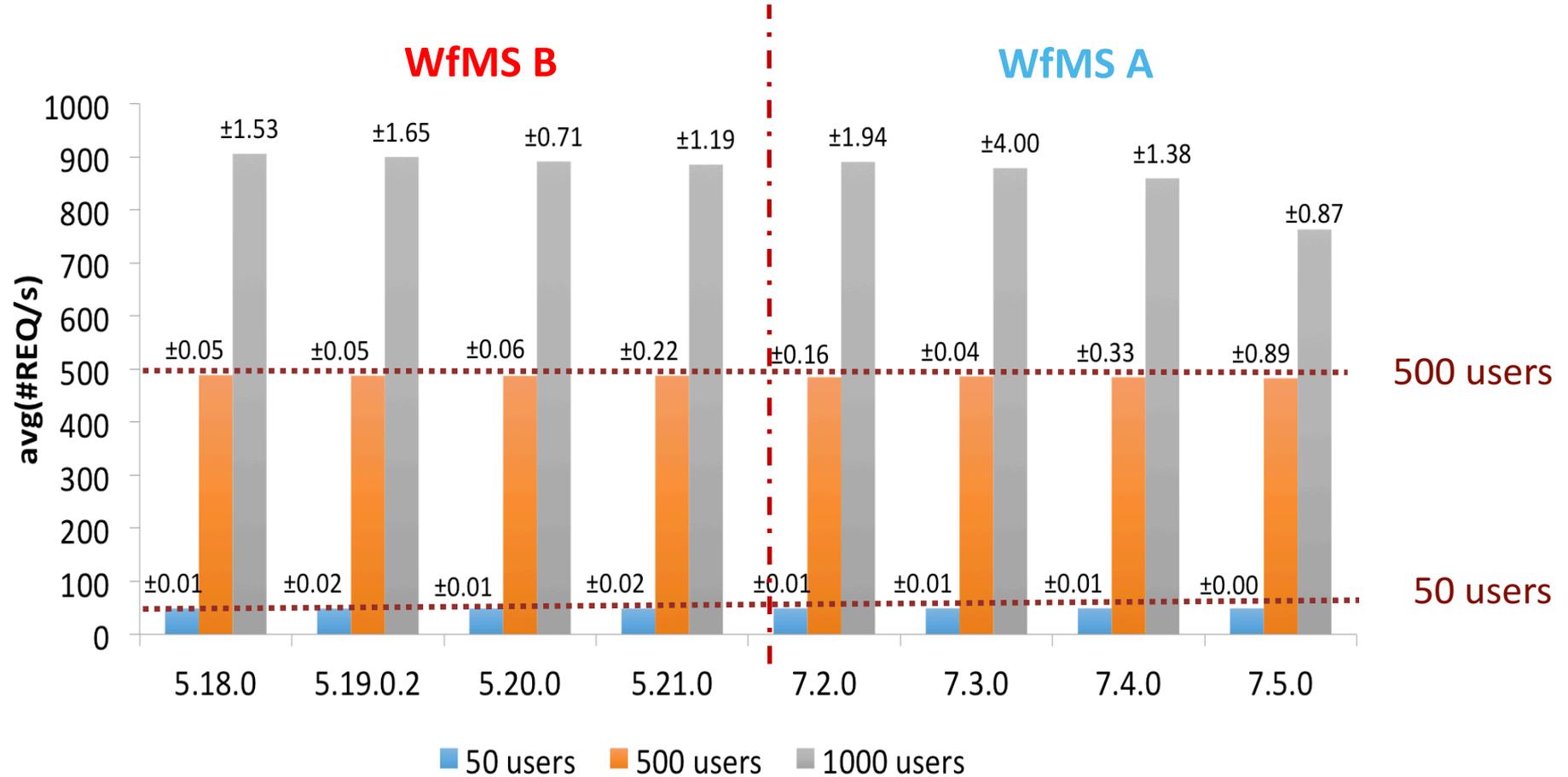
RESULTS

| | | Load | WfMS A | | | | WfMS B | | | |
|----------------------|---------------------------------------|---------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|----------------|
| | | | 7.2.0 | 7.3.0 | 7.4.0 | 7.5.0 | 5.18.0 | 5.19.0.2 | 5.20.0 | 5.21.0 |
| Client-side | $avg(\#REQ/s) \pm ci$ | 50 | 49.13±0.04 | 49.17±0.03 | 49.16±0.02 | 49.09±0.01 | 48.84±0.02 | 48.72±0.05 | 48.66±0.03 | 48.65±0.04 |
| | | 500 | 484.87±0.39 | 486.44±0.10 | 484.84±0.82 | 482.91±2.20 | 488.61±0.12 | 487.72±0.13 | 487.34±0.15 | 487.71±0.55 |
| | | 1'000 | 890.84±4.82 | 879.15±9.94 | 859.81±3.42 | 763.46±2.17 | 906.10±3.81 | 900.14±4.09 | 891.62±1.76 | 885.80±2.97 |
| Server-side | $avg(T) \pm ci$ [#bp _i /s] | 50 | 118.23±0.21 | 119.72±0.17 | 120.08±0.79 | 120.05±0.42 | 119.87±0.07 | 119.82±0.09 | 119.99±0.09 | 119.82±0.14 |
| | | 500 | 1'185.12±0.45 | 1'185.56±0.33 | 1'180.80±4.34 | 1'175.10±0.58 | 1'161.88±0.98 | 1'160.46±1.13 | 1'160.96±1.67 | 1'132.92±2.43 |
| | | 1'000 | 2'121.35±6.23 | 2'130.90±9.50 | 2'087.26±2.72 | 1'849.88±5.17 | 2'182.78±8.25 | 2'202.37±6.37 | 2'189.36±4.83 | 1'974.01±10.59 |
| | $wavg(D)$ [ms] | 50 | 1.03 | 1.08 | 1.12 | 1.24 | 6.53 | 6.75 | 6.90 | 7.19 |
| | | 500 | 0.87 | 0.91 | 1.07 | 1.14 | 5.08 | 5.27 | 5.56 | 5.65 |
| | | 1'000 | 0.93 | 0.99 | 1.05 | 1.15 | 5.18 | 5.34 | 5.39 | 5.43 |
| Resource Consumption | $wavg(CPU)$ [%] | 50 | 1.66 | 1.33 | 1.33 | 1.35 | 3.01 | 1.59 | 1.63 | 1.66 |
| | | 500 | 8.07 | 6.26 | 7.03 | 6.95 | 8.82 | 7.55 | 9.59 | 9.50 |
| | | 1'000 | 11.39 | 8.33 | 8.81 | 8.79 | 12.05 | 12.74 | 12.78 | 11.84 |
| | $wavg(RAM)$ [MB] | 50 | 637.28 | 590.82 | 634.79 | 648.67 | 1'764.83 | 2'430.69 | 2'620.57 | 2'455.11 |
| | | 500 | 885.10 | 860.70 | 886.53 | 866.37 | 8'686.66 | 8'653.84 | 8'633.66 | 8'583.11 |
| | | 1'000 | 970.61 | 957.47 | 978.97 | 971.43 | 9'549.54 | 9'749.74 | 9'509.81 | 9'350.82 |
| Server-side | $wavg(D)$ [ms] | bp_i | | | | | | | | |
| | | Class 1 | 1.09 | 1.14 | 1.24 | 1.35 | 9.28 | 9.52 | 9.79 | 10.02 |
| | | Class 2 | 1.34 | 1.41 | 1.54 | 1.67 | 6.09 | 6.24 | 6.42 | 6.57 |
| | | Class 3 | 2.31 | 2.43 | 2.64 | 2.88 | 16.73 | 17.16 | 17.64 | 18.06 |
| | | Class 4 | 1.13 | 1.20 | 1.30 | 1.42 | 3.53 | 3.62 | 3.73 | 3.81 |
| | | Class 5 | 1.93 | 2.03 | 2.21 | 2.40 | 22.49 | 23.07 | 23.72 | 24.28 |

