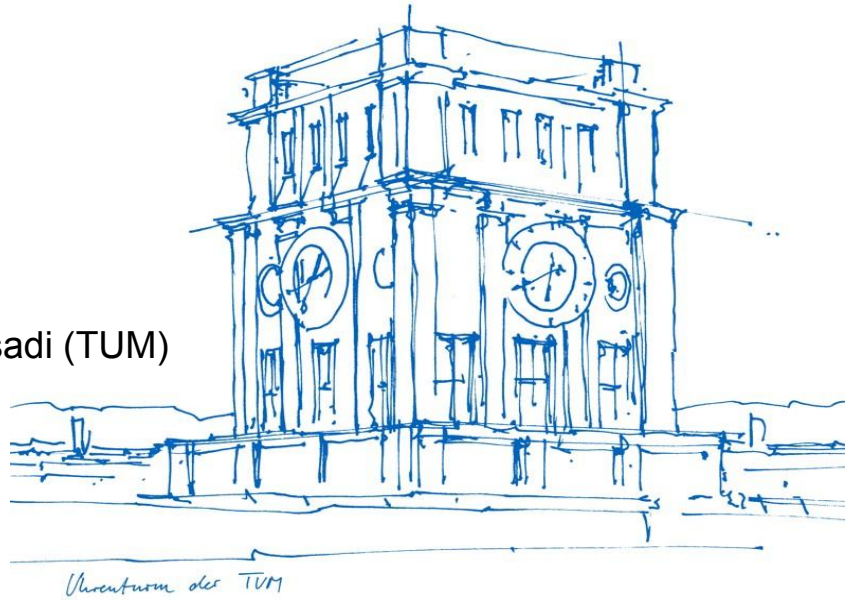
Chair of Communication Networks
TUM School of Computation, Information and Technology
Technical University of Munich

VM Selection for Financial Exchanges in the Cloud

Master Thesis Final Presentation

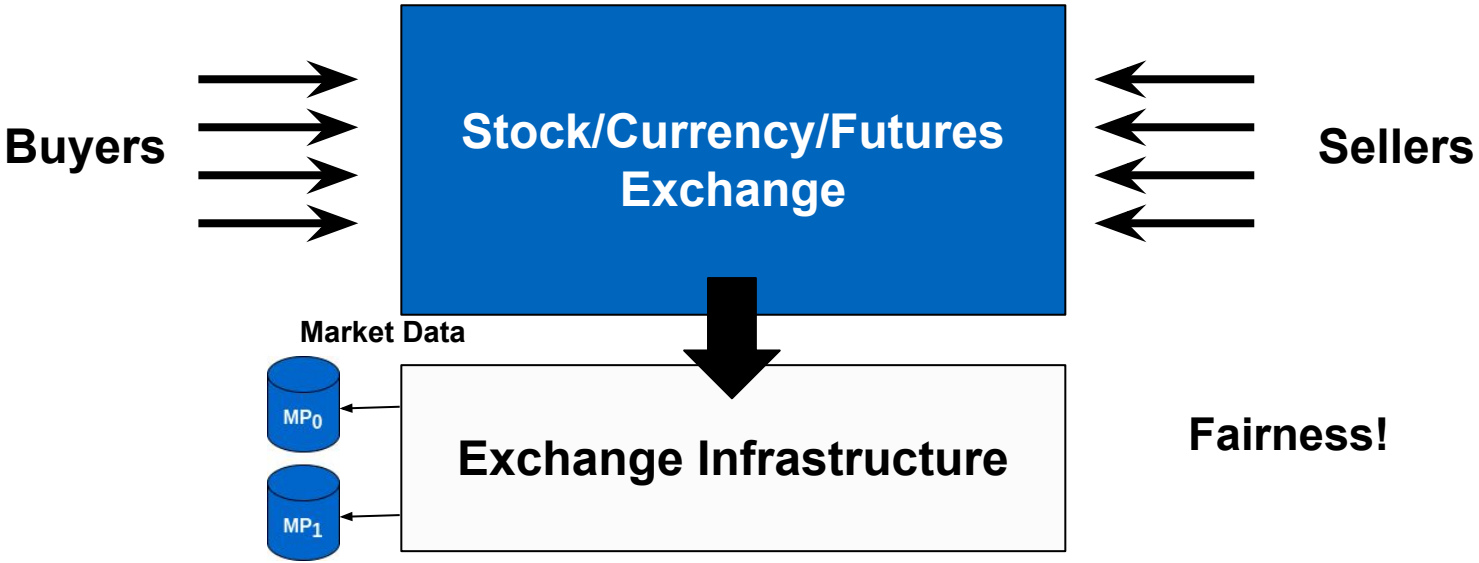
Student: Daniel Duclos-Cavalcanti

Supervisors: Muhammad Haseeb (NYU), Navidreza Asadi (TUM)



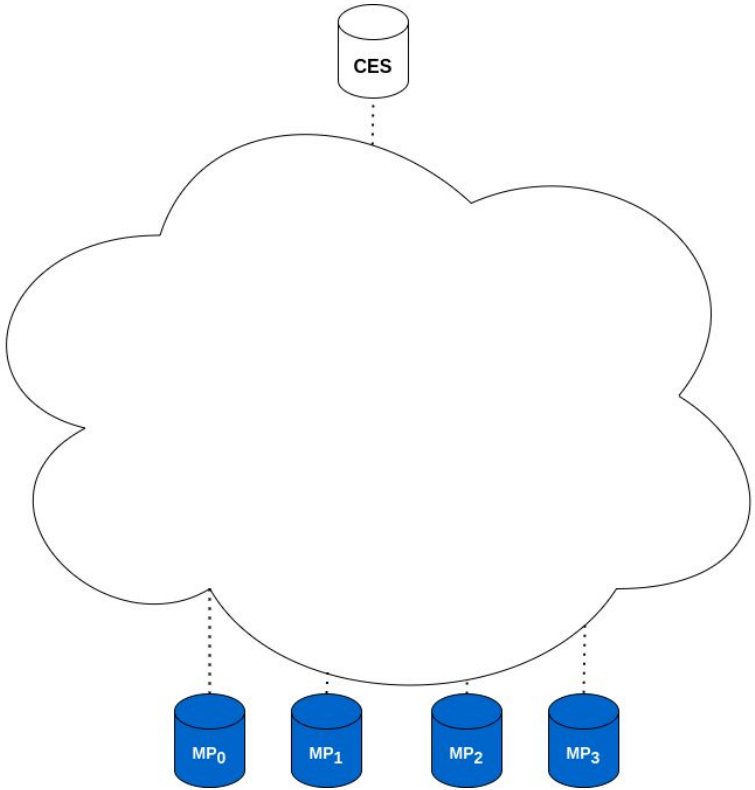
Background: Financial Exchanges and the Cloud

- What is a Financial Exchange?
- Migration: on-premise => public cloud
- **Cloud Medium: i) High Variance ii) ~~Mechanisms~~ for fair delivery**



State-of-the-Art: Jasper[1]: Scalable Exchange in the Cloud

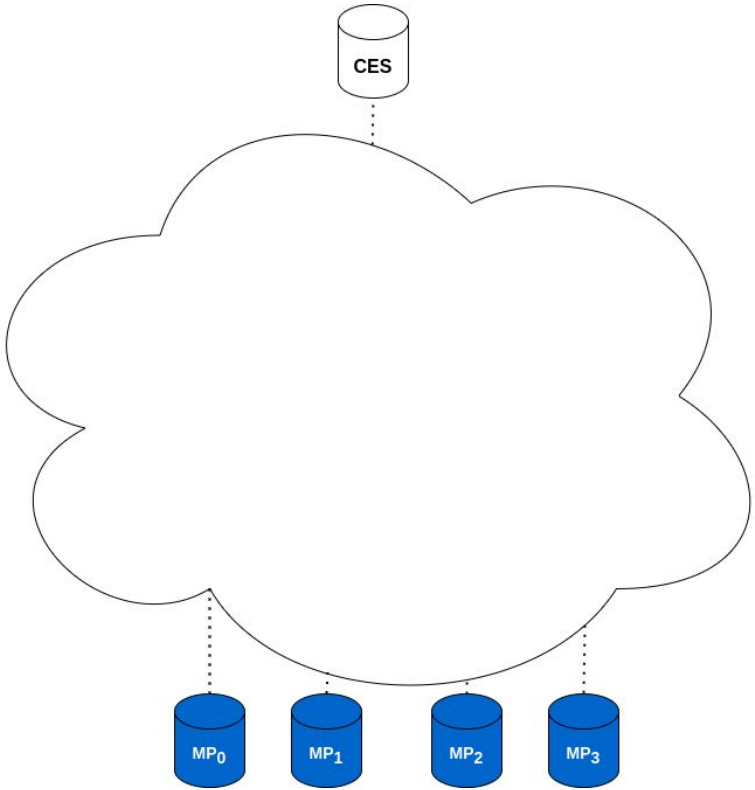
- Scalable, Fair, and Performant Multicast
- Market Data Dissemination



[1]Jasper: https://arxiv.org/abs/2402.09527, https://sigcomm24posterdemo.hotcrp.com/paper/
[2]Hyugens: https://www.usenix.org/system/files/conference/nsdi18/nsdi18-geng.pdf

State-of-the-Art: Jasper[1]: Scalable Exchange in the Cloud

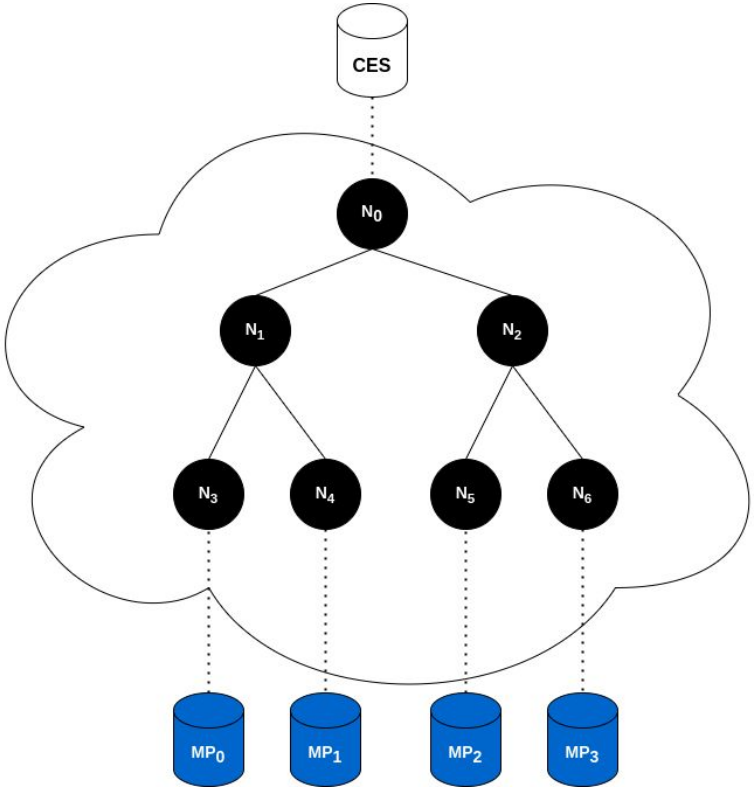
- Scalable, Fair, and Performant Multicast
- Market Data Dissemination
- Techniques:



[1]Jasper: https://arxiv.org/abs/2402.09527, https://sigcomm24posterdemo.hotcrp.com/paper/
[2]Hyugens: https://www.usenix.org/system/files/conference/nsdi18/nsdi18-geng.pdf

State-of-the-Art: Jasper[1]: Scalable Exchange in the Cloud

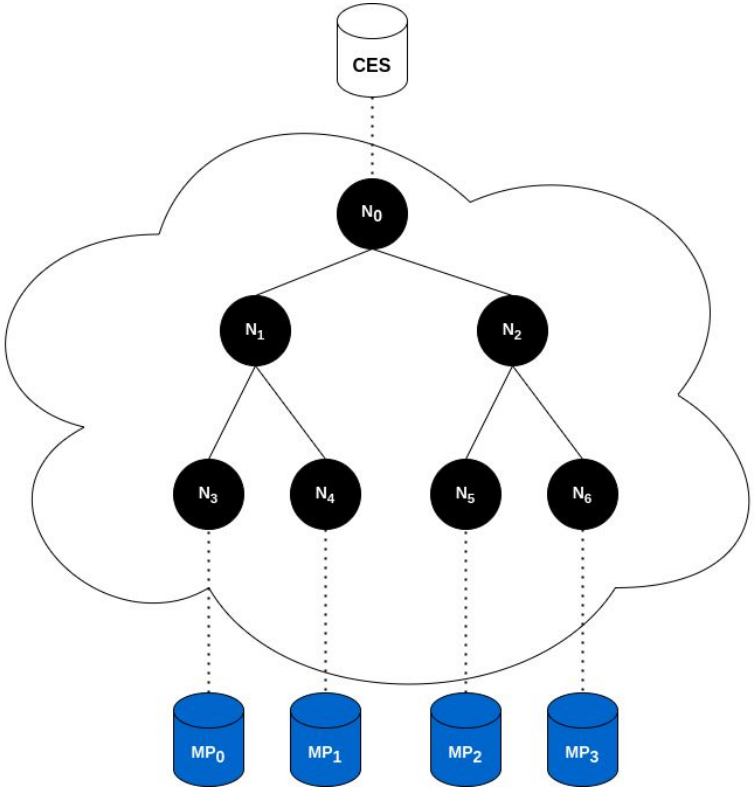
- Scalable, Fair, and Performant Multicast
- Market Data Dissemination
- Techniques:
  - Overlay Multicast Tree



[1]Jasper: https://arxiv.org/abs/2402.09527, https://sigcomm24posterdemo.hotcrp.com/paper/
[2]Hyugens: https://www.usenix.org/system/files/conference/nsdi18/nsdi18-geng.pdf

State-of-the-Art: Jasper[1]: Scalable Exchange in the Cloud

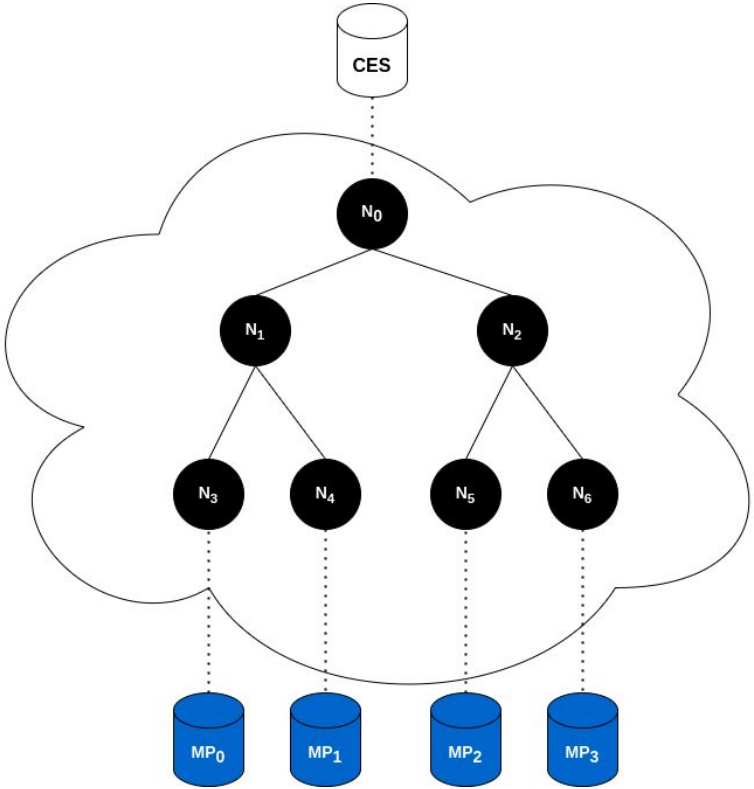
- Scalable, Fair, and Performant Multicast
- Market Data Dissemination
- Techniques:
  - Overlay Multicast Tree
  - Clock-Synchronization[2]



[1]Jasper: https://arxiv.org/abs/2402.09527, https://sigcomm24posterdemo.hotcrp.com/paper/
[2]Hyugens: https://www.usenix.org/system/files/conference/nsdi18/nsdi18-geng.pdf

State-of-the-Art: Jasper[1]: Scalable Exchange in the Cloud

- Scalable, Fair, and Performant Multicast
- Market Data Dissemination
- Techniques:
  - Overlay Multicast Tree
  - Clock-Synchronization[2]
  - Hold-and-Release

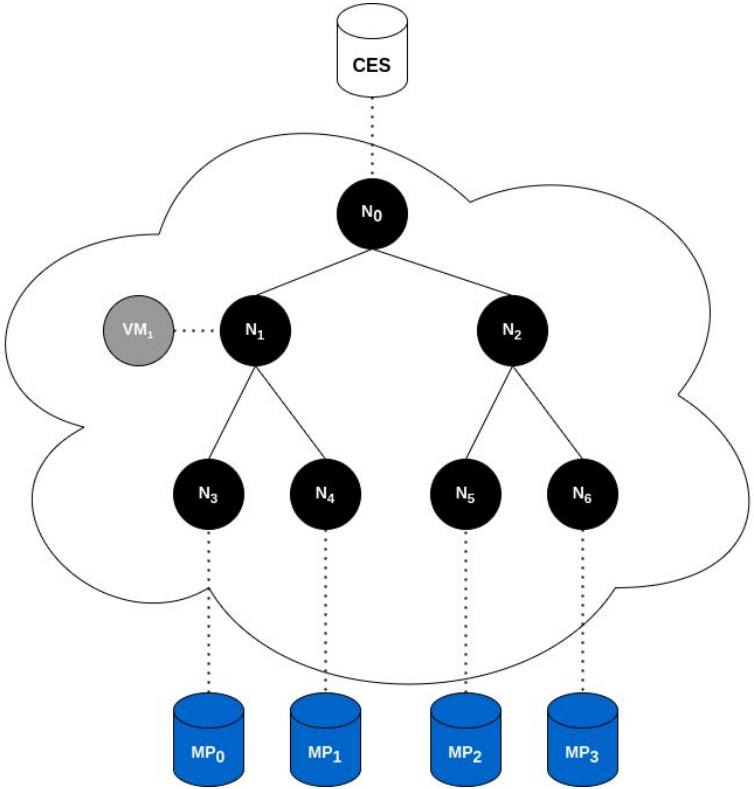


[1]Jasper: https://arxiv.org/abs/2402.09527, https://sigcomm24posterdemo.hotcrp.com/paper/
[2]Hyugens: https://www.usenix.org/system/files/conference/nsdi18/nsdi18-geng.pdf

State-of-the-Art: Jasper[1]: Scalable Exchange in the Cloud

- Scalable, Fair, and Performant Multicast
- Market Data Dissemination
- Techniques:
  - Overlay Multicast Tree
  - Clock-Synchronization[2]
  - Hold-and-Release
- Opportunity:
  - “Stragglers”
  - **Underperforming VMs**

Stragglers affect deadlines!