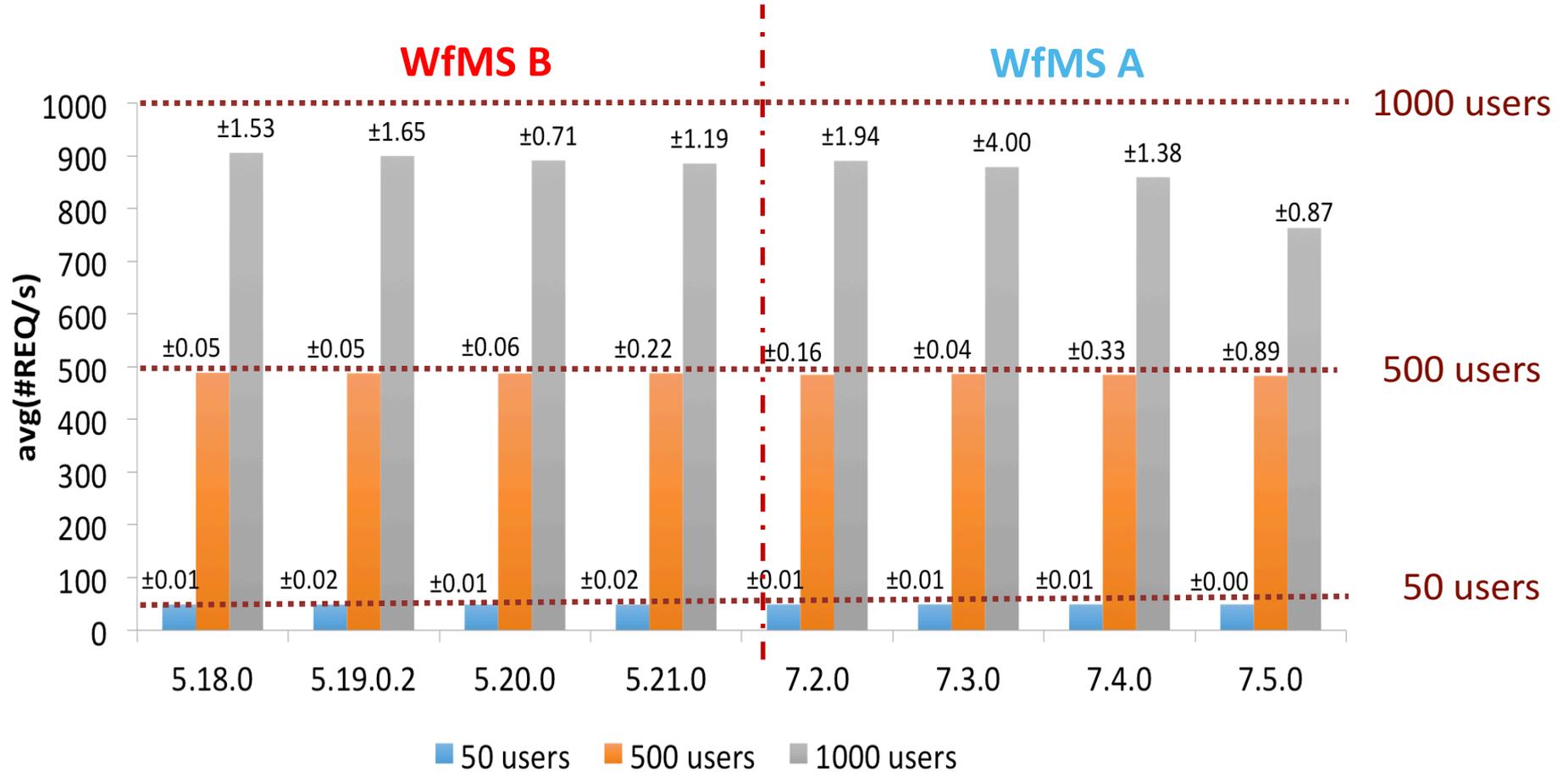
REQUESTS PER SECOND METRICS RESULTS



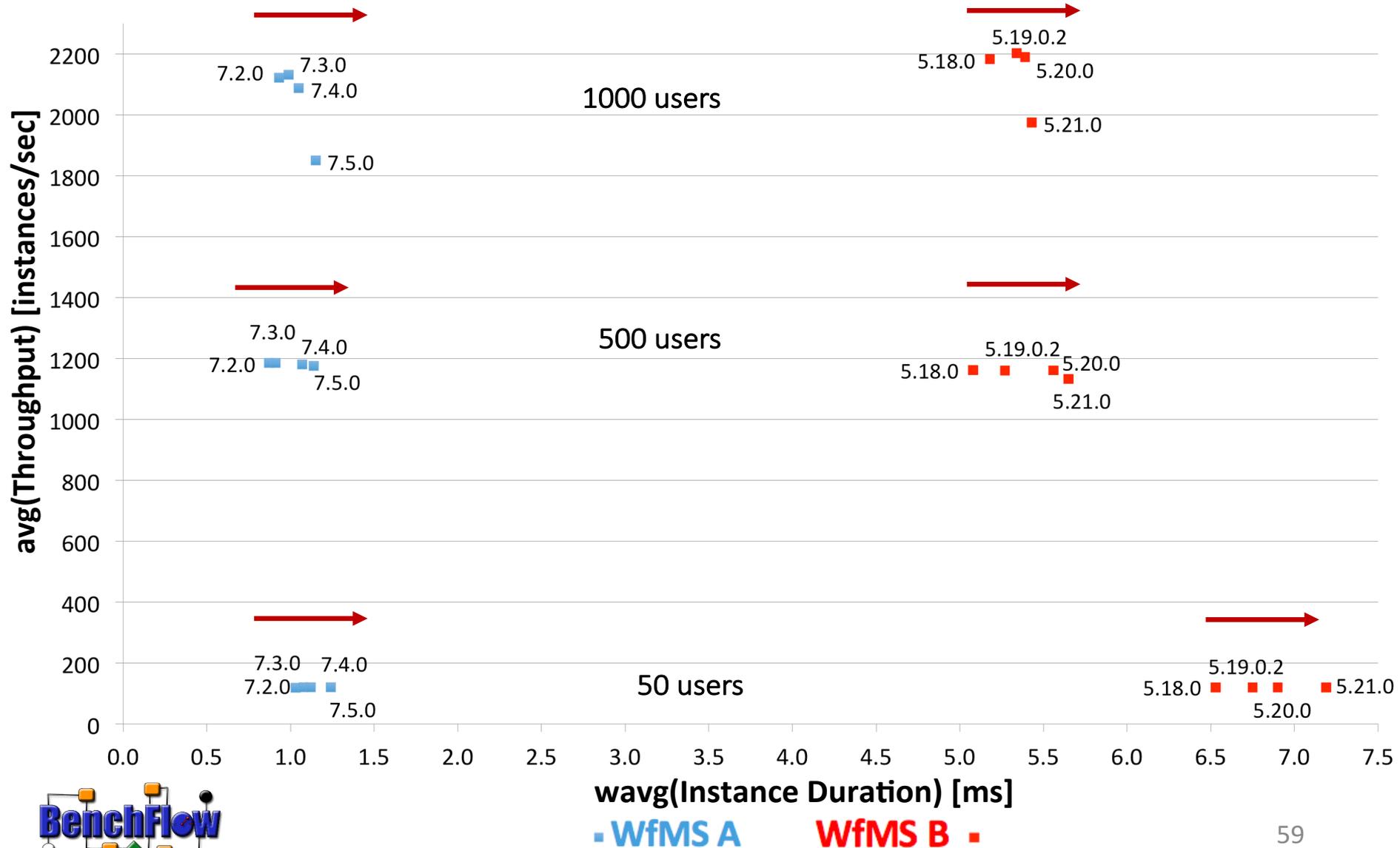
REQUESTS PER SECOND METRICS RESULTS



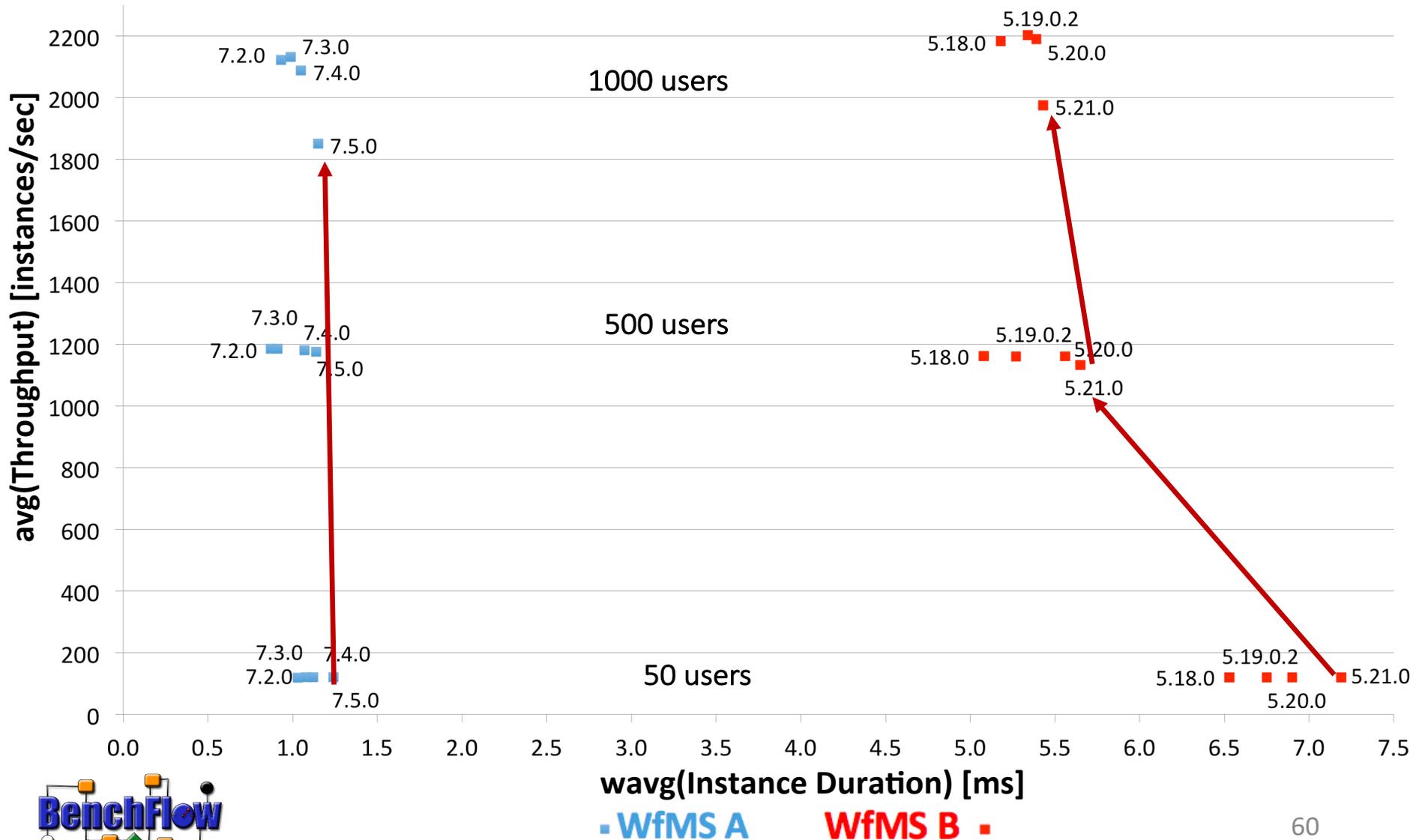
REQUESTS PER SECOND METRICS RESULTS



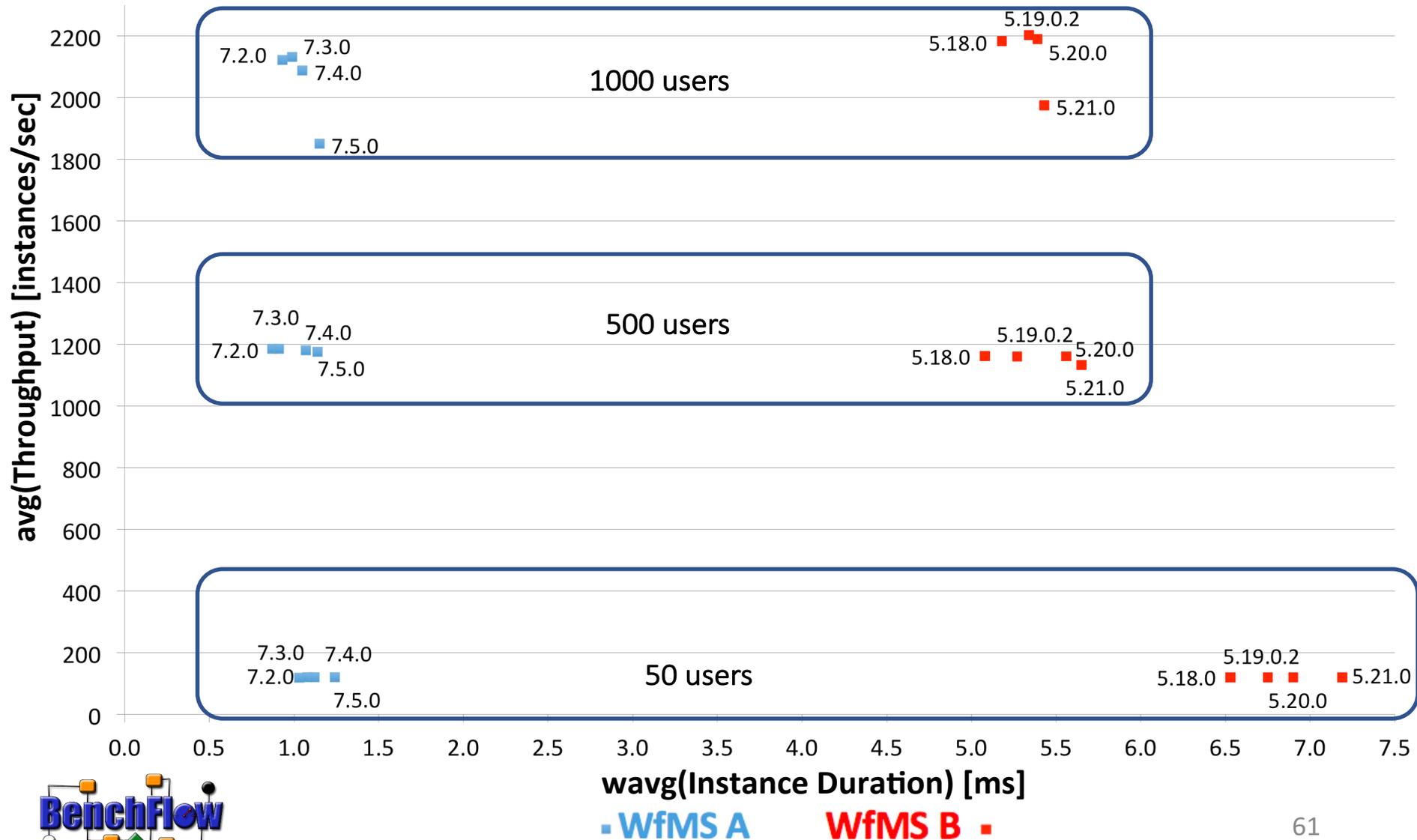
SERVER-SIDE PERFORMANCE METRICS RESULTS



SERVER-SIDE PERFORMANCE METRICS RESULTS