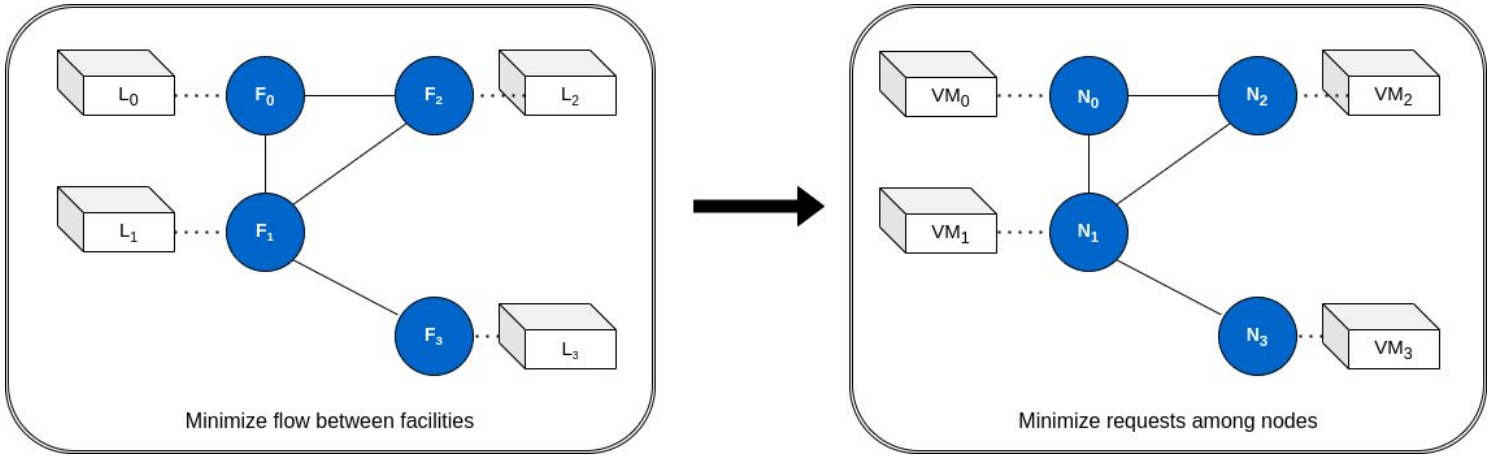


[1]Jasper: https://arxiv.org/abs/2402.09527, https://sigcomm24posterdemo.hotcrp.com/paper/
[2]Hyugens: https://www.usenix.org/system/files/conference/nsdi18/nsdi18-geng.pdf

State-of-the-Art: LemonDrop[1]

- Method of Selecting/Scheduling VMs on a cluster
- Framed QAP => NP-Hard => Scalability issues

Disproportionally Robust for Jasper's use-case



[1]LemonDrop: https://searchworks.stanford.edu/view/14423035

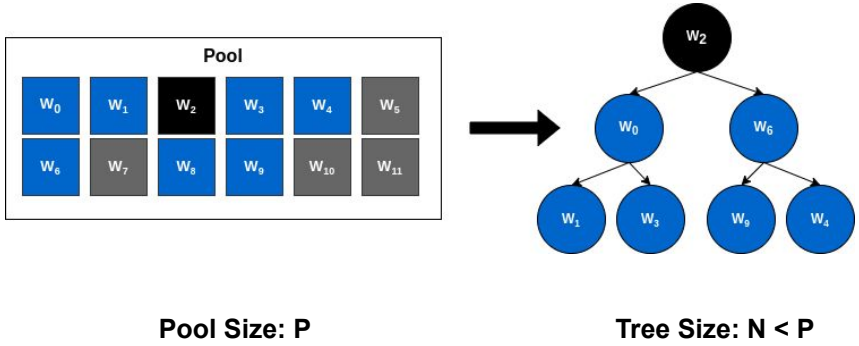
Objectives

Develop lightweight VM selection heuristic

- Minimize “Straggler” Effect
- Lightweight and Scalable

TreeBuilder[1]: Testbench and Design

- **Goal**: VM Selection Method to minimize *“stragglers”*
- **Testbench**:



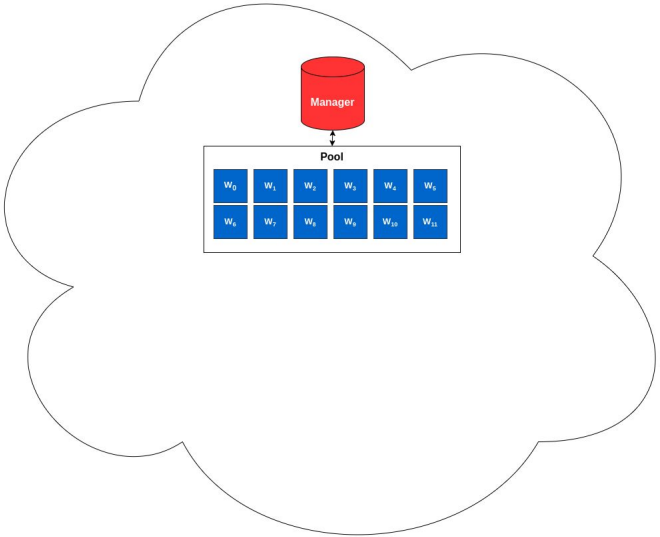
Pool Size: P

Tree Size: N < P

[1]TreeBuilder: https://github.com/duclos-cavalcanti/TreeBuilder

TreeBuilder[1]: Testbench and Design

- **Goal**: VM Selection Method to minimize *“stragglers”*
- **Testbench**:
  1. Launch a cluster of **P** VMs
     - **Workers => Pool**
     - **Manager => Supervisor**

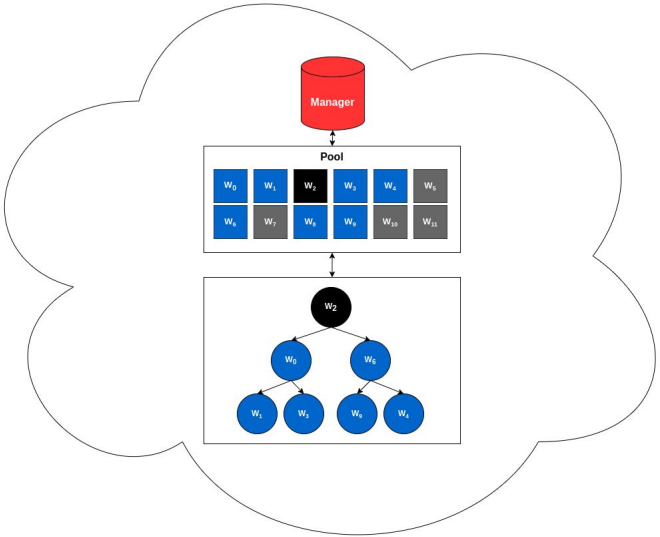


Manager x Workers

[1]TreeBuilder: https://github.com/duclos-cavalcanti/TreeBuilder

TreeBuilder[1]: Testbench and Design

- **Goal**: VM Selection Method to minimize *“stragglers”*
- **Testbench**:
  1. Launch a cluster of **P** VMs
  2. Apply Techniques:

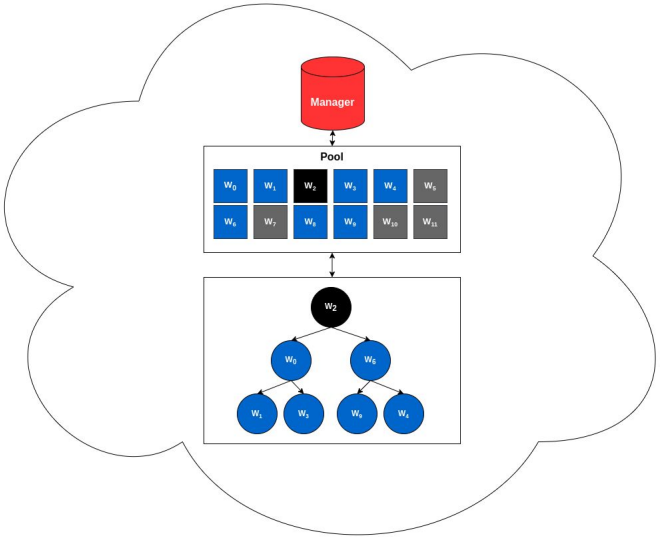


Manager x Workers

[1]TreeBuilder: https://github.com/duclos-cavalcanti/TreeBuilder

TreeBuilder[1]: Testbench and Design

- **Goal**: VM Selection Method to minimize *“stragglers”*
- **Testbench**:
  1. Launch a cluster of **P** VMs
  2. Apply Techniques:
     - Proposed Method

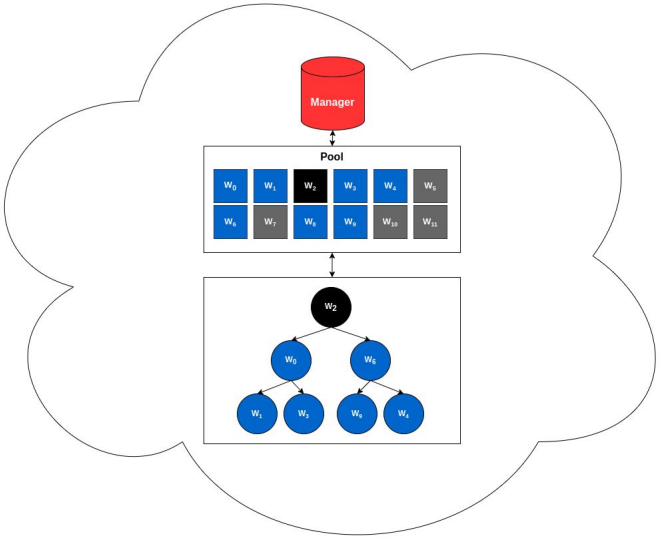


Manager x Workers

[1]TreeBuilder: https://github.com/duclos-cavalcanti/TreeBuilder

TreeBuilder[1]: Testbench and Design

- **Goal**: VM Selection Method to minimize *“stragglers”*
- **Testbench**:
  1. Launch a cluster of **P** VMs
  2. Apply Techniques:
     - Proposed Method
     - LemonDrop

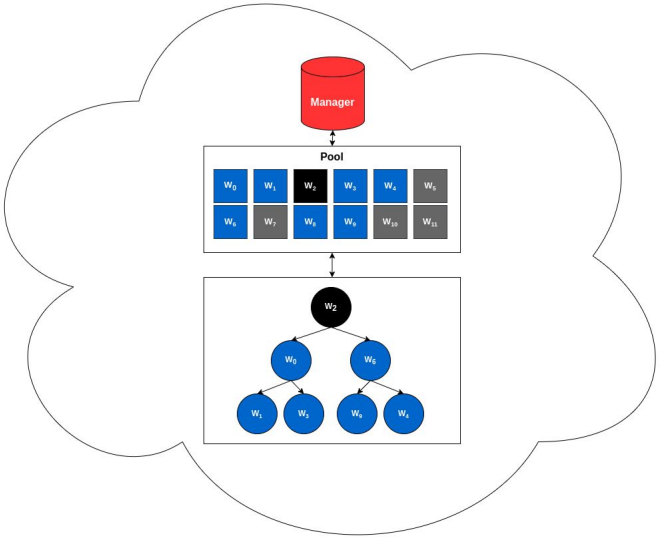


Manager x Workers

[1]TreeBuilder: https://github.com/duclos-cavalcanti/TreeBuilder

TreeBuilder[1]: Testbench and Design

- **Goal**: VM Selection Method to minimize *“stragglers”*
- **Testbench**:
  1. Launch a cluster of **P** VMs
  2. Apply Techniques:
     - Proposed Method
     - LemonDrop
     - Vanilla



Manager x Workers

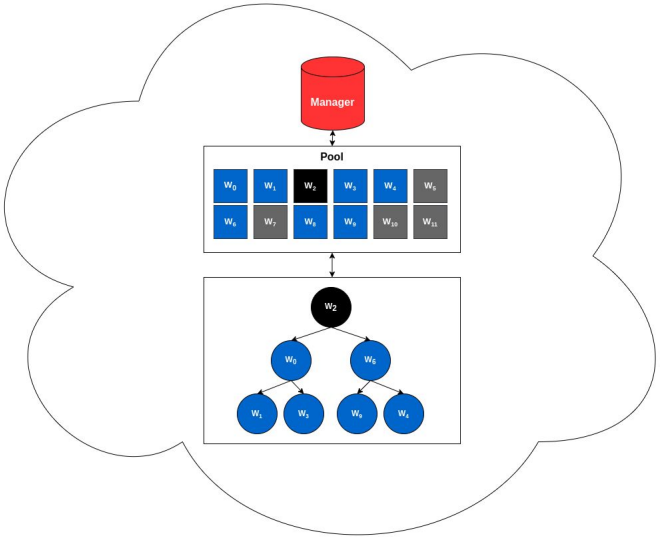
[1]TreeBuilder: https://github.com/duclos-cavalcanti/TreeBuilder

TreeBuilder[1]: Testbench and Design

- **Goal**: VM Selection Method to minimize *“stragglers”*
- **Testbench**:
  1. Launch a cluster of **P** VMs
  2. Apply Techniques:
     - Proposed Method
     - LemonDrop
     - Vanilla
  3. Evaluate trees

Q1: How are nodes examined and selected?

Q2: How are trees evaluated?

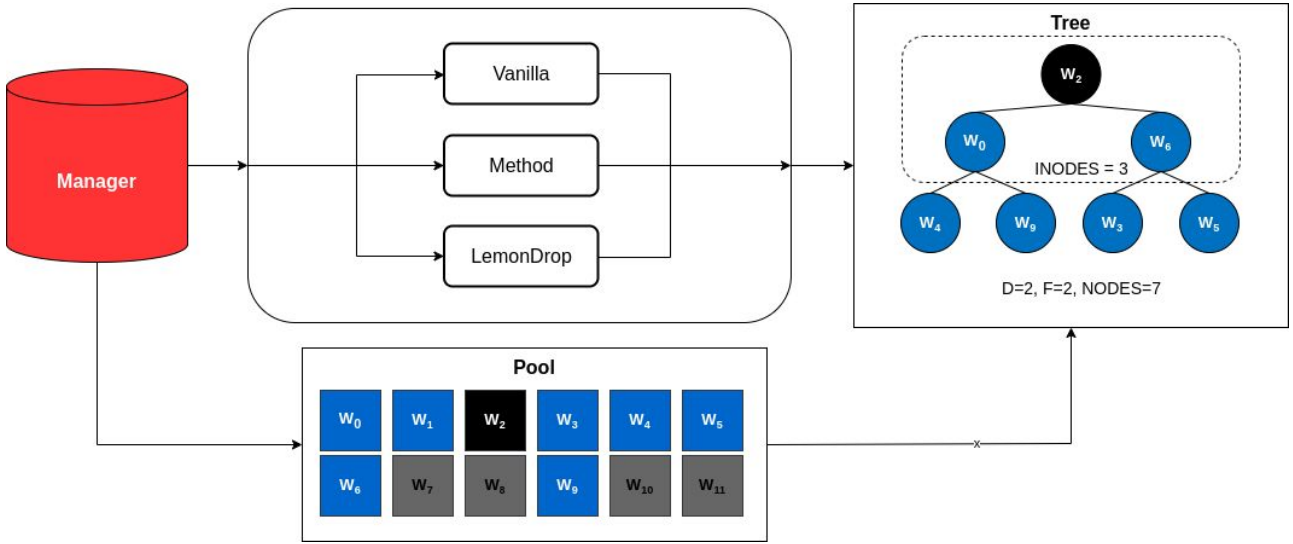


Manager x Workers

[1]TreeBuilder: https://github.com/duclos-cavalcanti/TreeBuilder

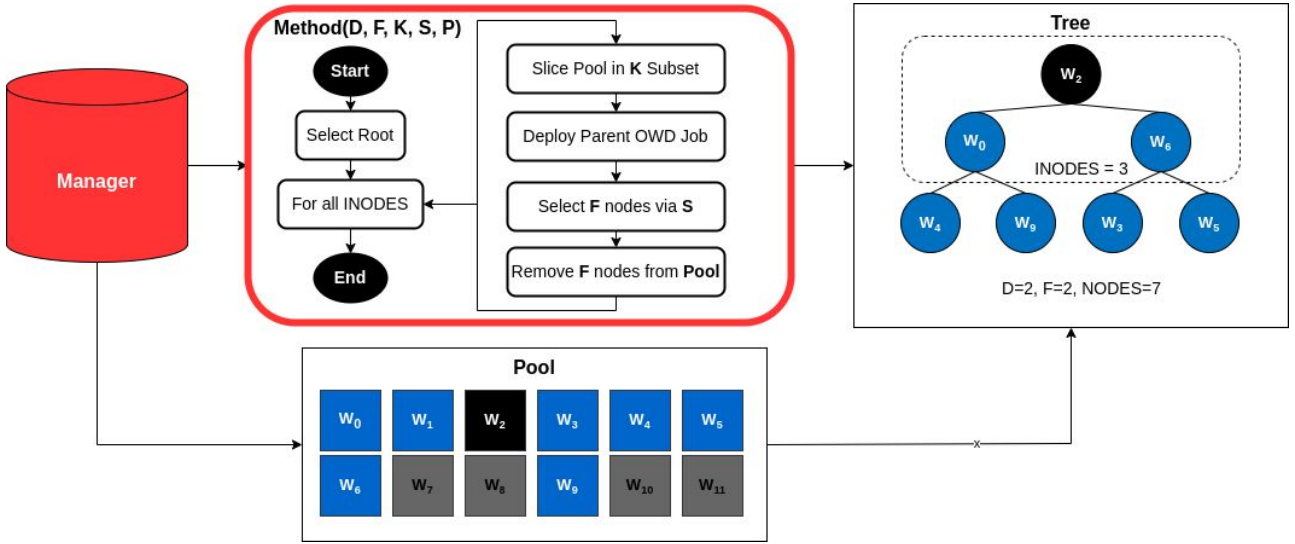
TreeBuilder: Testbench and Design

1. Launches cluster of **N** VMs (Manager x Workers)
2. Apply: [Proposed Method, LemonDrop and Vanilla]
3. Evaluate Trees



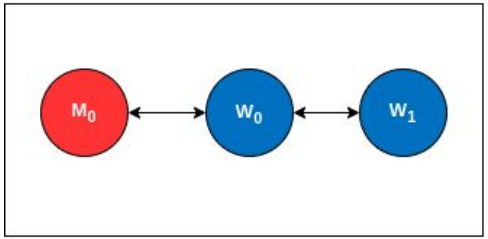
TreeBuilder: Testbench and Design

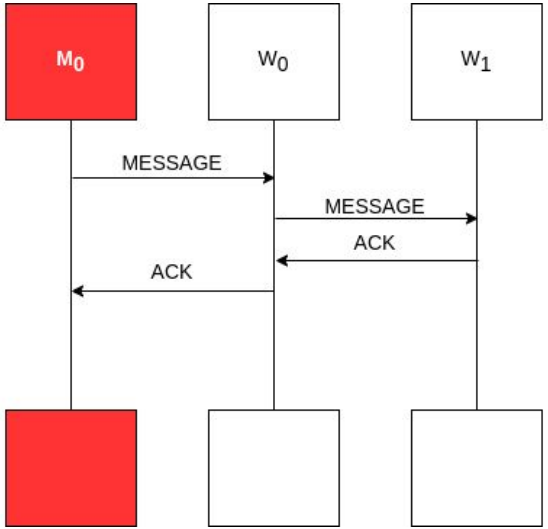
1. Launches cluster of **N** VMs (Manager x Workers)
2. Apply: [**Proposed Method**, LemonDrop and Vanilla]
3. Evaluate Trees



Implementation: Messages

- Request and Reply
- Defined and Flexible Data Structures





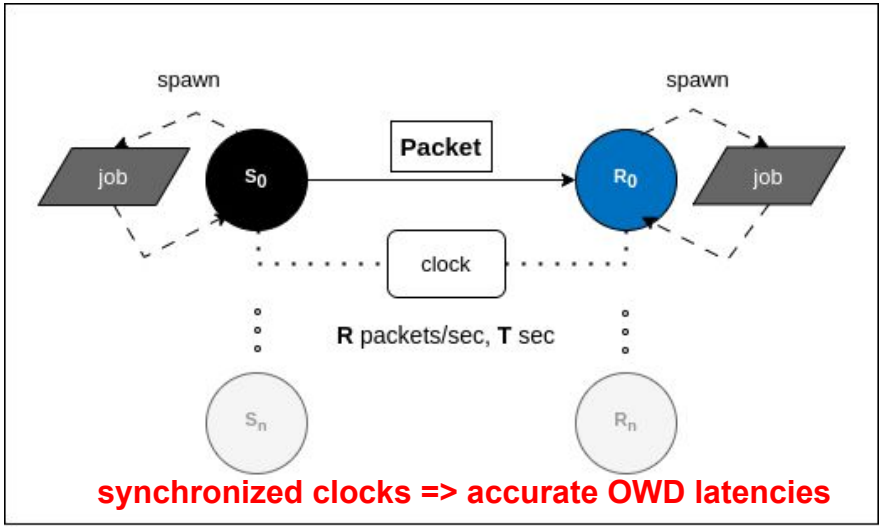
Message
+ int: id, ts;
+ string: src, dst, ref;
+ Type: type;
+ Metadata: mdata;

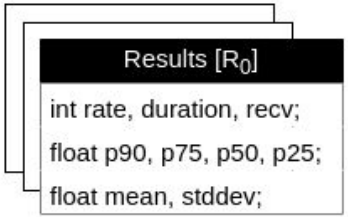
Type [enum]
ACK
CONNECT
COMMAND
REPORT
ERR
LOG

Metadata [Union]
Command c;
Job j;
Report r;