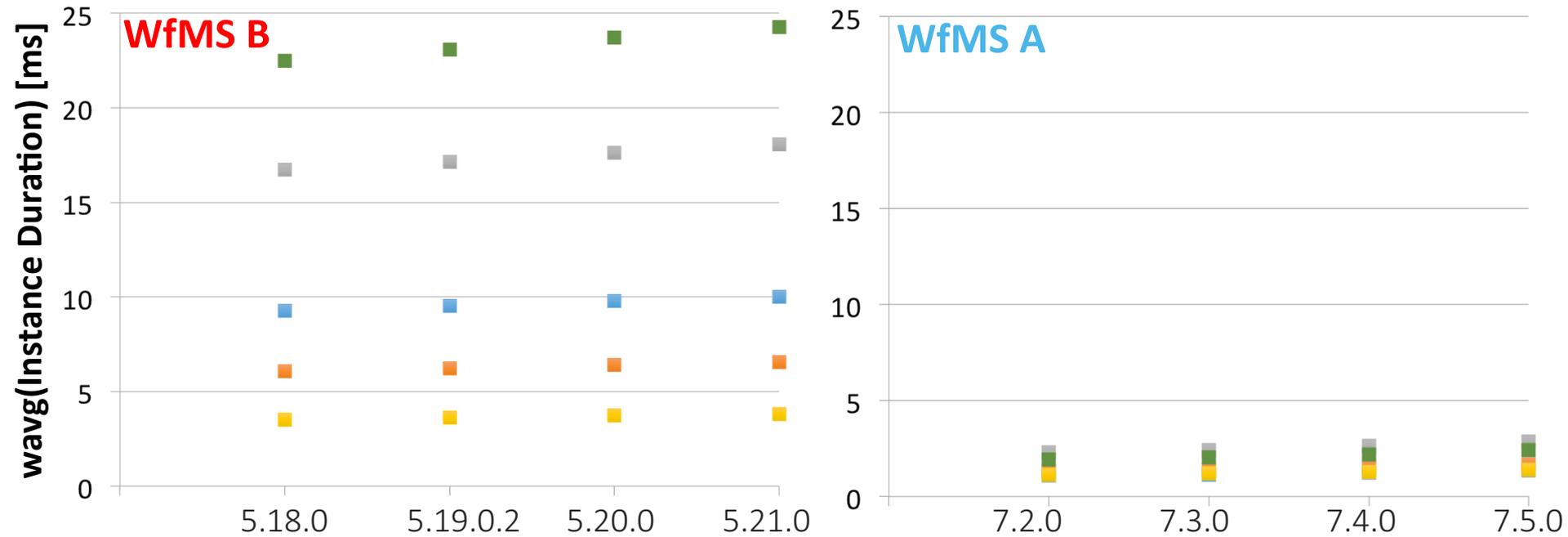


SERVER-SIDE PERFORMANCE METRICS RESULTS



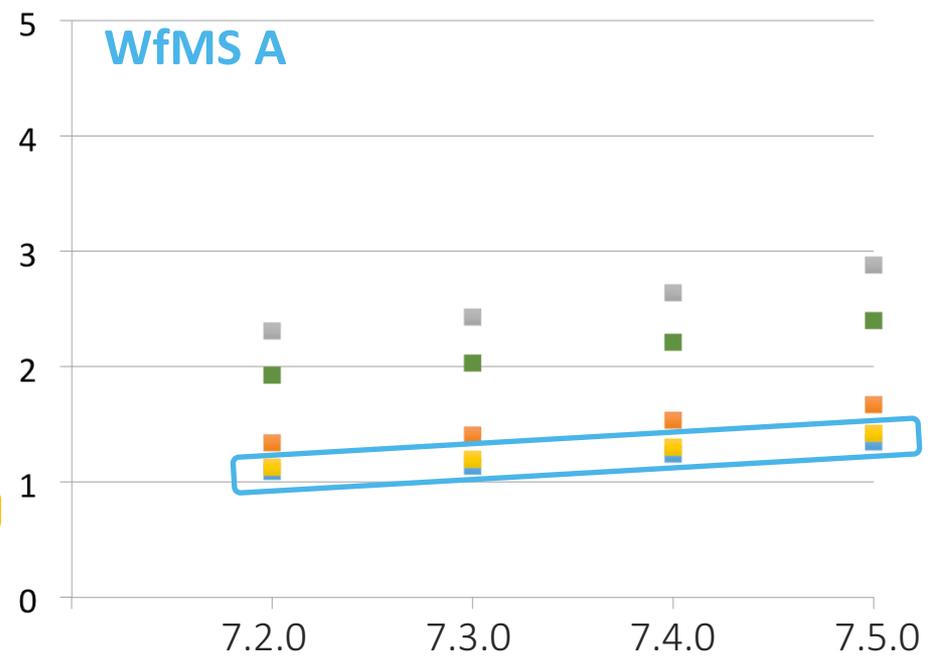
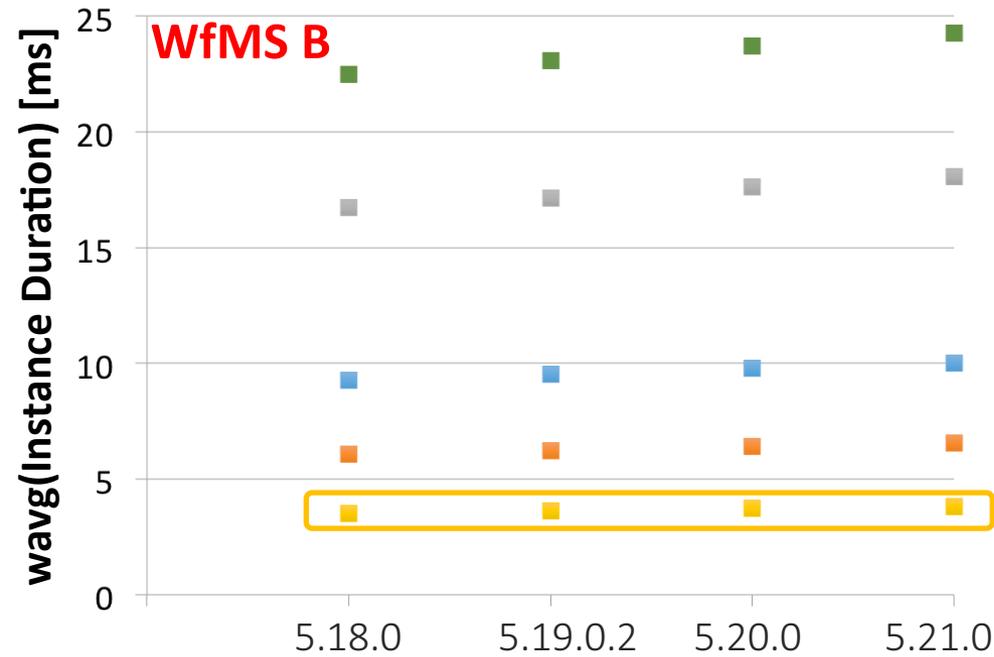
INSTANCE DURATION PER MODEL CLASS

Class 1 Class 2 Class 3 Class 4 Class 5

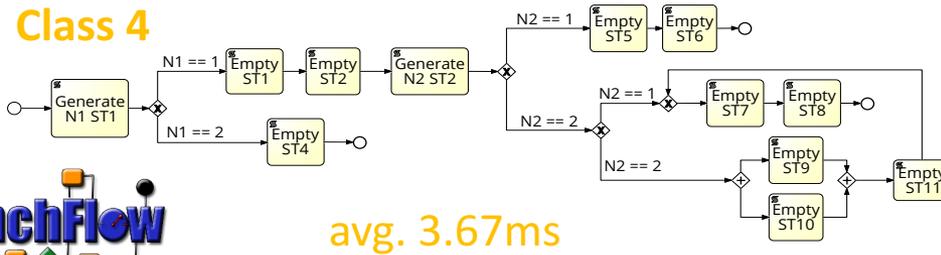


INSTANCE DURATION PER MODEL CLASS

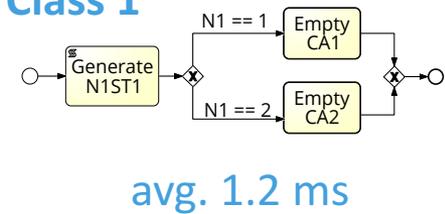
■ Class 1
 ■ Class 2
 ■ Class 3
 ■ Class 4
 ■ Class 5



Class 4

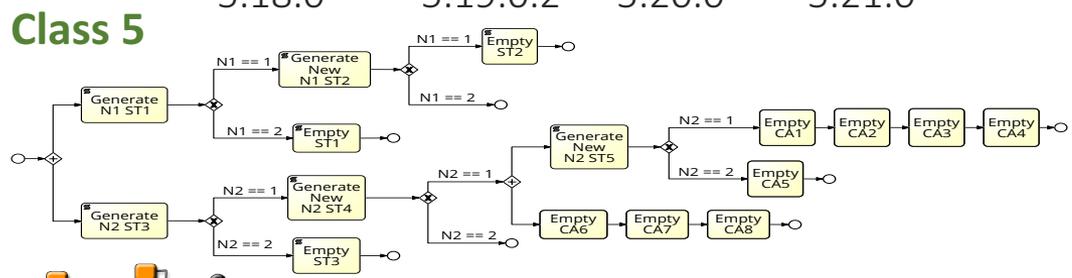
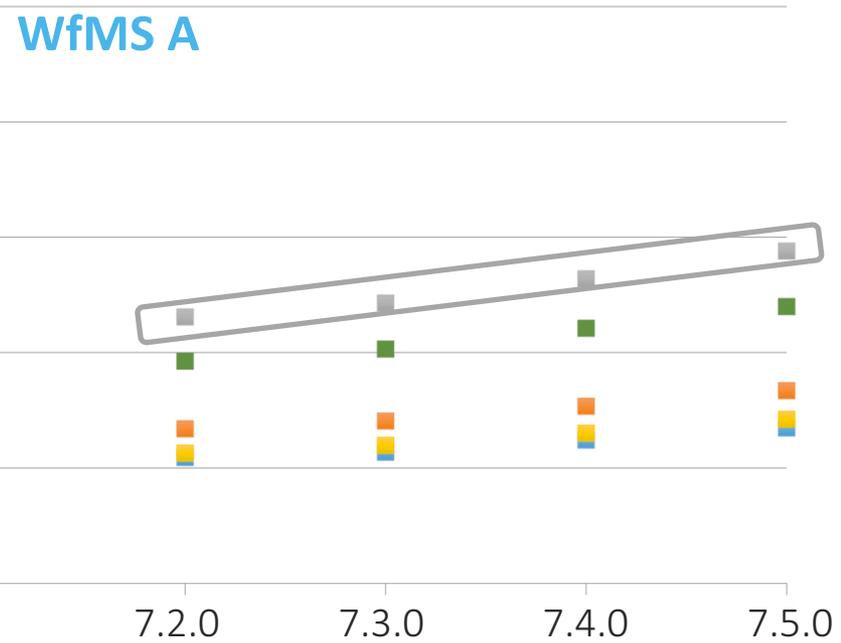
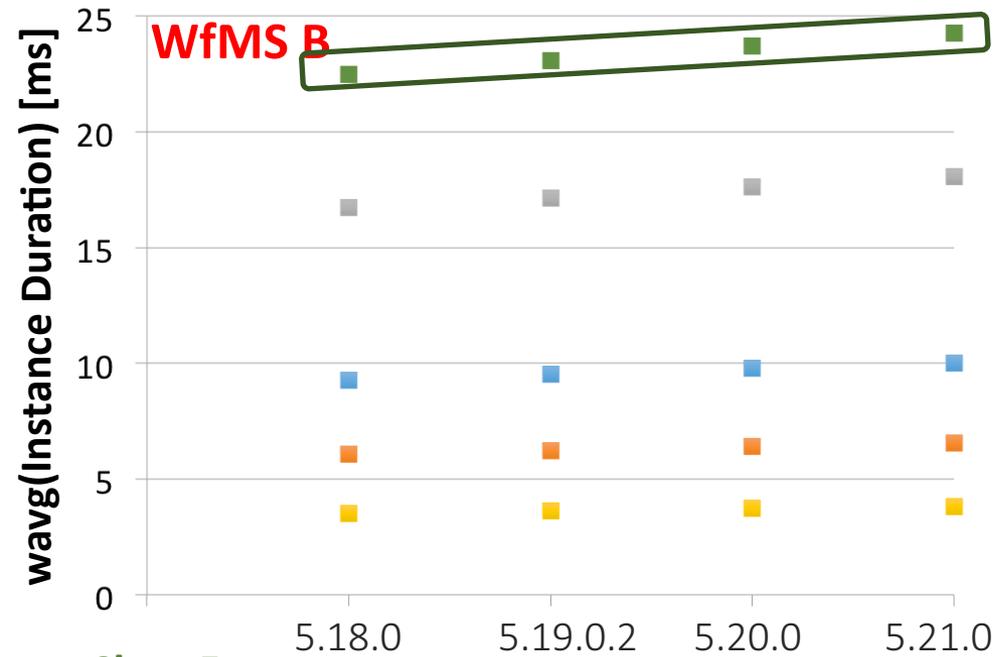