Implementation: Node(s) Examinations

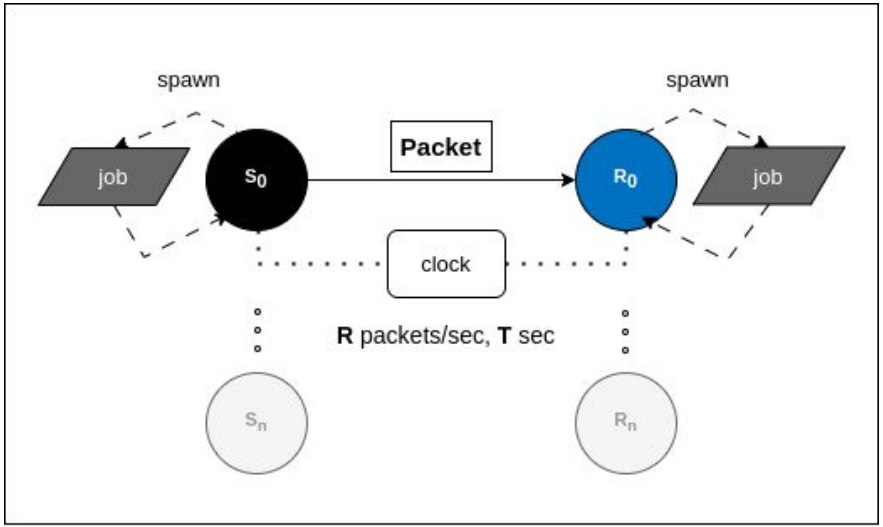
- Collective Tasks (Jobs)





Implementation: Node(s) Examinations

- Collective Tasks (Jobs)
- Generate Results



Results [R_0]

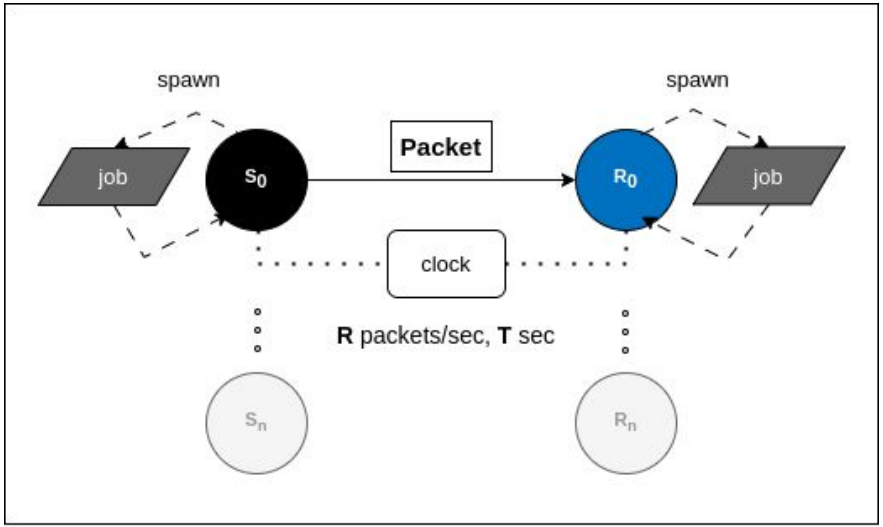
int rate, duration, recv;

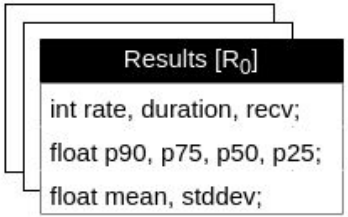
float p90, p75, p50, p25;

float mean, stddev;

Implementation: Node(s) Examinations

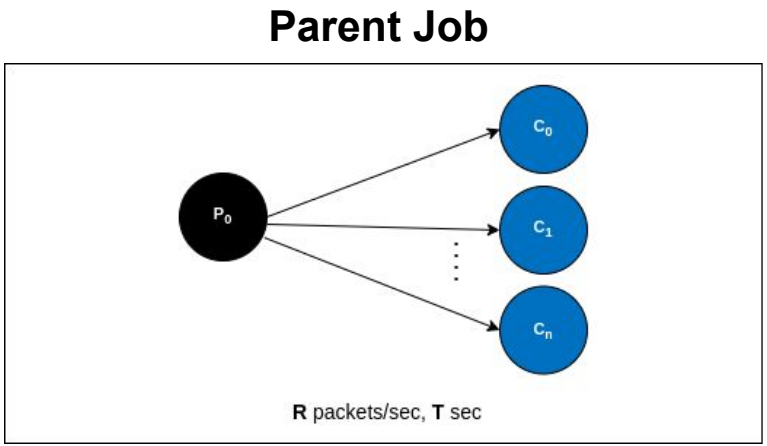
- Collective Tasks (Jobs)
- Generate Results
- Types:

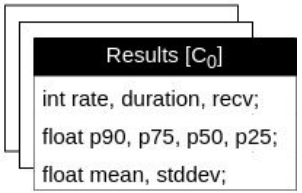




Implementation: Node(s) Examinations

- Collective Tasks (Jobs)
- Generate Results
- Types:
  - Parent

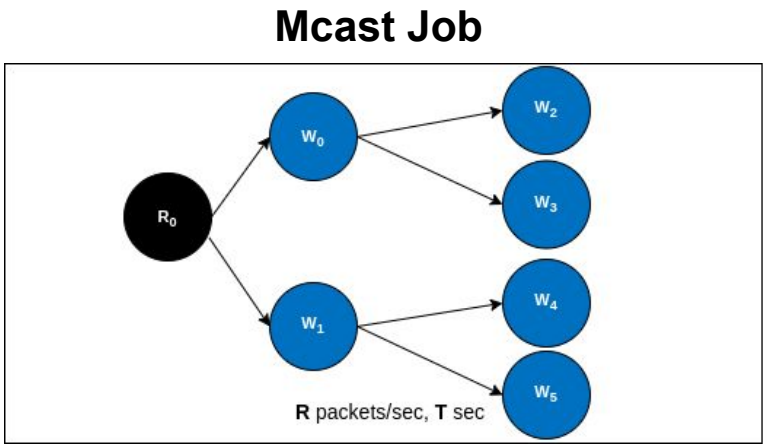


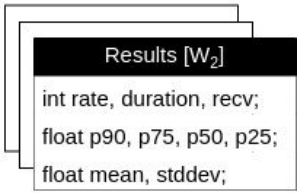


Examines potential children nodes to a given parent!

Implementation: Node(s) Examinations

- Collective Tasks (Jobs)
- Generate Results
- Types:
  - Parent
  - Mcast

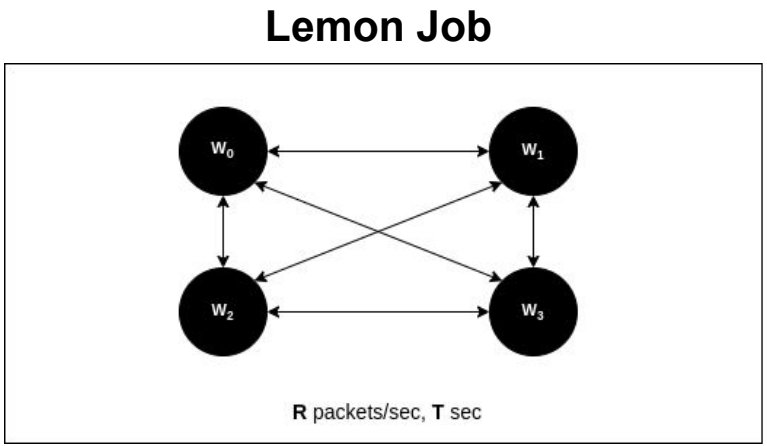


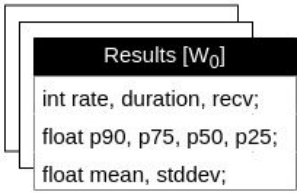


Evaluates performance of a built tree!

Implementation: Node(s) Examinations

- Collective Tasks (Jobs)
- Generate Results
- Types:
  - Parent
  - Mcast
  - Lemon

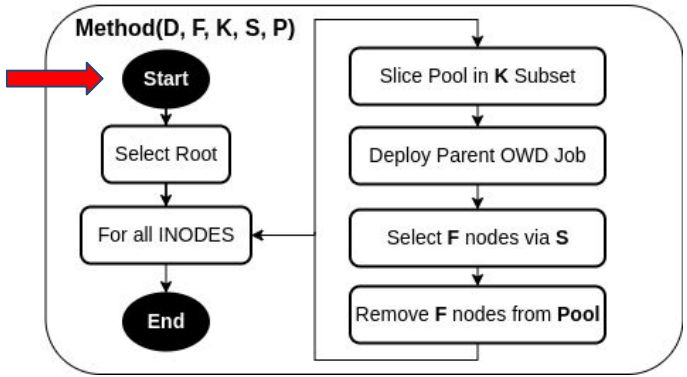


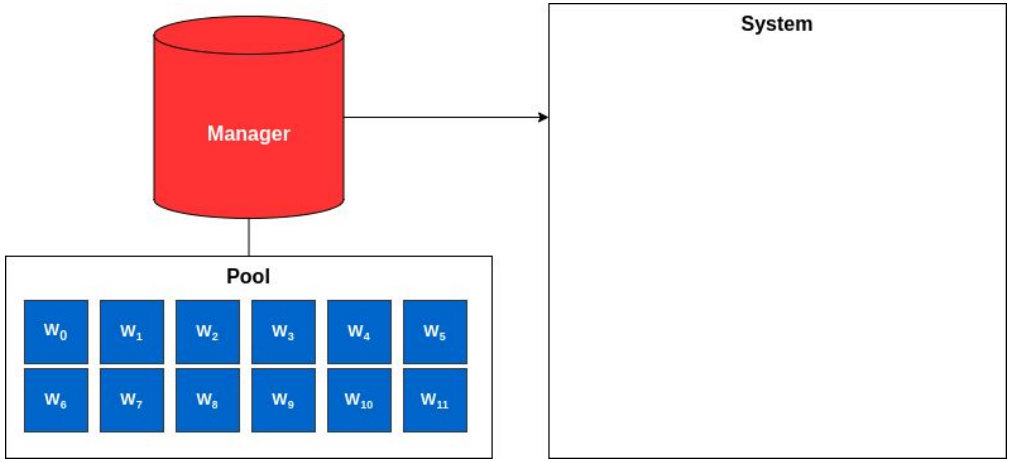


Collects pair-wise median OWDs for LemonDrop's Heuristic!

Implementation: Proposed Selection Method

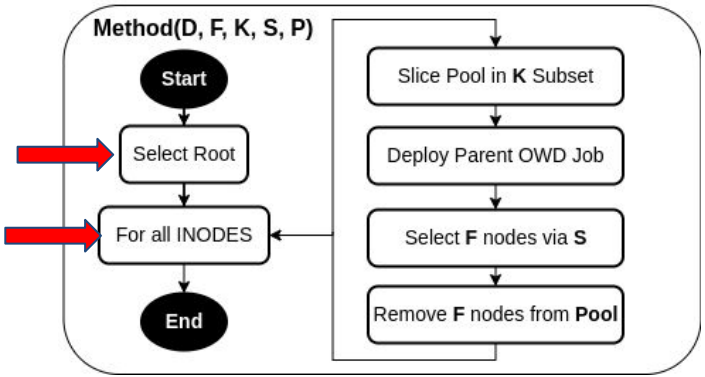
- Define depth and fan-out: **D , F**
- Bounding parameter: **K**
- Strategy: **S**
- Pool: **P**

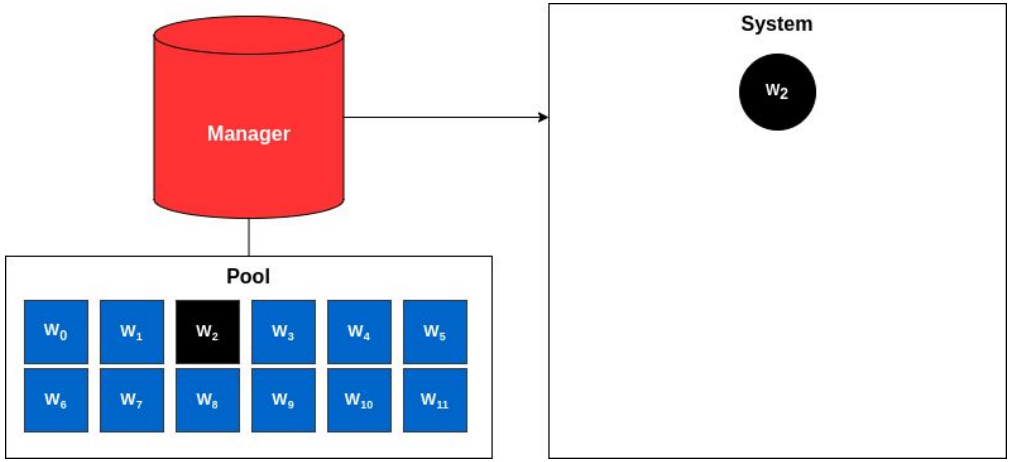




Implementation: Proposed Selection Method

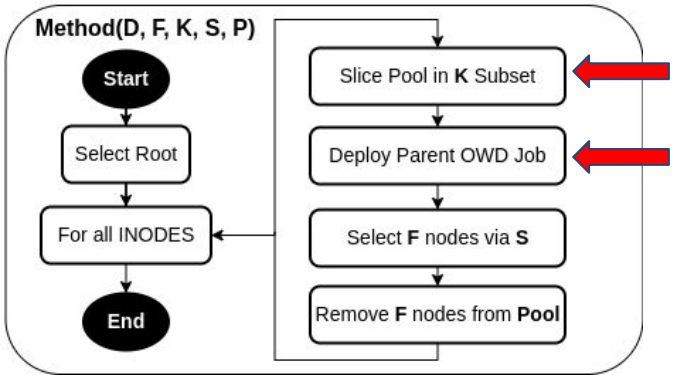
- Define depth and fan-out: **D , F**
- Bounding parameter: **K**
- Strategy: **S**
- Pool: **P**

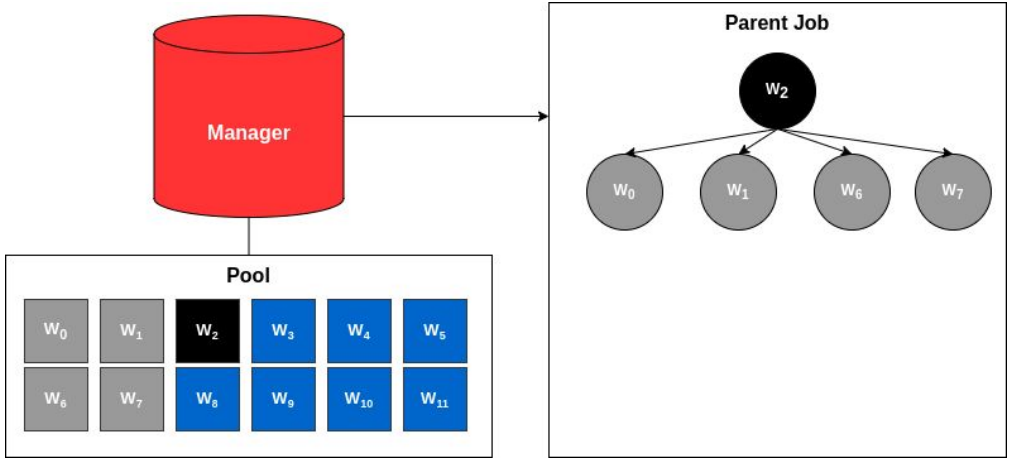




Implementation: Proposed Selection Method

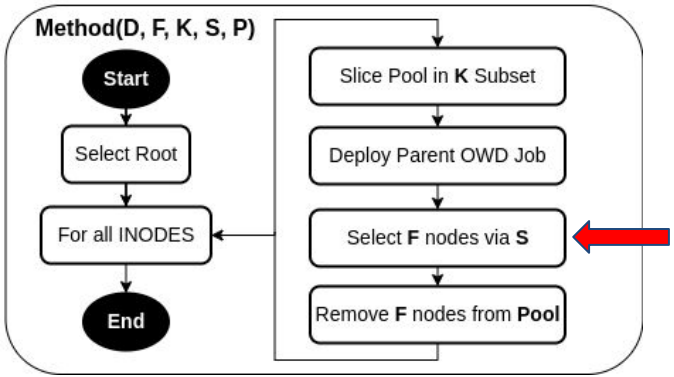
- Define depth and fan-out: **D , F**
- Bounding parameter: **K**
- Strategy: **S**
- Pool: **P**

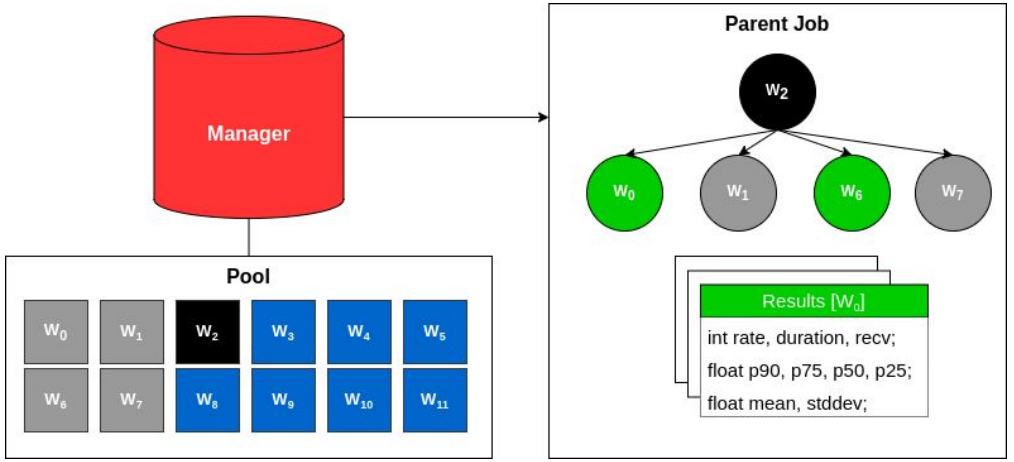




Implementation: Proposed Selection Method

- Define depth and fan-out: **D , F**
- Bounding parameter: **K**
- Strategy: **S**
- Pool: **P**





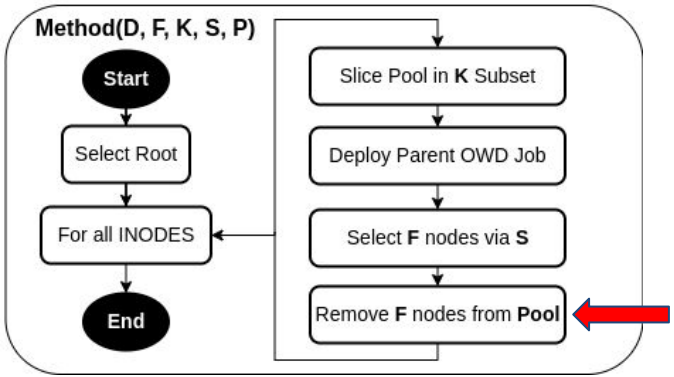
Strategies [**S**]:

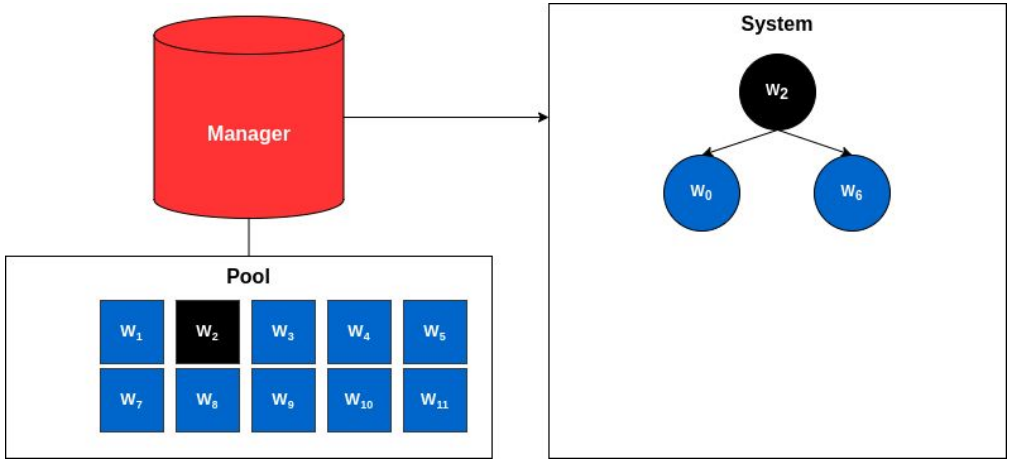
- **P90:** (1 x p90)
- **P50:** (1 x p50)
- **Weighted:** (0.7 x stddev) + (0.3 x p90)

lower => better

Implementation: Proposed Selection Method

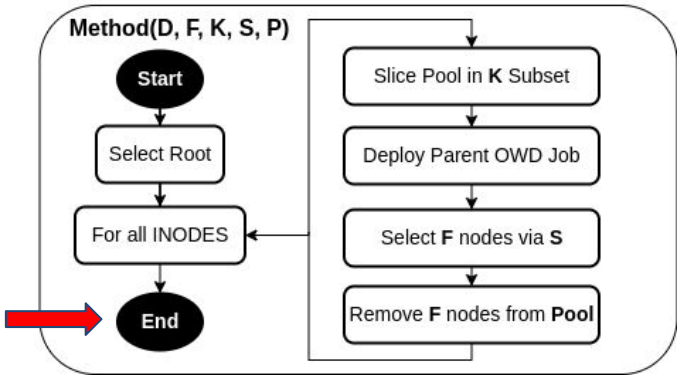
- Define depth and fan-out: **D , F**
- Bounding parameter: **K**
- Strategy: **S**
- Pool: **P**