Class 1

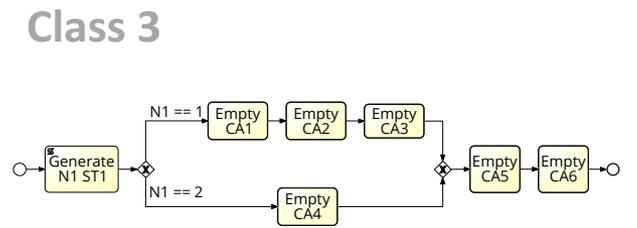


INSTANCE DURATION PER MODEL CLASS

■ Class 1
 ■ Class 2
 ■ Class 3
 ■ Class 4
 ■ Class 5



avg. 23.4 ms

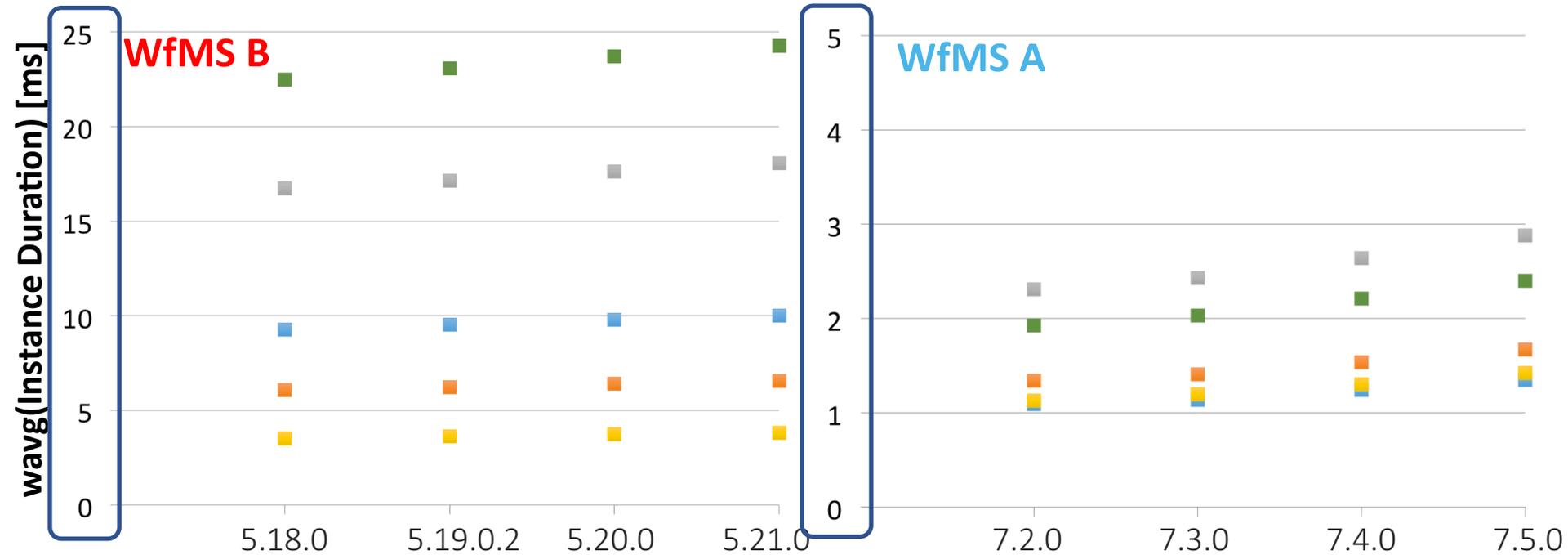


avg. 2.56 ms



INSTANCE DURATION PER MODEL CLASS

■ Class 1 ■ Class 2 ■ Class 3 ■ Class 4 ■ Class 5



Call activities:

avg. 5.06ms avg. 0.70ms

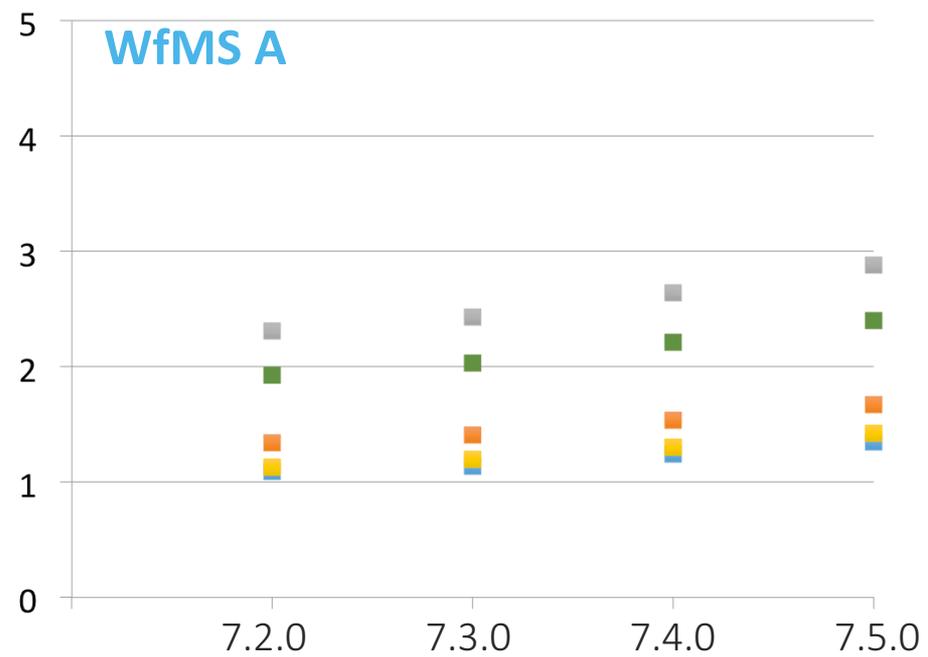
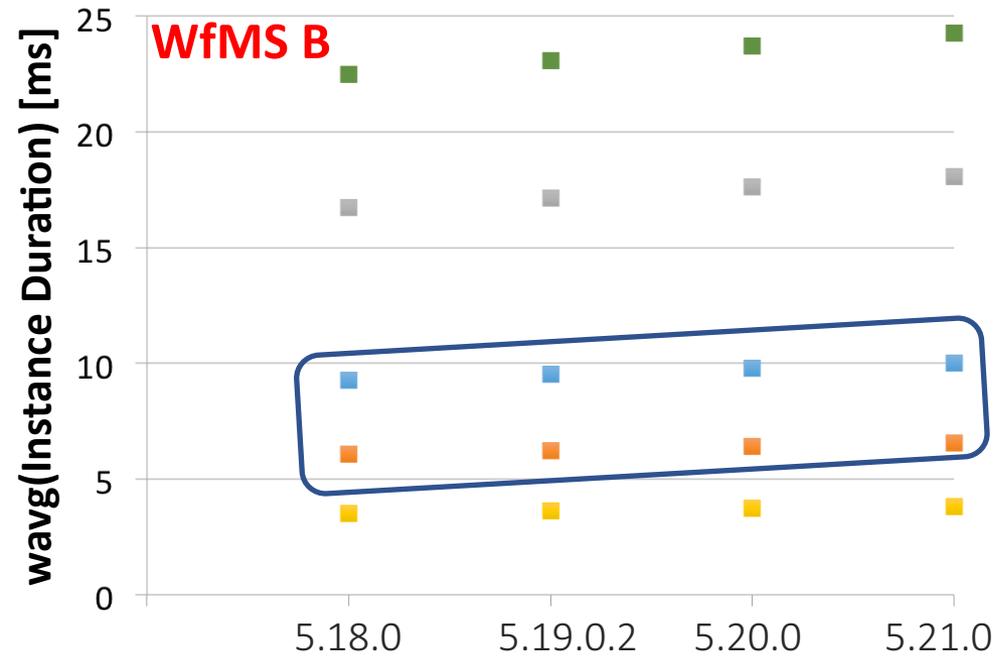
Start events:

avg. 0.56 ms avg. 0.04ms

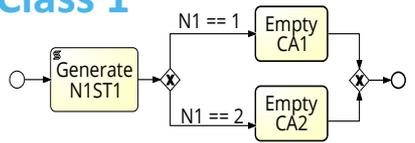


INSTANCE DURATION PER MODEL CLASS

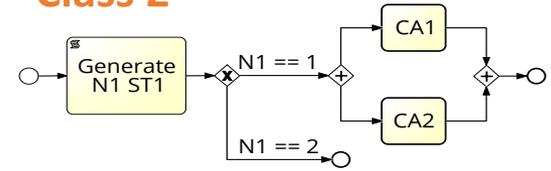
■ Class 1
 ■ Class 2
 ■ Class 3
 ■ Class 4
 ■ Class 5