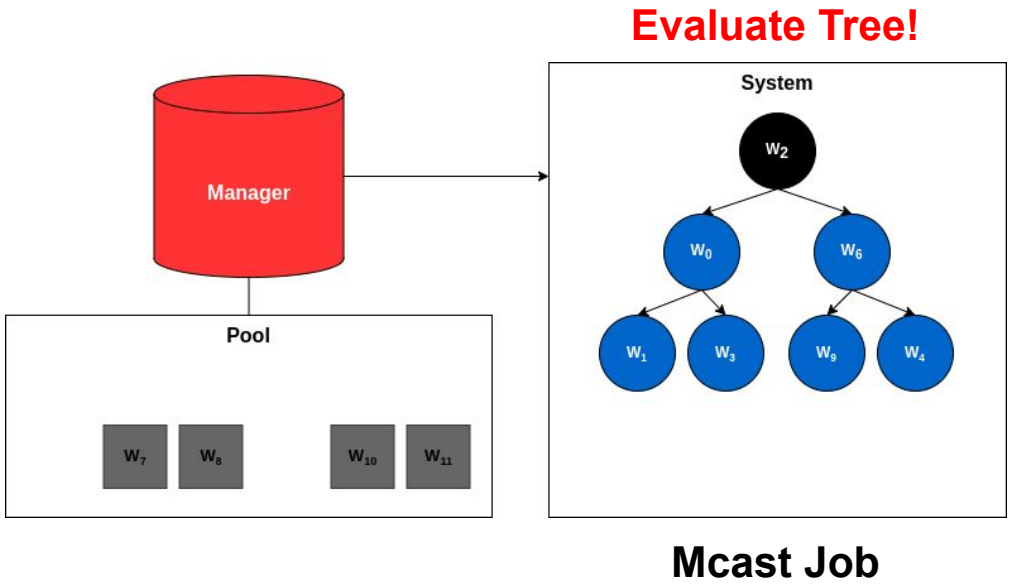




Implementation: Proposed Selection Method

- Define depth and fan-out: **D , F**
- Bounding parameter: **K**
- Strategy: **S**
- Pool: **P**





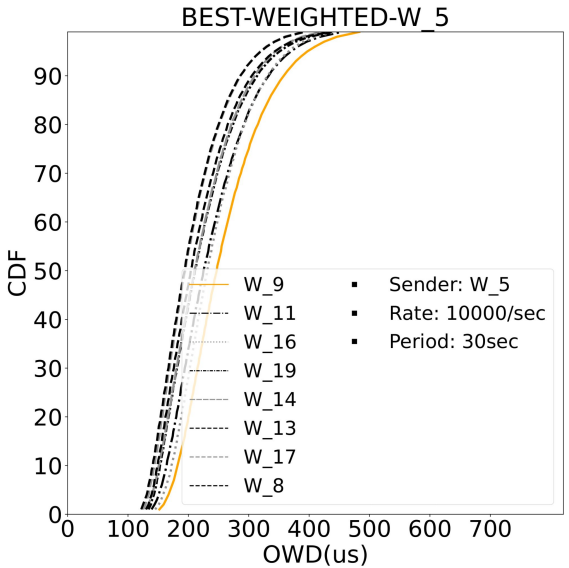
Experimental Setup

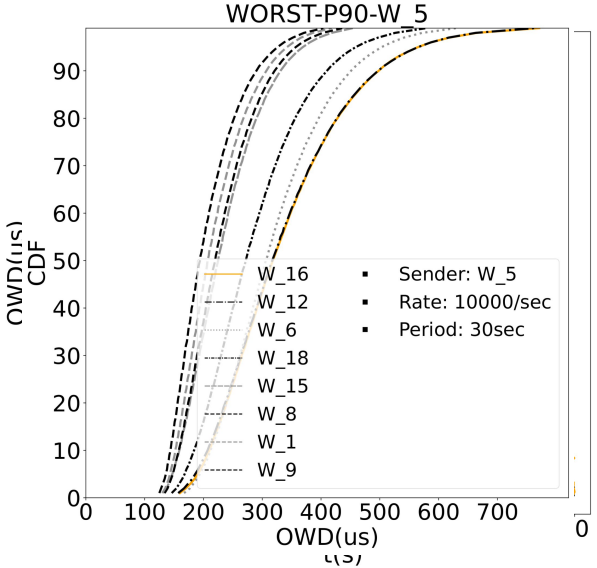
- Build Trees:
  - **Best**: P90 x P50 x WEIGHTED
  - **Worst**: P90
  - **Random**
  - **Lemon:** ϵ = 1 x 10e-4 , n = 1000
- GCP Cloud Instances: “c2d-highcpu-8”
  - 8 virtual AMD Milan CPU’s
  - 16GB of Memory
  - 16Gbps of Network Bandwidth

D (Depth)	3
F (Fanout)	2
K (Pool Subset)	2*F = 4
P (Pool Size)	25
Duration Parent Job	10 sec
Duration Mcast Job	30 sec
Rate	10K
Number of Runs	2
Number of Roots	2

Results: Straggler Impact – BEST x WORST

- **WORST** tree displays sparser CDF profiles and right-shifted vs **BEST-WEIGHTED**
- **WORST** shows higher variancy
- **BEST-WEIGHTED** 28.11% lower OWD latency at 90th %



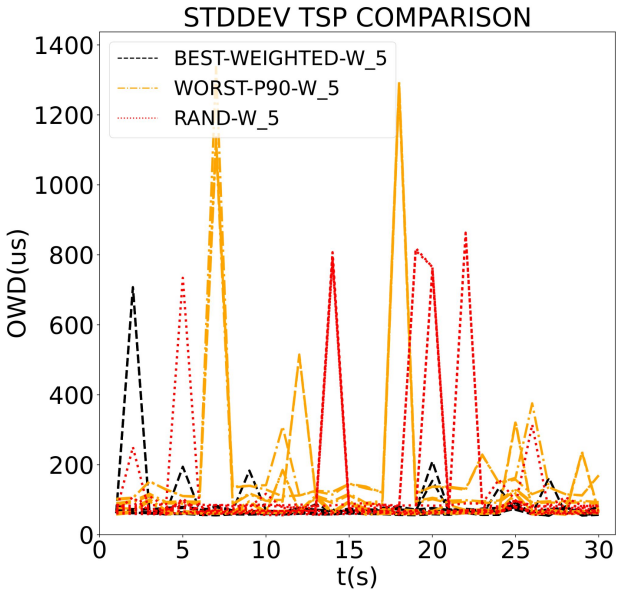


	BEST-WEIGHTED	WORST
90th %	**358 µs**	**498 µs**
median	247 µs	318 µs
mean	261.4 µs	353.66 µs
stddev	78.95 µs	352.29 µs
ID	W_9	W_16

➢ **Best-WEIGHTED is uniformly better & more stable!**

Results: Straggler Impact – BEST x WORST x RAND

- **BEST** 6.53% lower OWD 90th % vs **RAND**
- **RAND** in between **BEST** and **WORST**
- **RAND** displays considerable variance vs **BEST**

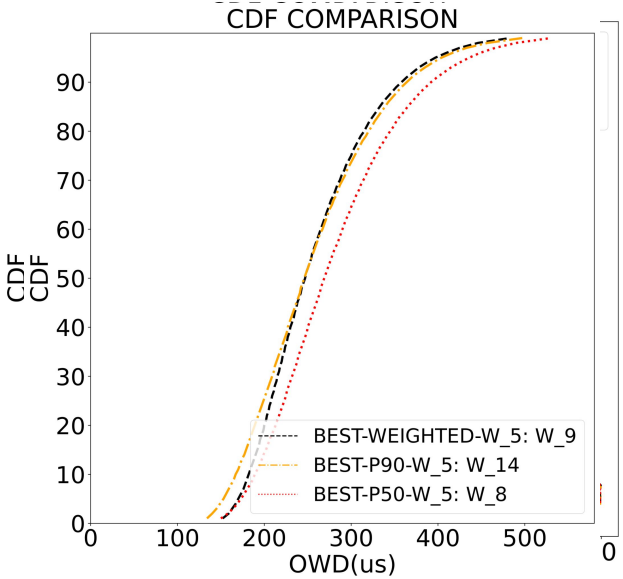


	BEST-WEIGHTED	RAND	WORST
90th %	**358 µs**	**383 µs**	**498 µs**
median	247 µs	256 µs	318 µs
mean	261.4 µs	281.1 µs	353.66 µs
stddev	78.95 µs	308.3 µs	352.29 µs
ID	W_9	W_15	W_16

➢ **There is value in optimizing against stragglers!**

Results: Strategy Comparison – WEIGHTED x P90 x P50

- **WEIGHTED** 6.53% lower 90th % than **P50**
- **P90** shows some of the best latency profiles
- **P50** shows less variance

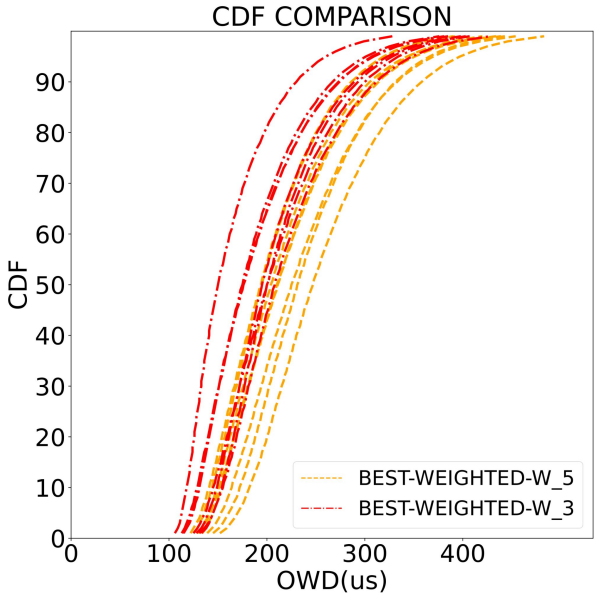


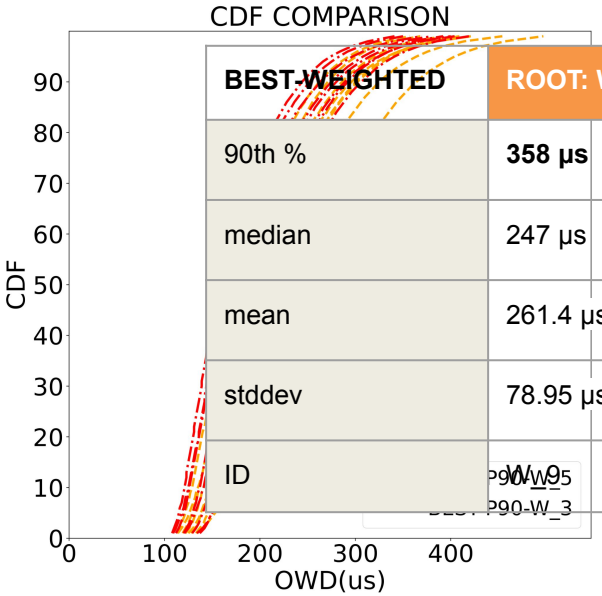
	BEST-WEIGHTED	BEST-P90	BEST-P50
90th %	**358 µs**	**364 µs**	**392 µs**
median	247 µs	246 µs	269 µs
mean	261.4 µs	262.11 µs	282.57 µs
stddev	78.95 µs	159.2 µs	83.57 µs
ID	W_9	W_14	W_18

➢ **P50 prioritizes stability, P90 performance, WEIGHTED seeks balance**

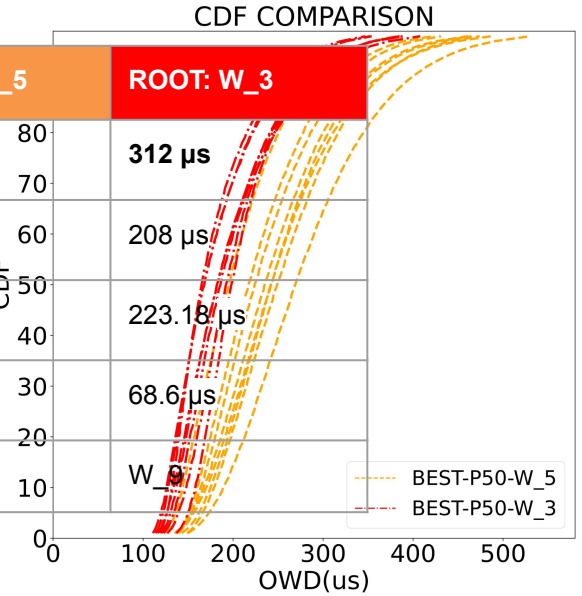
Results: Root Selection

- All strategies performed worse with **W_5** as root
- **W_3**-based root shows 12.85% lower latency in 90th%





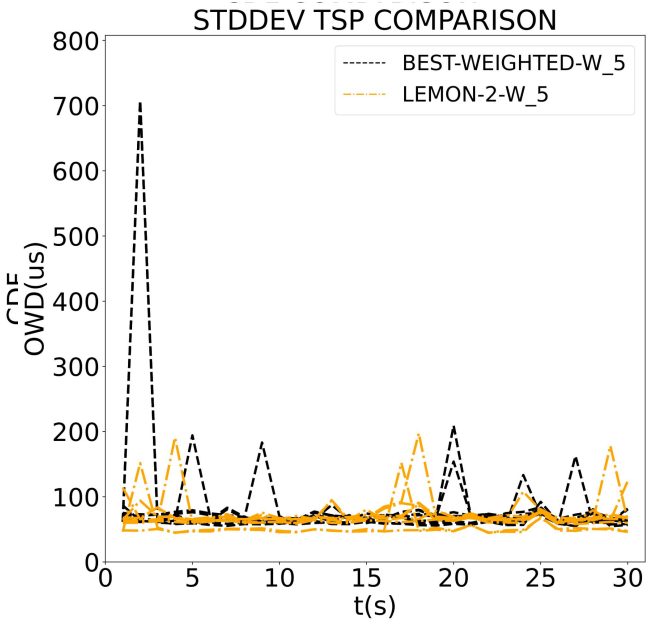
BEST-WEIGHTED	ROOT: W_5	ROOT: W_3
90th %	**358 µs**	**312 µs**
median	247 µs	208 µs
mean	261.4 µs	223.18 µs
stddev	78.95 µs	68.6 µs
ID	W_9	W_9



➢ **Root selection impact is significant!**

Results: LemonDrop Comparison

- **LEMON-2** 9.22% lower 90th % than **BEST-WEIGHTED**
- **LEMON-2** left-shifted CDF profile vs **BEST-WEIGHTED**
- **LEMON-2** slightly less variant

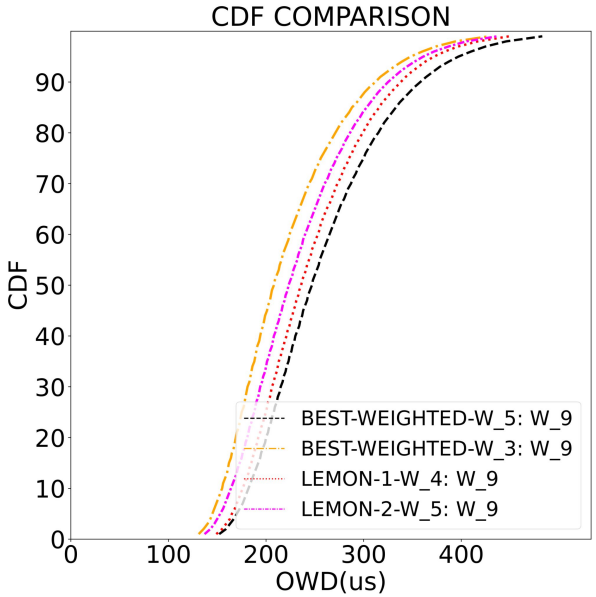


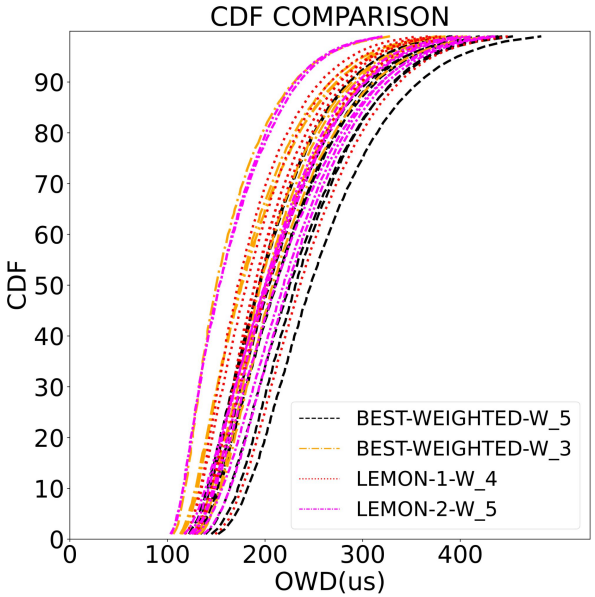
	BEST-WEIGHTED	LEMON-2
90th %	**358 µs**	**325 µs**
median	247 µs	223 µs
mean	261.4 µs	236.7 µs
stddev	78.95 µs	76.39 µs
ID	W_9	W_9

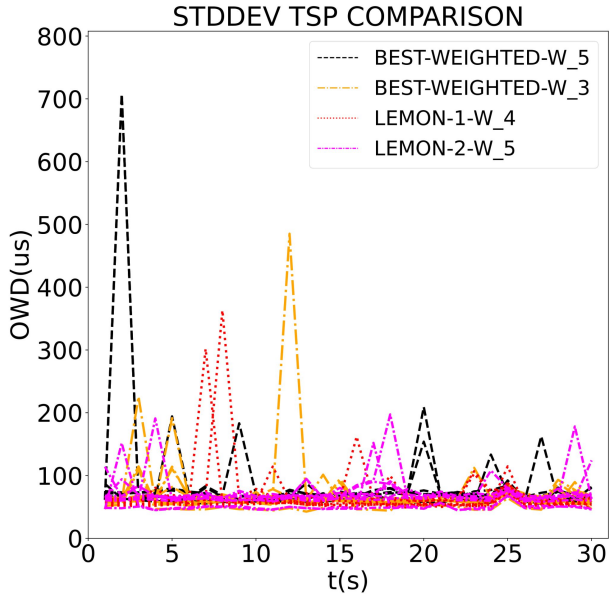
LEMON-2 converged in 0.0016sec

Results: LemonDrop Comparison

- **BEST-WEIGHTED-W_3** displays better performing worst receiver
- **LEMON-1** and **LEMON-2** show overall healthier receiver profiles and less variance







LEMON-1 and LEMON-2 converged respectively in 0.0346 and 0.0016 seconds

➢ **Comparable performance, Lemon offers more stability**

Conclusions

Takeaways

1. Flexible Selection Framework:
   a. Detects stragglers
   b. Improves multicast latency (µs)
   c. Trade-offs between variance x perf.
   d. Straight-Forward, Lightweight
2. Comparable to LemonDrop
   a. pre-defined hyperparameters (ε , n)

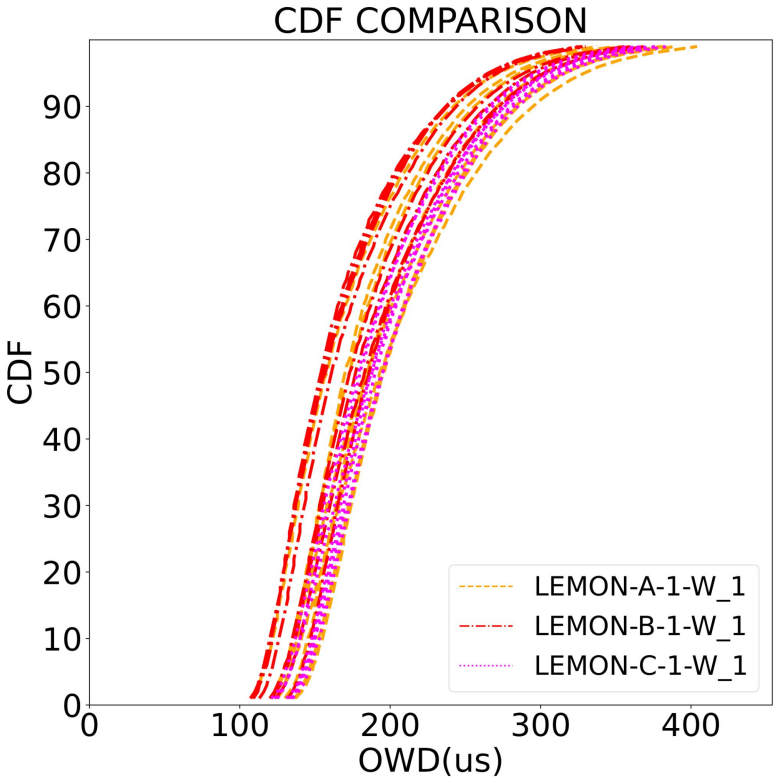
Future Work

1. Incremental Updates
   a. Latency profile per subtree
   b. auto-scale triggers
2. Root selection mechanism
3. Refining heuristic:
   a. high-performance
   b. stability

Questions!

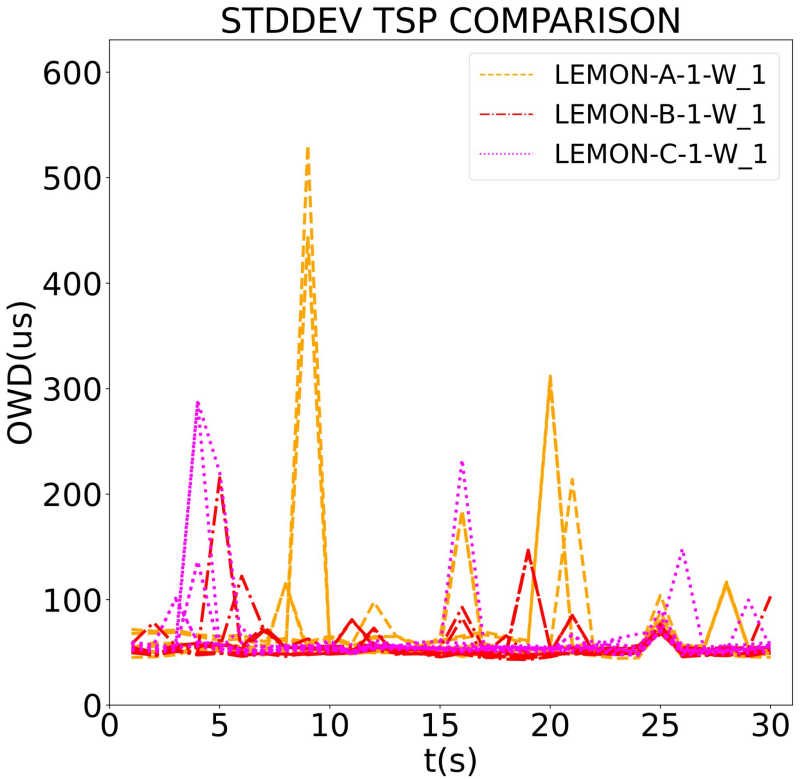
LemonDrop Parametrization

	LEMON-A	LEMON-B	LEMON-C
90th %	**295 µs**	**273 µs**	**285 µs**
stddev	83.35 µs	64.19 µs	57.64 µs
epsilon	1 x 10e-4	2.2 x 10e-5	1.7 x 10e-5
n	1000	10 000	100 000
conv.	TRUE	FALSE	FALSE
time	0.0381 sec	11.26 sec	111.94 sec

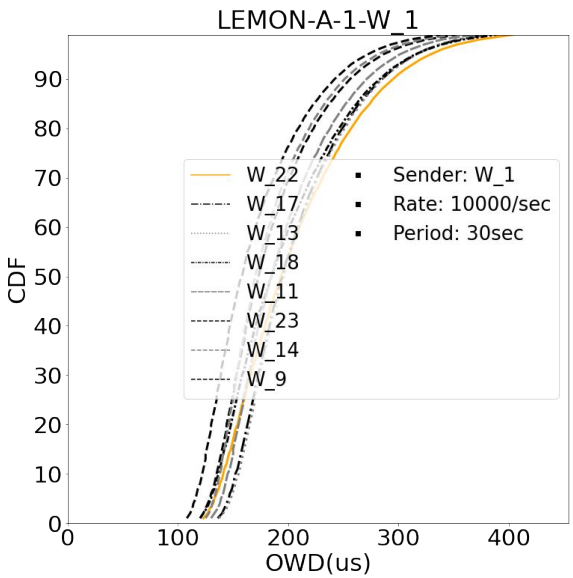


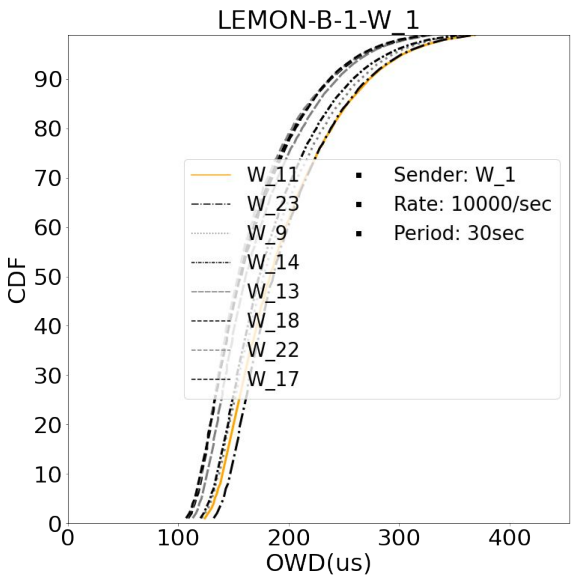
LemonDrop Parametrization

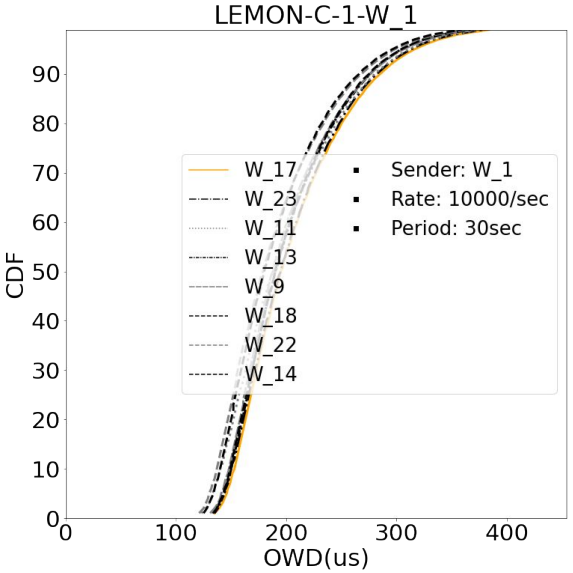
	LEMON-A	LEMON-B	LEMON-C
90th %	**295 µs**	**273 µs**	**285 µs**
stddev	83.35 µs	64.19 µs	57.64 µs
epsilon	1 x 10e-4	2.2 x 10e-5	1.7 x 10e-5
n	1000	10 000	100 000
conv.	TRUE	FALSE	FALSE
time	0.0381 sec	11.26 sec	111.94 sec



LemonDrop Parametrization





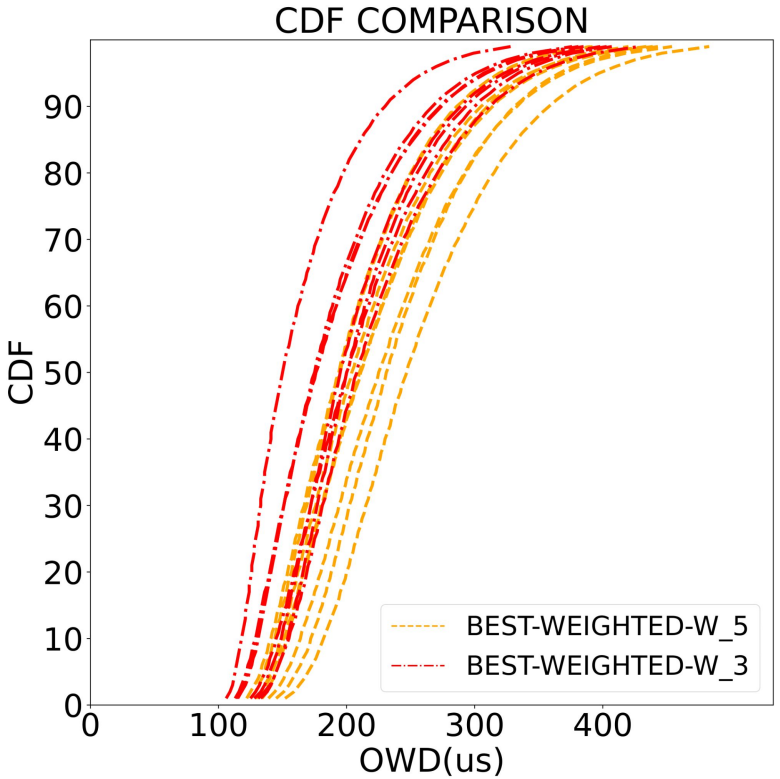


Root Selection impacts final system significantly – WEIGHTED

1. Contrast all receivers

BEST-WEIGHTED	ROOT: W_5	ROOT: W_3
90th %	**358 µs**	**312 µs**
median	247 µs	208 µs
mean	261.4 µs	223.18 µs
stddev	78.95 µs	68.6 µs
ID	W_9	W_9

Weighted = (0.7 x stddev) + (0.3 x p90)

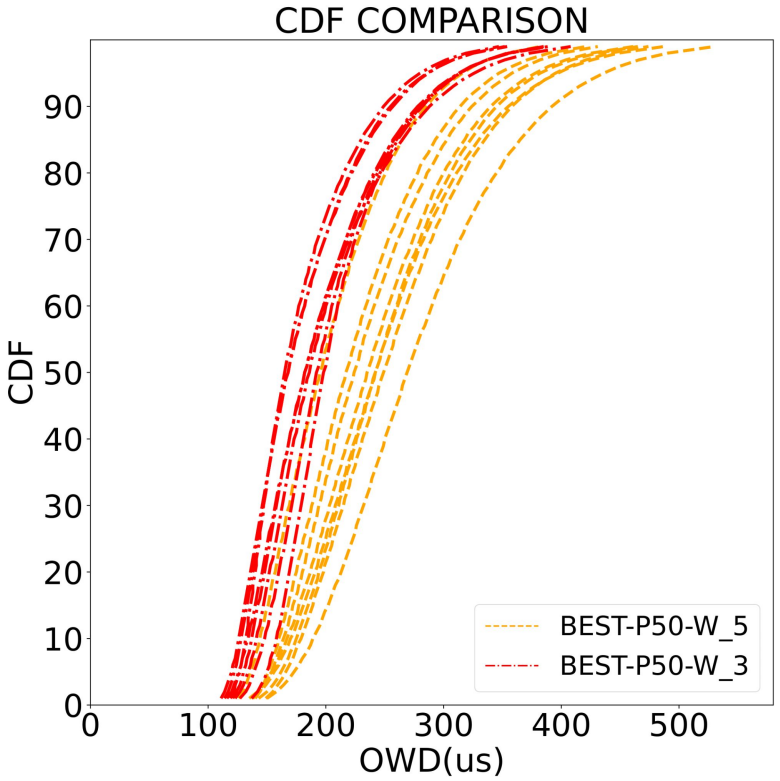


Root Selection impacts final system significantly – P50

1. Contrast all receivers

BEST-P50	ROOT: W_5	ROOT: W_3
90th %	**392 µs**	**288 µs**
median	269 µs	186 µs
mean	282.57 µs	205.25 µs
stddev	83.57 µs	156.08 µs
ID	W_18	W_9

P50: (1.0 x p50)

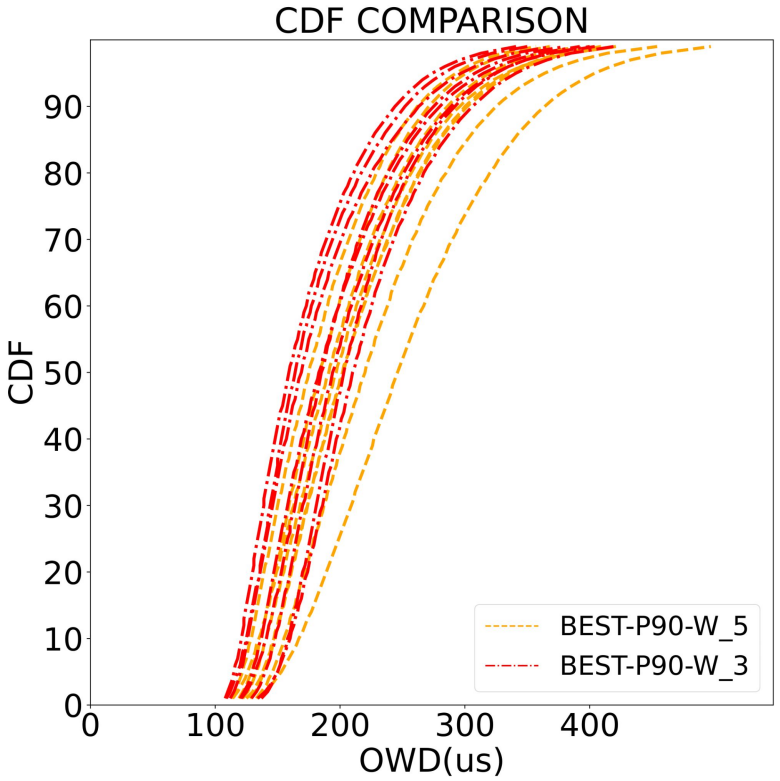


Root Selection impacts final system significantly – P90

1. Contrast all receivers

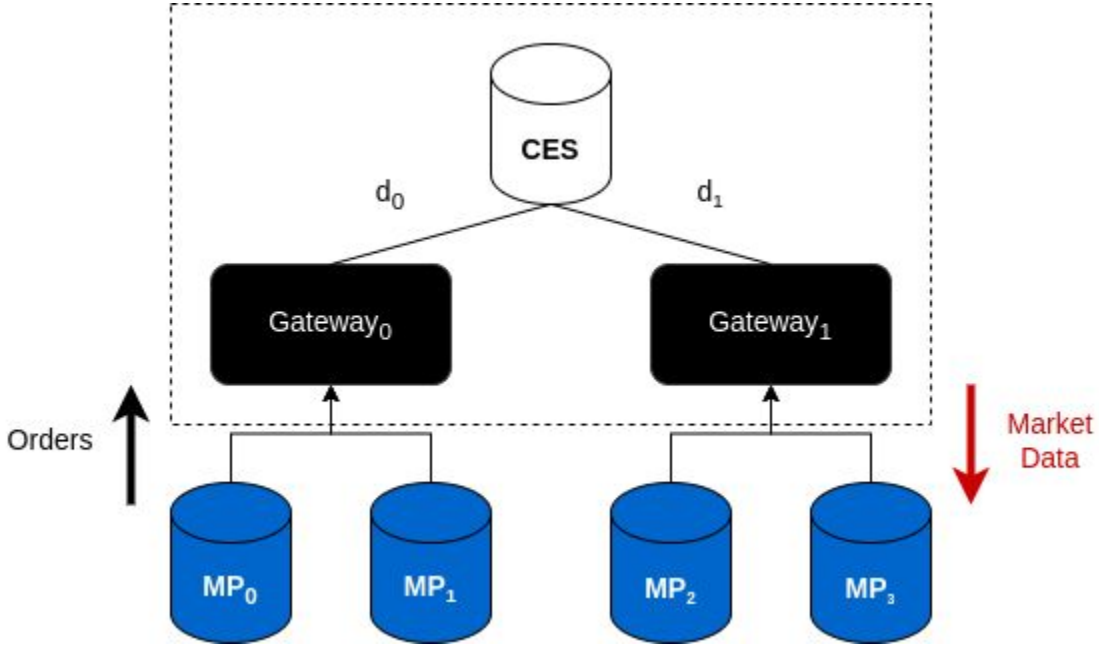
BEST-P90	ROOT: W_5	ROOT: W_3
90th %	**364 µs**	**306 µs**
median	246 µs	210 µs
mean	262.11 µs	226.45 µs
stddev	159.2 µs	153.93 µs
ID	W_14	W_6

P90: (1.0 x p90)



Background: Financial Exchanges

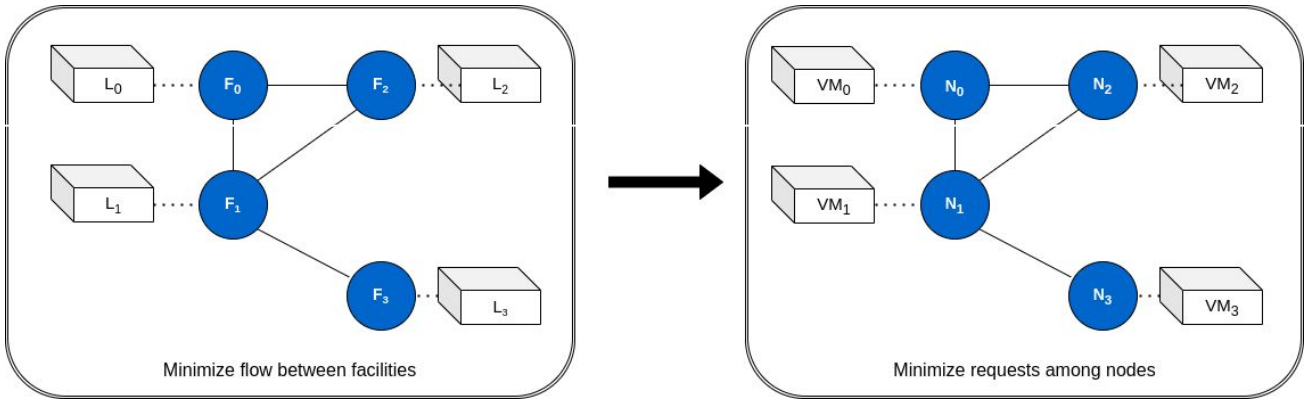
- Fairness or “Fair Access” to Market Participants (MPs)
- Migration to the Cloud: **i) ~~Mechanisms~~ for fair delivery ii) Latency Variance**



Background: Financial Exchanges to the Cloud

State-of-the-Art: LemonDrop[1]

- Core of method:
  - Clock Synchronization
  - (QAP) Optimization Method: Facilities=Nodes, VMs=Locations
- Λ: Load Matrix, Δ: OWD Matrix, P: Permutation Matrix
- $D = \sum^{L}_{i,j=1} \lambda_{ij} d_{ij}$, $D(P) = \text{trace}(\Lambda P \Delta^{T} P^{T})$



[1]LemonDrop: https://searchworks.stanford.edu/view/14423035

State-of-the-Art: LemonDrop[1]

- Λ: Load Matrix, Δ: OWD Matrix, P: Permutation Matrix
- Hadamard Product: $D = \sum_{i,j=1}^{L} \lambda_{ij} d_{ij}$
- Goal: Minimize $D(P) = \text{trace}(\Lambda P \Delta^T P^T)$

Δ: (NxN)

$$\begin{bmatrix} d_{00} & d_{01} & d_{02} \\ d_{10} & d_{11} & d_{12} \\ d_{20} & d_{21} & d_{22} \end{bmatrix}$$

Λ: (LxL)

$$\begin{bmatrix} \lambda_{00} & \lambda_{01} & \lambda_{02} \\ \lambda_{10} & \lambda_{11} & \lambda_{12} \end{bmatrix}$$

P: (NxN)

$$\rightarrow \quad \text{trace}\left(\Lambda \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \Delta^T P^T \right)$$

[1]LemonDrop: https://searchworks.stanford.edu/view/14423035

State-of-the-Art: LemonDrop[1]