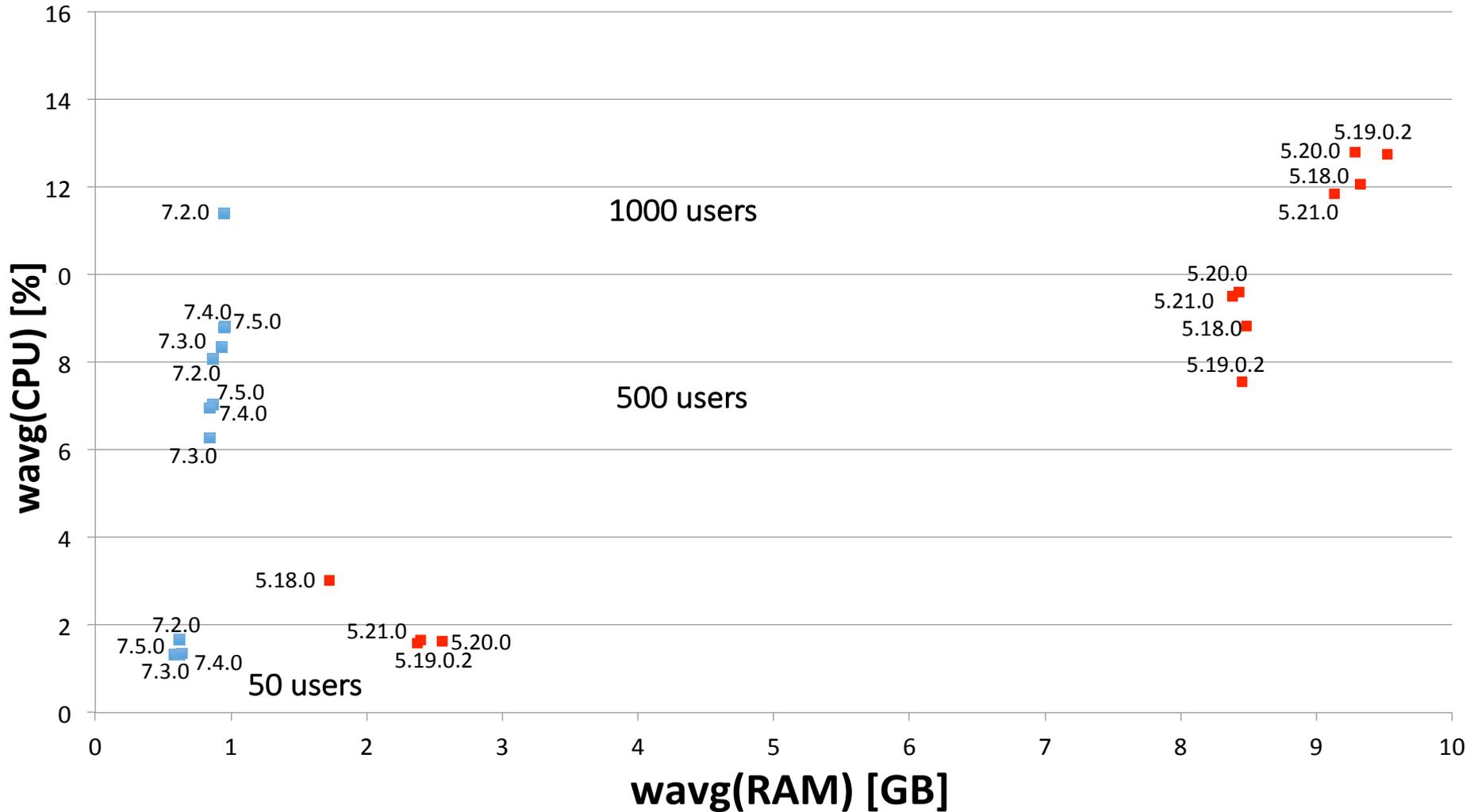
Class 1



Class 2

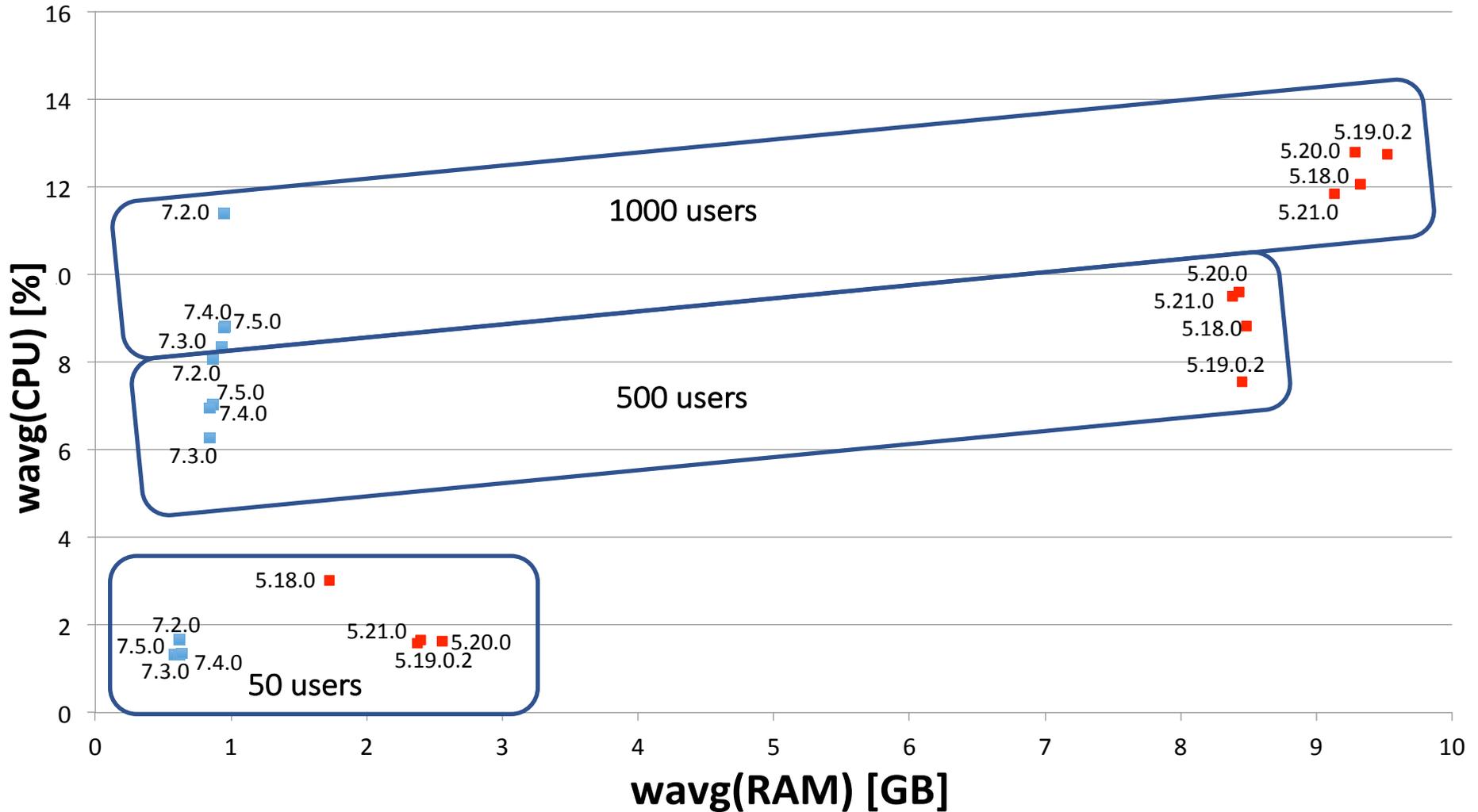


RESOURCE CONSUMPTION METRICS RESULTS



■ WfMS A ■ WfMS B

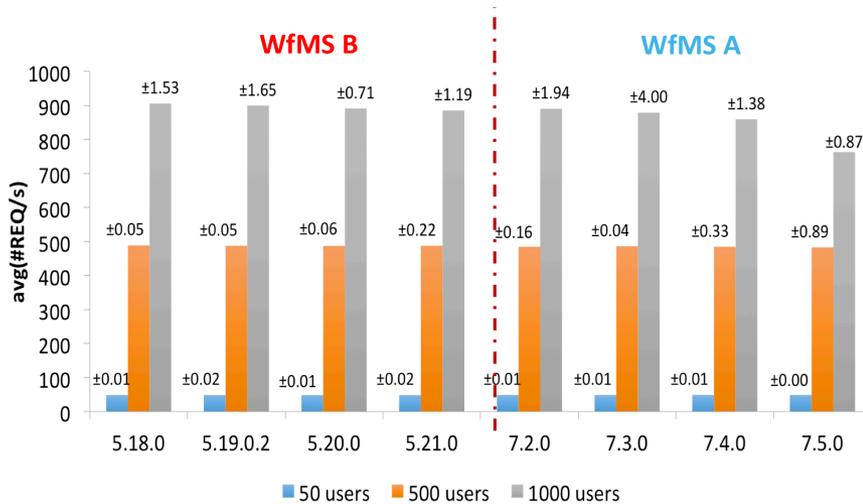
RESOURCE CONSUMPTION METRICS RESULTS



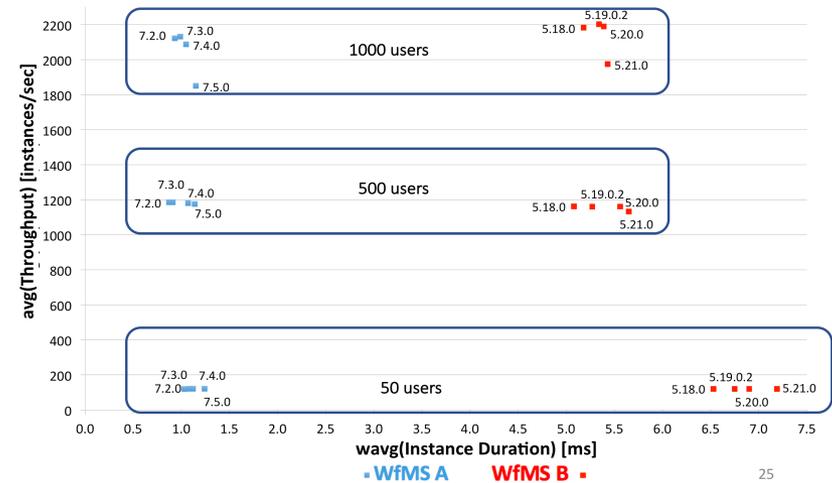
■ WfMS A ■ WfMS B

TAKE AWAY

Newer version

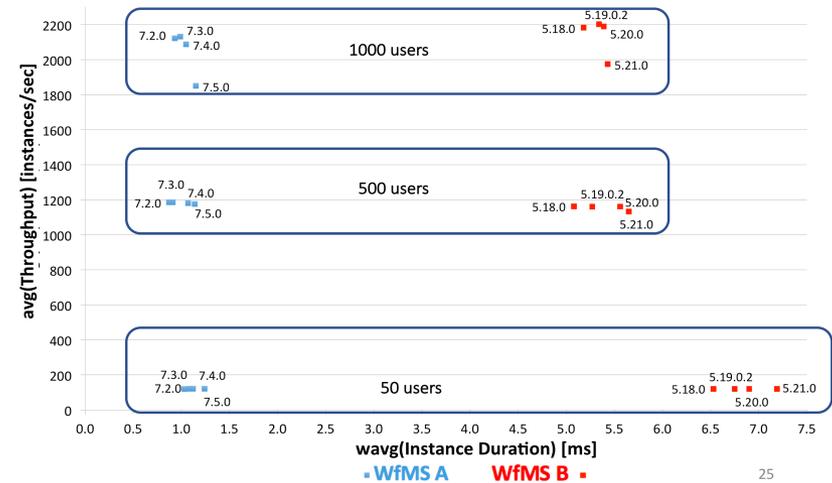
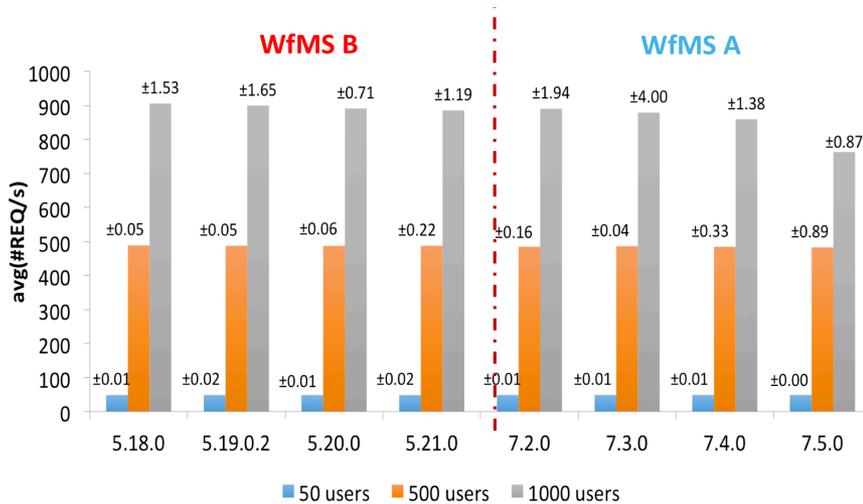


Better performance



TAKE AWAY

Newer version \neq Better performance



Differences are more evident with more users!

Future work



WfMSs?

Load function?

Workload mix?

Metrics?

- more WfMSs and with different configurations settings
- event logs for realistic load functions
- domain specific models
- support for events / human tasks / external interaction
- maybe new metrics of interest

Highlights

2 . 5 . 0
BREAKING . FEATURE . FIX

- breaking change**: incompatible API changes
- new feature**: add backwards-compatible functionality
- bug fix**: make backwards-compatible bug fix

4