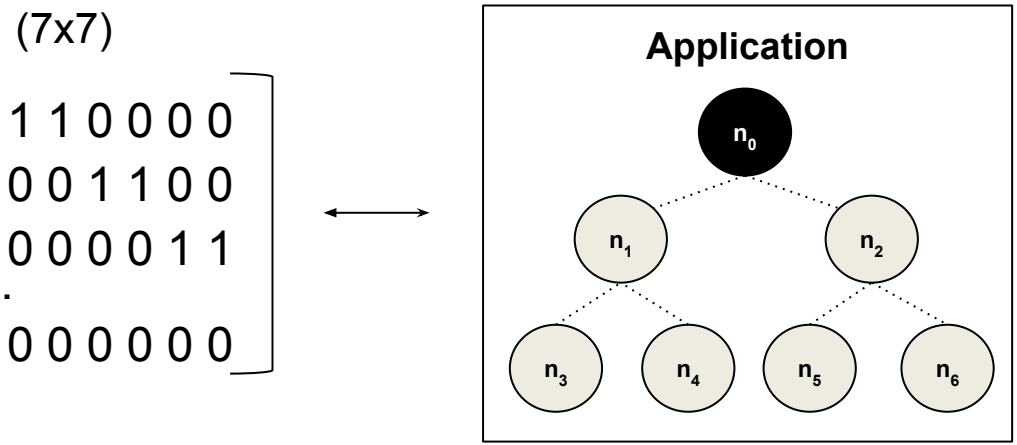
- Λ: Load Matrix, Δ: OWD Matrix, P: Permutation Matrix
- Hadamard Product: $D = \sum^{L}_{i,j=1} \lambda_{ij} d_{ij}$
- Goal: Minimize $D(P) = \text{trace}(\Lambda P \Delta^{T} P^{T})$

Δ: (10x10)

$$\begin{bmatrix} d_{00} & d_{01} \dots d_{09} \\ d_{10} & d_{11} \dots d_{19} \\ \dots & \\ d_{90} & d_{91} \dots d_{99} \end{bmatrix}$$

Λ: (7x7)

$$\begin{bmatrix} 0\,1\,1\,0\,0\,0\,0 \\ 0\,0\,0\,1\,1\,0\,0 \\ 0\,0\,0\,0\,0\,1\,1 \\ \dots \\ 0\,0\,0\,0\,0\,0\,0 \end{bmatrix}$$



[1]LemonDrop: https://searchworks.stanford.edu/view/14423035

State-of-the-Art: LemonDrop[1]

- Λ: Load Matrix, Δ: OWD Matrix, P: Permutation Matrix
- $D = \sum^{L}_{i,j=1} \lambda_{ij} d_{ij}$, $D(P) = \text{trace}(\Lambda P \Delta^T P^T)$

Algorithm 1 FAQ Algorithm [136]

```
function FAQ(Λ, Δ, ϵ, n)
    P^(0) = 11^T
    i = 0                                                  ▷ Iteration number
    s = 0                                                  ▷ Stopping condition
    while s = 0 do
        ∇f(P^(i)) = −ΛP^(i)Δ^T − Λ^T P^(i)Δ                ▷ Gradient wrt current solution
        Q^(i) = argmin_{P∈D} trace(∇f(P^(i))^T P)          ▷ Direction which minimizes 1st order approx of f(P)
        α^i = min_{α∈[0,1]} f(P^(i) + αQ^(i))              ▷ Best step size in chosen direction
        P^(i+1) = P^(i) + α^(i) Q^(i)                      ▷ Update our current solution
        if ||P^(i) − P^(i−1)||_F < ϵ  then s = 1           ▷ Stop opt
        if i > n  then s = 1                               ▷ Stop opt
        i = i + 1
    P^(final) = argmin_{P∈P} P^(i−1) P^T
    return P^(final)
```

[1]LemonDrop: https://searchworks.stanford.edu/view/14423035

Results

EVALUATION COMPARISON - BEST x LEMONS

N=15, D=3, F=2, K=4, POOL=24, Rate=10000 p/sec, Duration=30 sec

BEST-heuristic-W_5

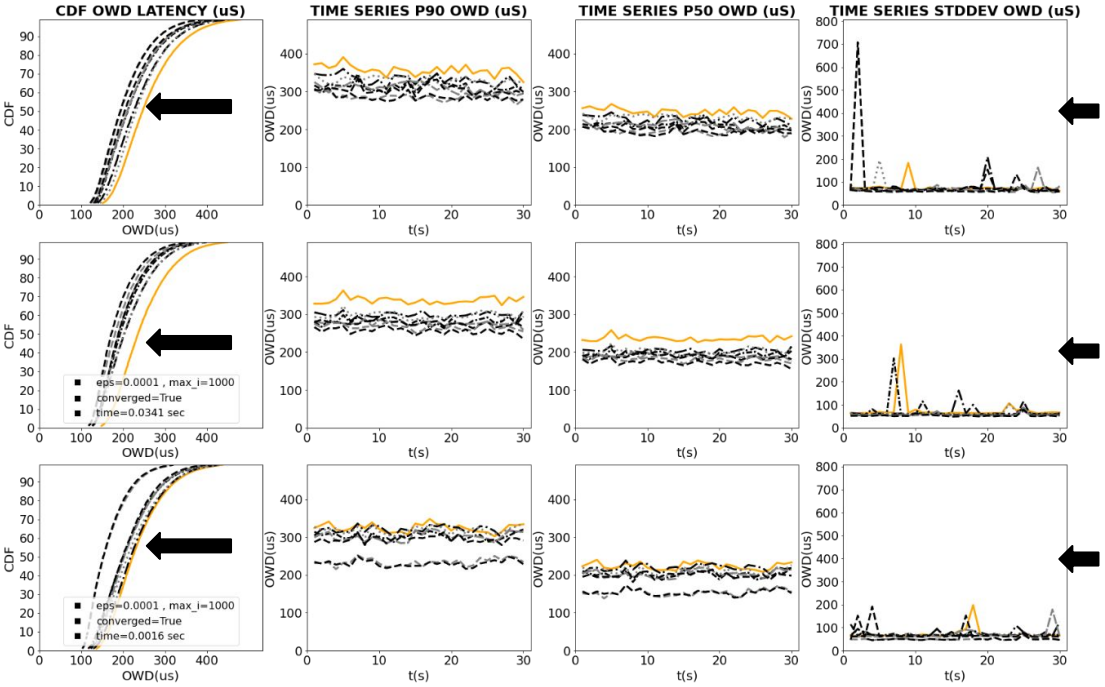
	99(%)	90(%)	50(%)	MEAN	STD	RX(%)
W_9	483.0	358.0	247.0	261.0	78.95	100.0
W_11	454.0	333.0	225.0	239.5	78.93	100.0
W_16	443.0	331.0	231.0	244.2	76.52	100.0
W_19	439.0	314.0	211.0	224.9	68.74	100.0
W_14	420.0	310.0	210.0	222.8	72.15	100.0
W_13	434.0	305.0	204.0	218.5	72.16	100.0
W_17	390.0	286.0	195.0	209.2	143.0	100.0
W_8	392.0	286.0	194.0	207.8	144.8	100.0

LEMON-1xW_4

	99(%)	90(%)	50(%)	MEAN	STD	RX(%)
W_9	449.0	338.0	236.0	249.5	94.94	100.0
W_18	400.0	299.0	203.0	215.4	69.67	100.0
W_19	399.0	298.0	202.0	214.5	64.15	100.0
W_21	383.0	283.0	193.0	206.3	81.9	100.0
W_8	377.0	277.0	190.0	202.6	56.15	100.0
W_11	373.0	276.0	189.0	202.0	57.5	100.0
W_17	367.0	267.0	181.0	192.9	56.7	100.0
W_22	356.0	257.0	172.0	185.8	53.72	100.0

LEMON-2xW_5

	99(%)	90(%)	50(%)	MEAN	STD	RX(%)
W_9	437.0	325.0	223.0	236.7	76.39	100.0
W_16	429.0	320.0	219.0	231.0	74.5	100.0
W_22	434.0	319.0	211.0	224.9	71.36	100.0
W_10	412.0	303.0	200.0	213.1	71.07	100.0
W_11	405.0	302.0	203.0	216.5	72.88	100.0
W_18	397.0	295.0	200.0	211.4	72.03	100.0
W_19	321.0	235.0	153.0	166.4	50.59	100.0
W_21	320.0	231.0	153.0	165.7	61.93	100.0



- **LemonDrop(Epsilon, MAX_I): Smaller Epsilon => Pickier Solution**
- **HEURISTIC** shows comparable performance to **LemonDrop** => **358us vs 338us vs 325us**

Background: Jasper

10 Receivers		
D	F	OML (µs)
1	10	66
2	4	88

100 Receivers		
D	F	OML (µs)
1	100	351
2	10	139
3	5	141

1000 Receivers		
D	F	OML (µs)
2	32	282
3	10	217
4	6	226

- $D = Log_{10}(N)$
- N = 100, D=2, F=10
- N = 1000, D=3, F=10

Results

EVALUATION COMPARISON - BEST x LEMONS

N=15, D=3, F=2, K=4, POOL=24, Rate=10000 p/sec, Duration=30 sec

BEST-heuristic-W_3

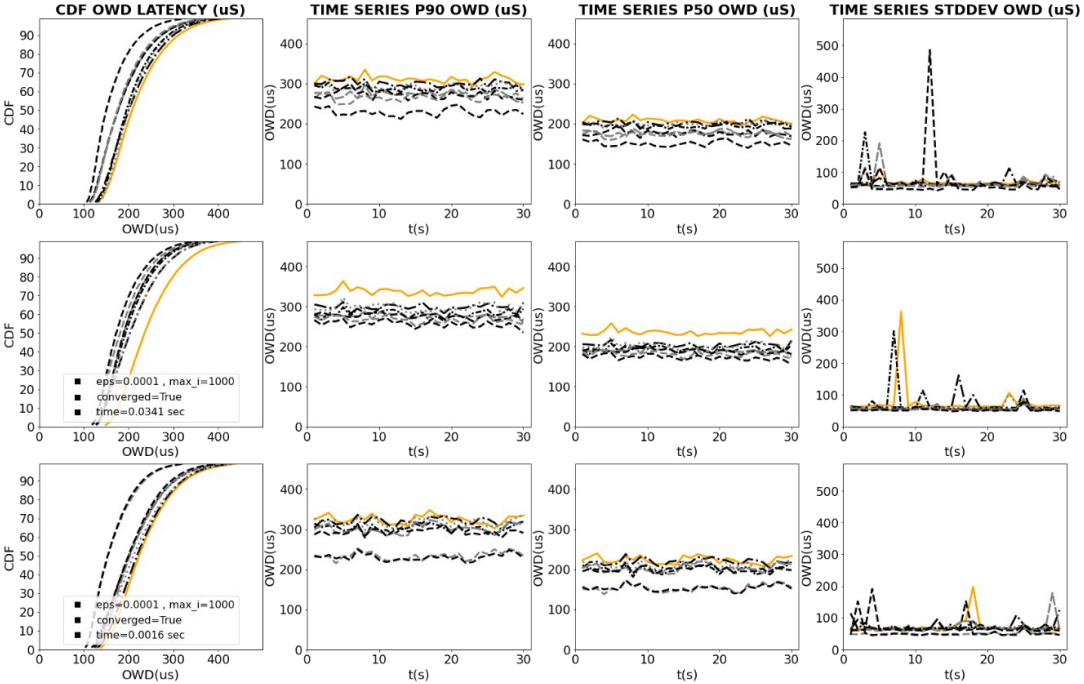
	99(%)	90(%)	50(%)	MEAN	STD	RX(%)
W_9	428.0	312.0	208.0	223.1	68.6	100.0
W_4	407.0	300.0	200.0	213.8	67.13	100.0
W_19	403.0	293.0	200.0	214.1	62.75	100.0
W_1	398.0	289.0	195.0	209.6	72.58	100.0
W_10	385.0	274.0	176.0	191.6	70.17	100.0
W_17	381.0	273.0	176.0	192.2	109.5	100.0
W_8	376.0	267.0	175.0	189.4	64.88	100.0
W_13	328.0	230.0	150.0	164.6	56.76	100.0

LEMON-1xW_4

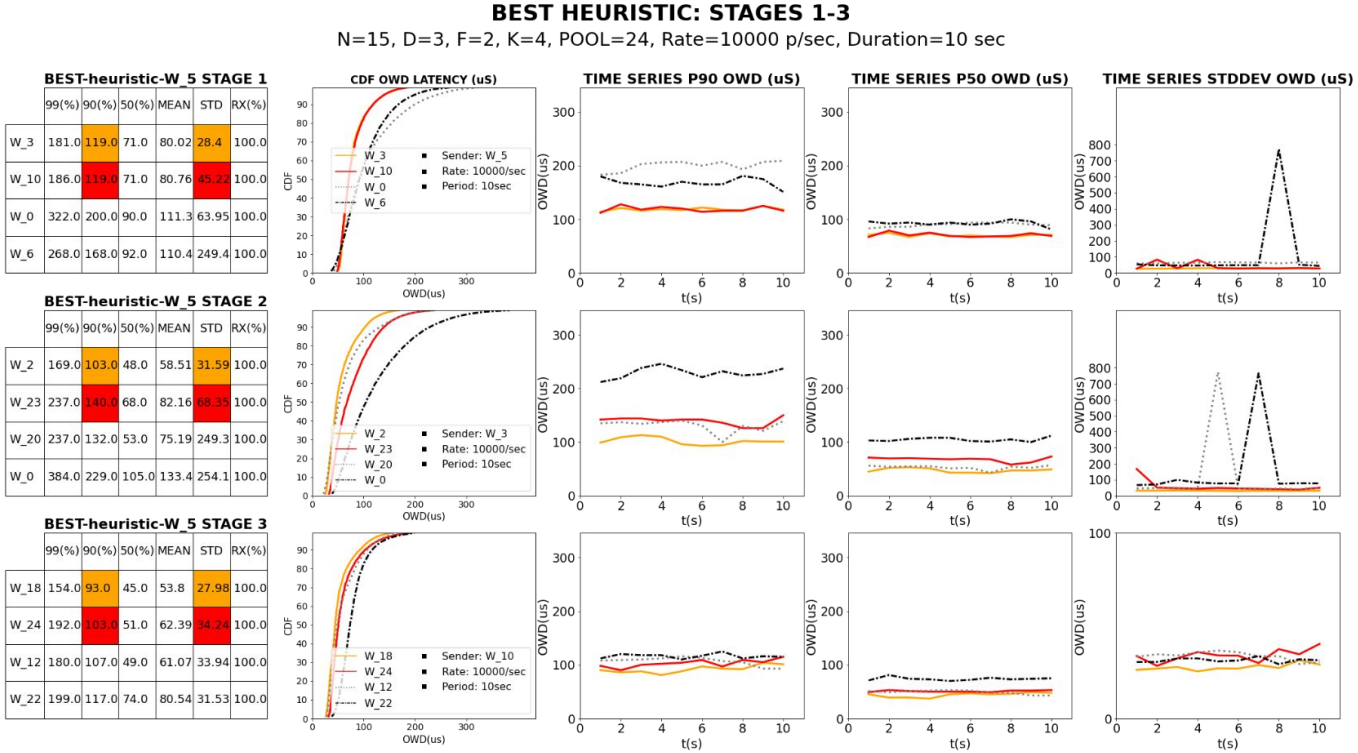
	99(%)	90(%)	50(%)	MEAN	STD	RX(%)
W_9	449.0	338.0	236.0	249.5	94.94	100.0
W_18	400.0	299.0	203.0	215.4	69.67	100.0
W_19	399.0	298.0	202.0	214.5	64.15	100.0
W_21	383.0	283.0	193.0	206.3	81.9	100.0
W_8	377.0	277.0	190.0	202.6	56.15	100.0
W_11	373.0	276.0	189.0	202.0	57.5	100.0
W_17	367.0	267.0	181.0	192.9	56.7	100.0
W_22	356.0	257.0	172.0	185.8	53.72	100.0

LEMON-2xW_5

	99(%)	90(%)	50(%)	MEAN	STD	RX(%)
W_9	437.0	325.0	223.0	236.7	76.39	100.0
W_16	429.0	320.0	219.0	231.0	74.5	100.0
W_22	434.0	319.0	211.0	224.9	71.36	100.0
W_10	412.0	303.0	200.0	213.1	71.07	100.0
W_11	405.0	302.0	203.0	216.5	72.88	100.0
W_18	397.0	295.0	200.0	211.4	72.03	100.0
W_19	321.0	235.0	153.0	166.4	50.59	100.0
W_21	320.0	231.0	153.0	165.7	61.93	100.0



Results



	99(%)	90(%)	50(%)	MEAN	STD	RX(%)
W_3	181.0	119.0	71.0	80.02	28.4	100.0
W_10	186.0	119.0	71.0	80.76	45.22	100.0
W_0	322.0	200.0	90.0	111.3	63.95	100.0
W_6	268.0	168.0	92.0	110.4	249.4	100.0

BEST-heuristic-W_5 STAGE 2

	99(%)	90(%)	50(%)	MEAN	STD	RX(%)
W_2	169.0	103.0	48.0	58.51	31.59	100.0
W_23	237.0	140.0	68.0	82.16	68.35	100.0
W_20	237.0	132.0	53.0	75.19	249.3	100.0
W_0	384.0	229.0	105.0	133.4	254.1	100.0

BEST-heuristic-W_5 STAGE 3

	99(%)	90(%)	50(%)	MEAN	STD	RX(%)
W_18	154.0	93.0	45.0	53.8	27.98	100.0
W_24	192.0	103.0	51.0	62.39	34.24	100.0
W_12	180.0	107.0	49.0	61.07	33.94	100.0
W_22	199.0	117.0	74.0	80.54	31.53	100.0

- **Simple quick selection** seems to at least prune bad VMs

Results

BEST HEURISTIC: STAGES 4-7

N=15, D=3, F=2, K=4, POOL=24, Rate=10000 p/sec, Duration=10 sec

BEST-heuristic-W_5 STAGE 4

	99(%)	90(%)	50(%)	MEAN	STD	RX(%)
W_17	201.0	116.0	58.0	69.77	36.24	100.0
W_8	198.0	112.0	56.0	67.73	41.65	100.0
W_9	239.0	134.0	82.0	90.96	37.88	100.0
W_13	153.0	99.0	62.0	69.52	73.59	100.0

BEST-heuristic-W_5 STAGE 5

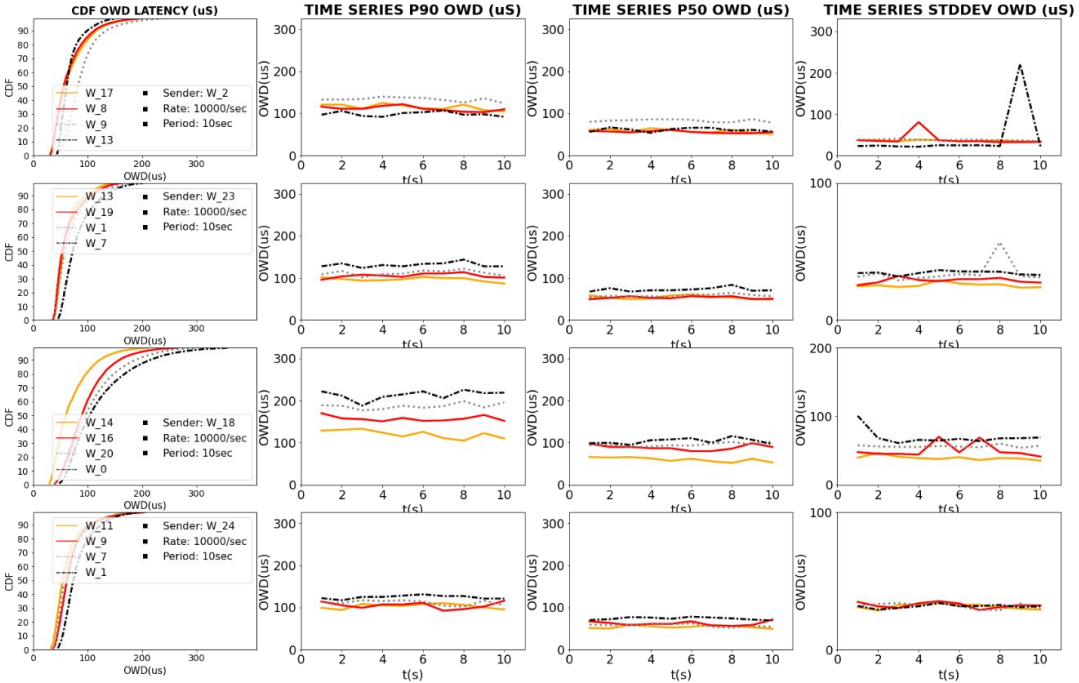
	99(%)	90(%)	50(%)	MEAN	STD	RX(%)
W_13	151.0	97.0	55.0	63.11	25.83	100.0
W_19	162.0	106.0	54.0	64.45	29.1	100.0
W_1	187.0	113.0	58.0	69.61	35.41	100.0
W_7	202.0	132.0	73.0	83.96	34.76	100.0

BEST-heuristic-W_5 STAGE 6

	99(%)	90(%)	50(%)	MEAN	STD	RX(%)
W_14	211.0	122.0	60.0	72.08	39.73	100.0
W_16	265.0	158.0	89.0	100.1	51.49	100.0
W_20	300.0	187.0	96.0	110.5	56.43	100.0
W_0	360.0	214.0	103.0	124.3	70.57	100.0

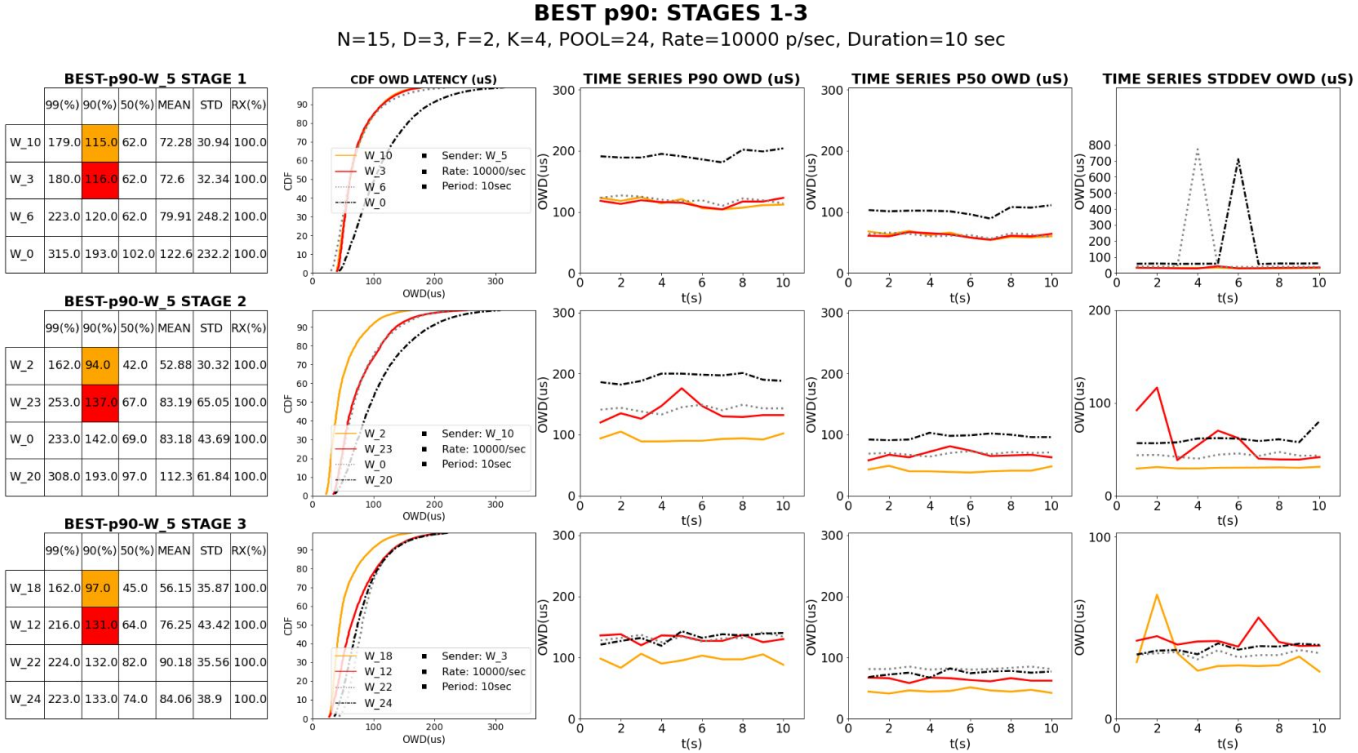
BEST-heuristic-W_5 STAGE 7

	99(%)	90(%)	50(%)	MEAN	STD	RX(%)
W_11	179.0	103.0	54.0	63.96	31.53	100.0
W_9	204.0	105.0	62.0	70.88	32.72	100.0
W_7	181.0	112.0	58.0	68.75	32.05	100.0
W_1	194.0	125.0	74.0	82.79	31.6	100.0



- **Simple quick selection** seems to at least prune bad VMs

Results



	99(%)	90(%)	50(%)	MEAN	STD	RX(%)
W_10	179.0	115.0	62.0	72.28	30.94	100.0
W_3	180.0	116.0	62.0	72.6	32.34	100.0
W_6	223.0	120.0	62.0	79.91	248.2	100.0
W_0	315.0	193.0	102.0	122.6	232.2	100.0

	99(%)	90(%)	50(%)	MEAN	STD	RX(%)
W_2	162.0	94.0	42.0	52.88	30.32	100.0
W_23	253.0	137.0	67.0	83.19	65.05	100.0
W_0	233.0	142.0	69.0	83.18	43.69	100.0
W_20	308.0	193.0	97.0	112.3	61.84	100.0

	99(%)	90(%)	50(%)	MEAN	STD	RX(%)
W_18	162.0	97.0	45.0	56.15	35.87	100.0
W_12	216.0	131.0	64.0	76.25	43.42	100.0
W_22	224.0	132.0	82.0	90.18	35.56	100.0
W_24	223.0	133.0	74.0	84.06	38.9	100.0

- **Simple quick selection** seems to at least prune bad VMs

Results

BEST p90: STAGES 4-7

N=15, D=3, F=2, K=4, POOL=24, Rate=10000 p/sec, Duration=10 sec

BEST-p90-W_5 STAGE 4

	99(%)	90(%)	50(%)	MEAN	STD	RX(%)
W_19	161.0	99.0	57.0	65.79	63.22	100.0
W_8	190.0	108.0	56.0	66.81	39.66	100.0
W_9	227.0	126.0	80.0	87.76	35.09	100.0
W_14	260.0	162.0	89.0	101.5	74.17	100.0

BEST-p90-W_5 STAGE 5

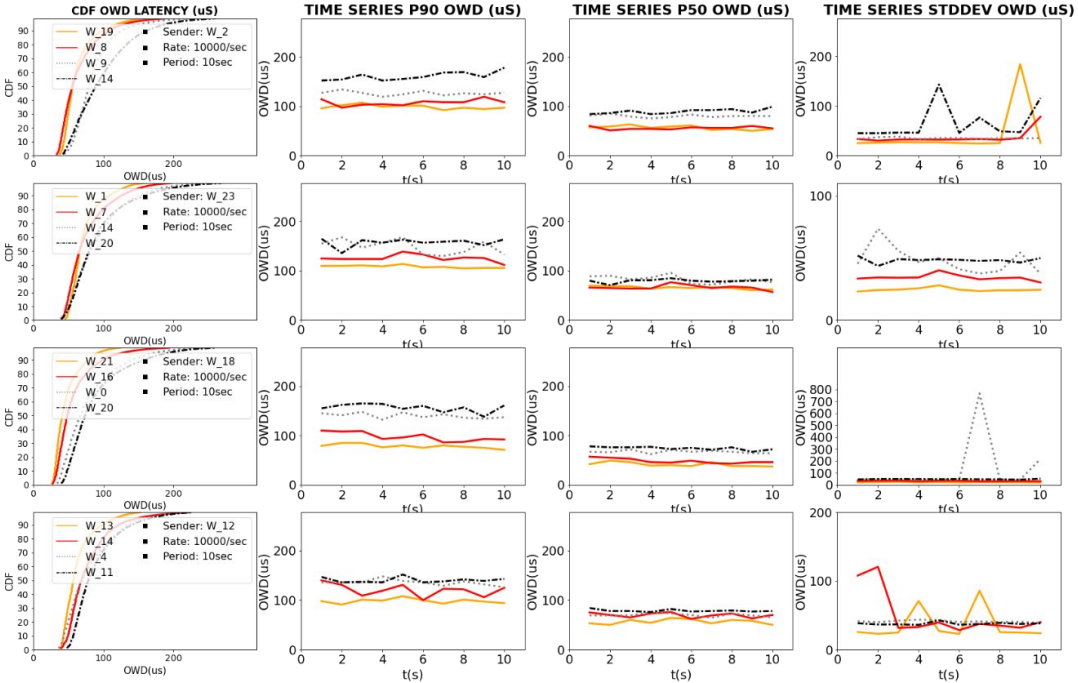
	99(%)	90(%)	50(%)	MEAN	STD	RX(%)
W_1	159.0	109.0	66.0	73.93	24.8	100.0
W_7	197.0	126.0	66.0	77.85	34.74	100.0
W_14	255.0	150.0	82.0	93.88	49.81	100.0
W_20	268.0	159.0	80.0	94.28	48.48	100.0

BEST-p90-W_5 STAGE 6

	99(%)	90(%)	50(%)	MEAN	STD	RX(%)
W_21	124.0	78.0	42.0	48.75	21.93	100.0
W_16	194.0	99.0	49.0	59.48	32.95	100.0
W_0	249.0	140.0	67.0	89.57	256.4	100.0
W_20	258.0	157.0	74.0	89.83	48.22	100.0

BEST-p90-W_5 STAGE 7

	99(%)	90(%)	50(%)	MEAN	STD	RX(%)
W_13	153.0	99.0	57.0	65.26	41.63	100.0
W_14	222.0	122.0	69.0	80.98	60.11	100.0
W_4	223.0	136.0	70.0	82.28	41.12	100.0
W_11	225.0	141.0	79.0	90.69	37.87	100.0



- **Simple quick selection** seems to at least prune bad VMs

Results

BEST p50: STAGES 1-3

N=15, D=3, F=2, K=4, POOL=24, Rate=10000 p/sec, Duration=10 sec

BEST-p50-W_5 STAGE 1

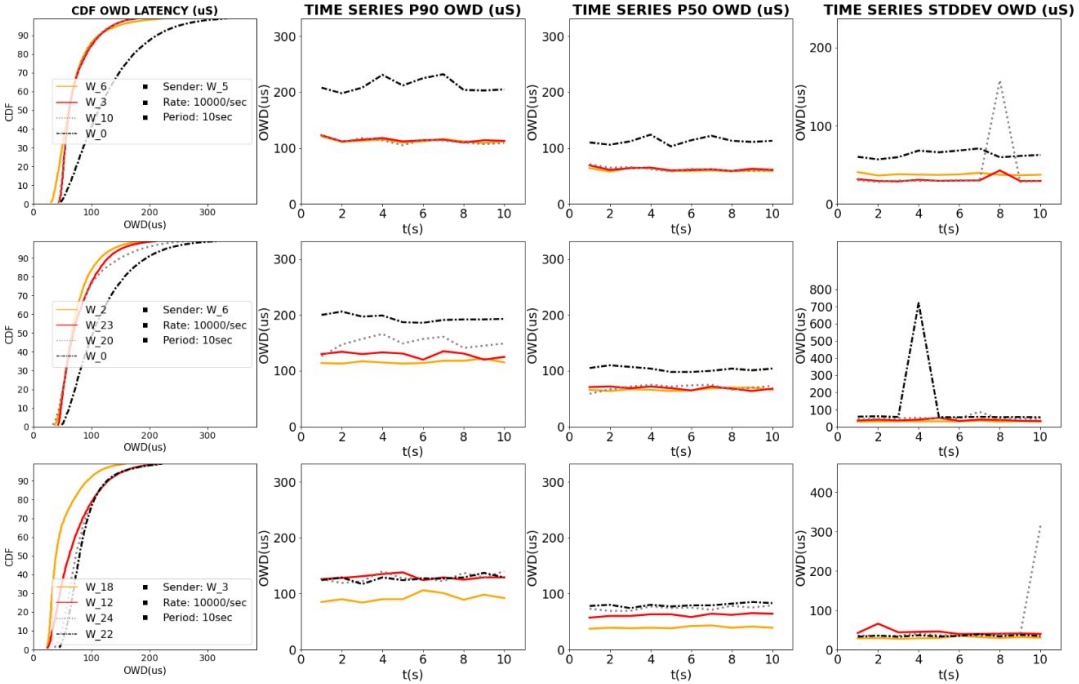
	99(%)	90(%)	50(%)	MEAN	STD	RX(%)
W_6	225.0	113.0	61.0	70.9	37.97	100.0
W_3	179.0	115.0	62.0	73.59	31.56	100.0
W_10	182.0	113.0	63.0	73.77	57.47	100.0
W_0	334.0	213.0	113.0	127.7	63.93	100.0

BEST-p50-W_5 STAGE 2

	99(%)	90(%)	50(%)	MEAN	STD	RX(%)
W_2	188.0	116.0	67.0	75.31	31.78	100.0
W_23	211.0	129.0	69.0	81.2	39.57	100.0
W_20	261.0	150.0	70.0	85.73	53.3	100.0
W_0	315.0	195.0	103.0	124.7	235.0	100.0

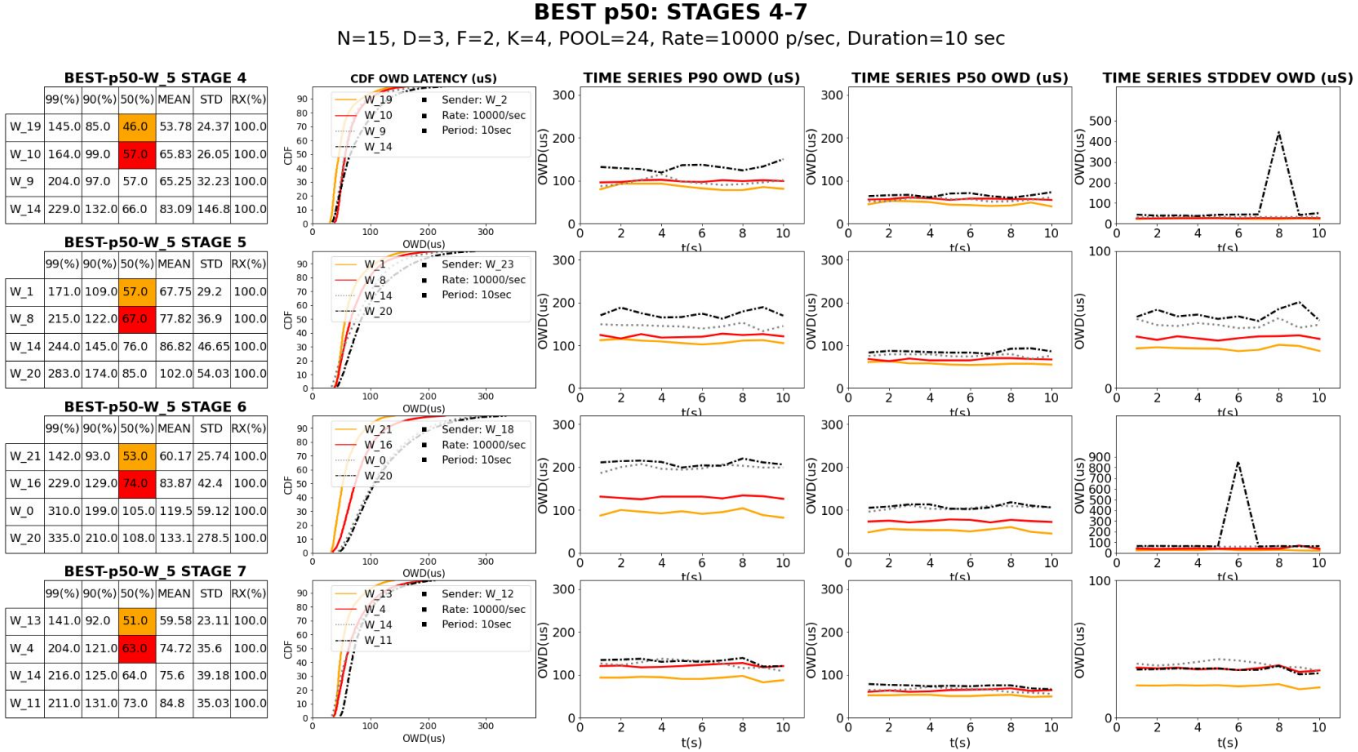
BEST-p50-W_5 STAGE 3

	99(%)	90(%)	50(%)	MEAN	STD	RX(%)
W_18	158.0	93.0	39.0	51.25	30.15	100.0
W_12	215.0	129.0	62.0	73.77	44.87	100.0
W_24	223.0	129.0	74.0	85.72	105.8	100.0
W_22	221.0	127.0	80.0	87.76	35.05	100.0



- **Simple quick selection** seems to at least prune bad VMs

Results



	99(%)	90(%)	50(%)	MEAN	STD	RX(%)
W_19	145.0	85.0	46.0	53.78	24.37	100.0
W_10	164.0	99.0	57.0	65.83	26.05	100.0
W_9	204.0	97.0	57.0	65.25	32.23	100.0
W_14	229.0	132.0	66.0	83.09	146.8	100.0

BEST-p50-W_5 STAGE 5

	99(%)	90(%)	50(%)	MEAN	STD	RX(%)
W_1	171.0	109.0	57.0	67.75	29.2	100.0
W_8	215.0	122.0	67.0	77.82	36.9	100.0
W_14	244.0	145.0	76.0	86.82	46.65	100.0
W_20	283.0	174.0	85.0	102.0	54.03	100.0

BEST-p50-W_5 STAGE 6

	99(%)	90(%)	50(%)	MEAN	STD	RX(%)
W_21	142.0	93.0	53.0	60.17	25.74	100.0
W_16	229.0	129.0	74.0	83.87	42.4	100.0
W_0	310.0	199.0	105.0	119.5	59.12	100.0
W_20	335.0	210.0	108.0	133.1	278.5	100.0

BEST-p50-W_5 STAGE 7

	99(%)	90(%)	50(%)	MEAN	STD	RX(%)
W_13	141.0	92.0	51.0	59.58	23.11	100.0
W_4	204.0	121.0	63.0	74.72	35.6	100.0
W_14	216.0	125.0	64.0	75.6	39.18	100.0
W_11	211.0	131.0	73.0	84.8	35.03	100.0

- **Simple quick selection** seems to at least prune bad VMs

Results

LEMON PARAMETRIZATION

N=15, D=3, F=2, K=4, POOL=24, Rate=10000 p/sec, Duration=30 sec

LEMON-A-1xW_1

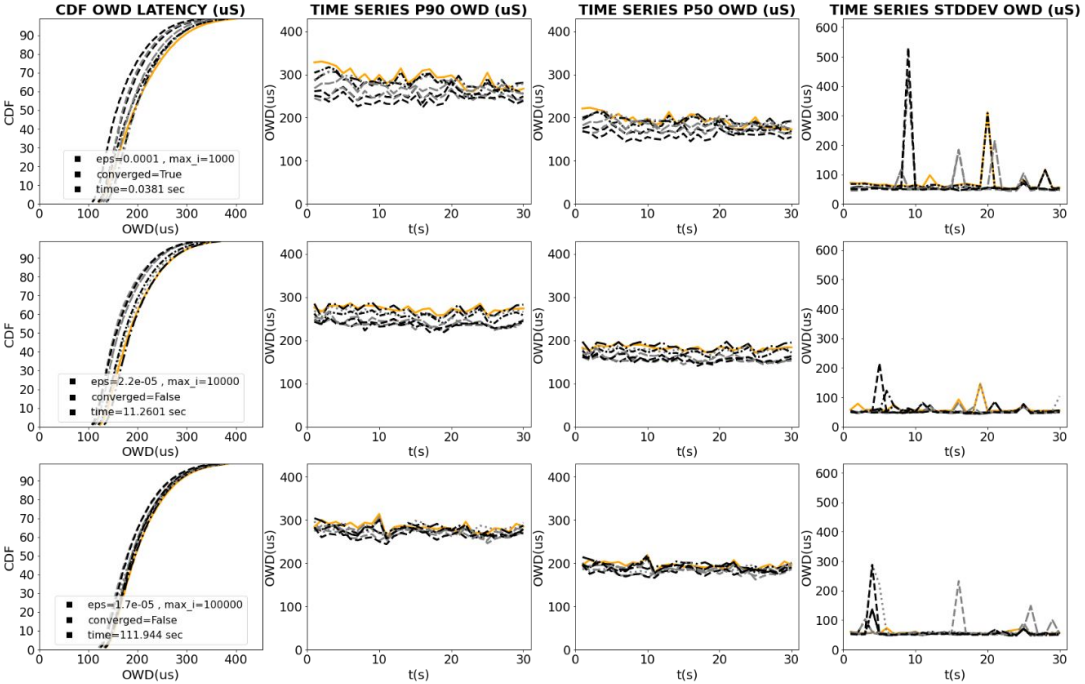
	99(%)	90(%)	50(%)	MEAN	STD	RX(%)
W_22	404.0	295.0	193.0	209.1	88.35	100.0
W_17	382.0	285.0	195.0	209.8	99.75	100.0
W_13	380.0	285.0	196.0	211.2	99.95	100.0
W_18	389.0	283.0	185.0	200.3	85.03	100.0
W_11	368.0	271.0	185.0	199.4	76.68	100.0
W_23	353.0	256.0	173.0	186.1	52.02	100.0
W_14	349.0	251.0	170.0	185.2	62.71	100.0
W_9	338.0	242.0	158.0	173.2	109.6	100.0

LEMON-B-1xW_1

	99(%)	90(%)	50(%)	MEAN	STD	RX(%)
W_11	369.0	273.0	183.0	197.1	64.19	100.0
W_23	369.0	272.0	186.0	200.0	60.13	100.0
W_9	360.0	265.0	177.0	191.2	62.17	100.0
W_14	356.0	258.0	173.0	187.3	56.36	100.0
W_13	330.0	244.0	161.0	175.2	51.33	100.0
W_18	328.0	238.0	155.0	169.8	63.81	100.0
W_22	327.0	237.0	154.0	168.0	51.5	100.0
W_17	325.0	237.0	157.0	169.6	50.21	100.0

LEMON-C-1xW_1

	99(%)	90(%)	50(%)	MEAN	STD	RX(%)
W_17	384.0	285.0	195.0	209.3	57.64	100.0
W_23	378.0	282.0	195.0	207.4	57.87	99.88
W_11	376.0	280.0	188.0	202.0	86.96	99.88
W_13	369.0	276.0	190.0	203.9	58.18	100.0
W_9	372.0	276.0	188.0	202.1	62.44	100.0
W_18	367.0	275.0	190.0	203.1	58.47	100.0
W_22	364.0	268.0	179.0	193.1	71.42	99.88
W_14	359.0	266.0	180.0	194.0	74.64	99.88



- **Smaller epsilon: negligible performance enhancement & too long to converge**

Results

LEMON-A RUNS

N=15, D=3, F=2, K=4, POOL=24, Rate=10000 p/sec, Duration=30 sec

LEMON-A-1xW_1

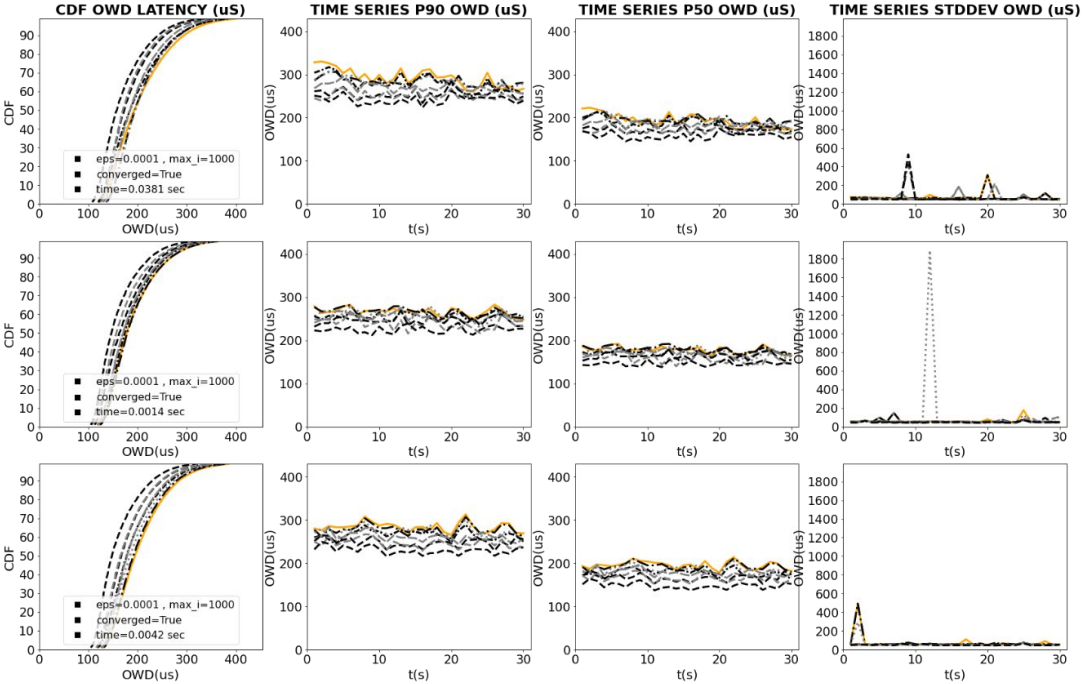
	99(%)	90(%)	50(%)	MEAN	STD	RX(%)
W_22	404.0	295.0	193.0	209.1	88.35	100.0
W_17	382.0	285.0	195.0	209.8	99.75	100.0
W_13	380.0	285.0	196.0	211.2	99.95	100.0
W_18	389.0	283.0	185.0	200.3	85.03	100.0
W_11	368.0	271.0	185.0	199.4	76.68	100.0
W_23	353.0	256.0	173.0	186.1	52.02	100.0
W_14	349.0	251.0	170.0	185.2	62.71	100.0
W_9	338.0	242.0	158.0	173.2	109.6	100.0

LEMON-A-2xW_1

	99(%)	90(%)	50(%)	MEAN	STD	RX(%)
W_15	364.0	267.0	178.0	192.7	63.25	100.0
W_18	363.0	266.0	179.0	193.3	54.49	100.0
W_10	358.0	259.0	171.0	194.1	353.1	100.0
W_13	348.0	258.0	174.0	187.9	51.97	100.0
W_22	345.0	254.0	171.0	184.4	60.8	100.0
W_17	338.0	247.0	163.0	177.4	62.83	100.0
W_23	334.0	240.0	157.0	171.0	59.29	100.0
W_14	314.0	225.0	147.0	160.7	48.19	100.0

LEMON-A-3xW_1

	99(%)	90(%)	50(%)	MEAN	STD	RX(%)
W_23	385.0	284.0	195.0	209.3	108.2	100.0
W_14	378.0	279.0	191.0	206.0	106.0	100.0
W_15	370.0	274.0	185.0	199.0	76.79	100.0
W_22	363.0	267.0	178.0	191.7	56.04	100.0
W_13	353.0	262.0	179.0	192.6	51.79	100.0
W_17	338.0	249.0	167.0	180.6	50.78	100.0
W_18	338.0	247.0	165.0	178.9	51.18	100.0
W_9	318.0	230.0	148.0	162.8	49.84	100.0



Results

LEMON-B RUNS

N=15, D=3, F=2, K=4, POOL=24, Rate=10000 p/sec, Duration=30 sec

LEMON-B-1xW_1

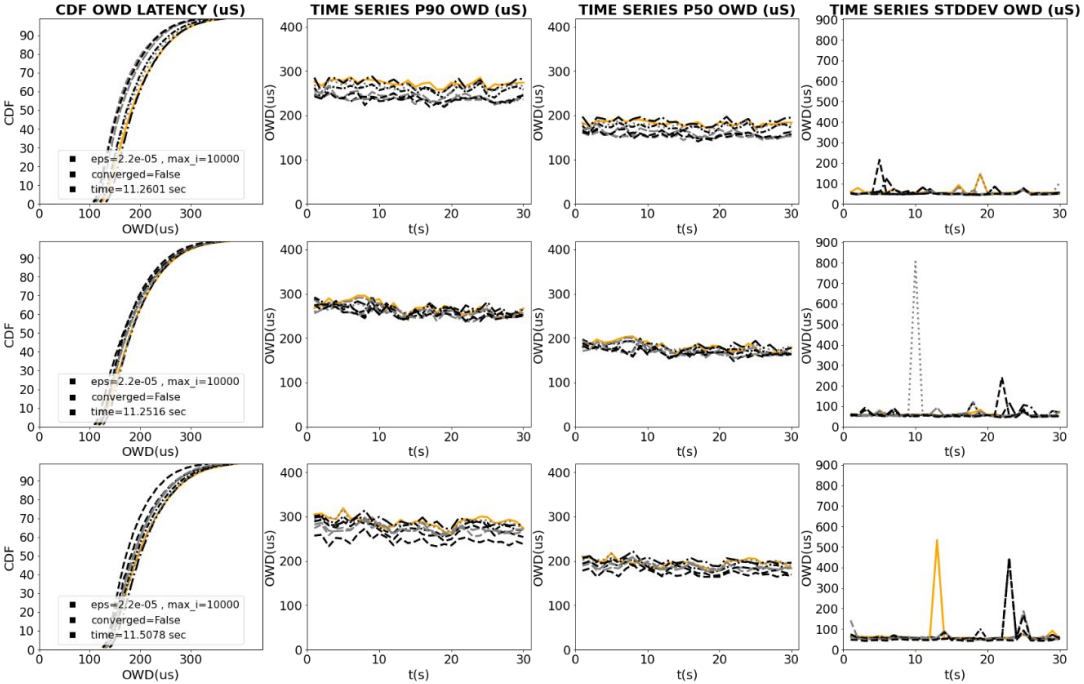
	99(%)	90(%)	50(%)	MEAN	STD	RX(%)
W_11	369.0	273.0	183.0	197.1	64.19	100.0
W_23	369.0	272.0	186.0	200.0	60.13	100.0
W_9	360.0	265.0	177.0	191.2	62.17	100.0
W_14	356.0	258.0	173.0	187.3	56.36	100.0
W_13	330.0	244.0	161.0	175.2	51.33	100.0
W_18	328.0	238.0	155.0	169.8	63.81	100.0
W_22	327.0	237.0	154.0	168.0	51.5	100.0
W_17	325.0	237.0	157.0	169.6	50.21	100.0

LEMON-B-2xW_1

	99(%)	90(%)	50(%)	MEAN	STD	RX(%)
W_18	377.0	271.0	182.0	194.9	61.16	100.0
W_13	372.0	269.0	182.0	196.8	66.96	100.0
W_9	368.0	266.0	181.0	197.7	159.2	100.0
W_22	366.0	265.0	174.0	188.4	57.38	100.0
W_17	371.0	264.0	175.0	188.3	61.7	100.0
W_15	360.0	261.0	167.0	182.0	74.44	100.0
W_23	355.0	257.0	169.0	184.1	53.54	100.0
W_14	349.0	255.0	168.0	183.1	55.6	100.0

LEMON-B-3xW_1

	99(%)	90(%)	50(%)	MEAN	STD	RX(%)
W_22	392.0	290.0	195.0	211.2	114.5	100.0
W_13	385.0	289.0	200.0	213.4	58.22	100.0
W_23	387.0	287.0	196.0	209.3	57.74	100.0
W_17	384.0	281.0	191.0	206.7	100.7	100.0
W_18	370.0	273.0	186.0	200.0	65.45	100.0
W_9	370.0	271.0	182.0	197.5	102.1	100.0
W_15	360.0	270.0	187.0	200.3	59.33	100.0
W_14	334.0	247.0	172.0	184.8	51.02	100.0



Results

LEMON-C RUNS

N=15, D=3, F=2, K=4, POOL=24, Rate=10000 p/sec, Duration=30 sec

LEMON-C-1xW_1

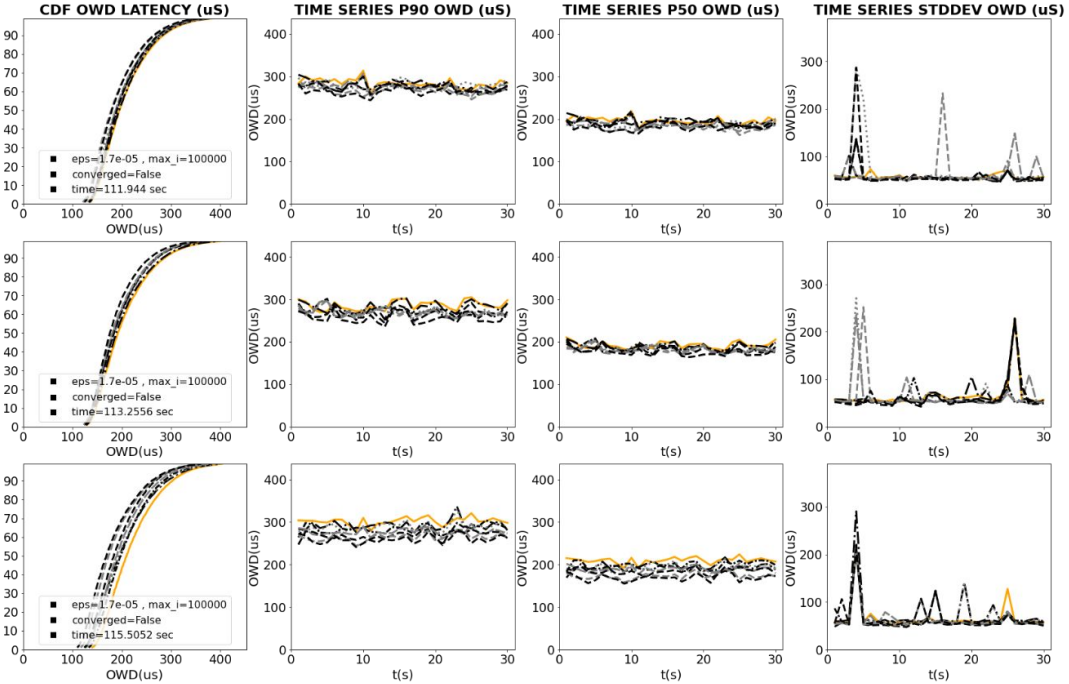
	99(%)	90(%)	50(%)	MEAN	STD	RX(%)
W_17	384.0	285.0	195.0	209.3	57.64	100.0
W_23	378.0	282.0	195.0	207.4	57.87	99.88
W_11	376.0	280.0	188.0	202.0	86.96	99.88
W_13	369.0	276.0	190.0	203.9	58.18	100.0
W_9	372.0	276.0	188.0	202.1	62.44	100.0
W_18	367.0	275.0	190.0	203.1	58.47	100.0
W_22	364.0	268.0	179.0	193.1	71.42	99.88
W_14	359.0	266.0	180.0	194.0	74.64	99.88

LEMON-C-2xW_1

	99(%)	90(%)	50(%)	MEAN	STD	RX(%)
W_17	399.0	287.0	192.0	207.8	72.61	100.0
W_13	394.0	283.0	189.0	204.8	73.49	100.0
W_9	370.0	273.0	184.0	199.1	75.69	100.0
W_18	366.0	271.0	184.0	197.4	57.55	100.0
W_15	365.0	270.0	182.0	196.2	72.74	100.0
W_23	377.0	270.0	184.0	198.6	71.22	100.0
W_11	364.0	268.0	181.0	194.8	70.82	100.0
W_14	364.0	257.0	176.0	190.7	69.63	100.0

LEMON-C-3xW_1

	99(%)	90(%)	50(%)	MEAN	STD	RX(%)
W_13	403.0	303.0	210.0	223.0	73.33	100.0
W_11	398.0	292.0	195.0	209.4	85.02	100.0
W_17	392.0	289.0	194.0	208.3	75.75	100.0
W_15	381.0	285.0	198.0	210.6	71.89	100.0
W_14	376.0	277.0	190.0	203.0	69.41	100.0
W_9	361.0	269.0	184.0	197.1	69.94	100.0
W_23	367.0	265.0	173.0	188.8	69.98	100.0
W_18	361.0	261.0	169.0	184.0	72.46	100.0



Results

EVALUATION COMPARISON - BEST x WORST x RAND

N=15, D=3, F=2, K=4, POOL=24, Rate=10000 p/sec, Duration=30 sec

BEST-heuristic-W_3

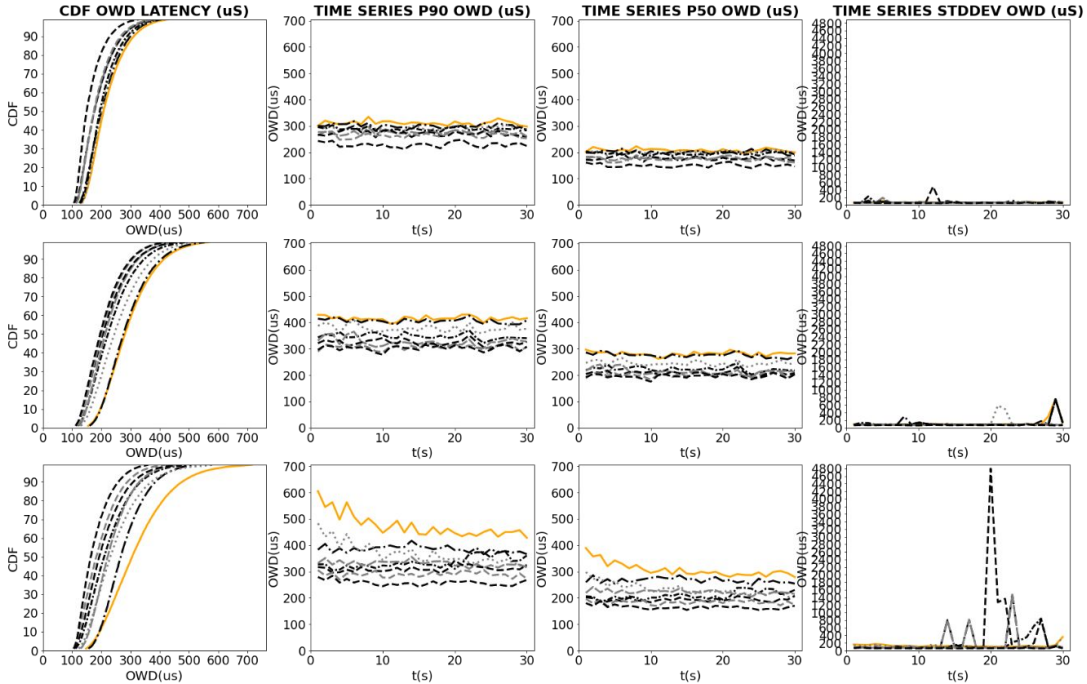
	99(%)	90(%)	50(%)	MEAN	STD	RX(%)
W_9	428.0	312.0	208.0	223.18	68.6	100.0
W_4	407.0	300.0	200.0	213.84	67.13	100.0
W_19	403.0	293.0	200.0	214.12	62.75	100.0
W_1	398.0	289.0	195.0	209.66	72.58	100.0
W_10	385.0	274.0	176.0	191.67	70.17	100.0
W_17	381.0	273.0	176.0	192.21	109.59	100.0
W_8	376.0	267.0	175.0	189.45	64.88	100.0
W_13	328.0	230.0	150.0	164.68	56.76	100.0

WORST-p90-W_3

	99(%)	90(%)	50(%)	MEAN	STD	RX(%)
W_6	568.0	415.0	281.0	299.19	172.97	100.0
W_9	554.0	406.0	276.0	294.8	173.35	100.0
W_20	526.0	378.0	245.0	262.98	164.58	100.0
W_14	491.0	348.0	223.0	240.99	164.56	100.0
W_8	456.0	327.0	214.0	228.73	77.47	100.0
W_18	460.0	323.0	206.0	223.71	159.7	100.0
W_12	424.0	307.0	201.0	214.52	68.94	100.0
W_15	421.0	303.0	196.0	209.82	70.15	100.0

RANDxW_3

	99(%)	90(%)	50(%)	MEAN	STD	RX(%)
W_20	714.0	477.0	306.0	326.82	139.82	100.0
W_22	523.0	386.0	267.0	291.76	350.35	100.0
W_21	586.0	372.0	231.0	252.98	122.74	100.0
W_16	492.0	336.0	210.0	233.77	229.18	100.0
W_19	460.0	331.0	222.0	246.79	349.25	100.0
W_1	471.0	310.0	193.0	256.17	971.11	100.0
W_14	418.0	290.0	183.0	198.06	79.52	100.0
W_18	361.0	259.0	164.0	179.22	61.21	100.0



Results

BEST HEURISTIC: STAGES 1-3

N=15, D=3, F=2, K=4, POOL=24, Rate=10000 p/sec, Duration=10 sec

BEST-heuristic-W_3 STAGE 1

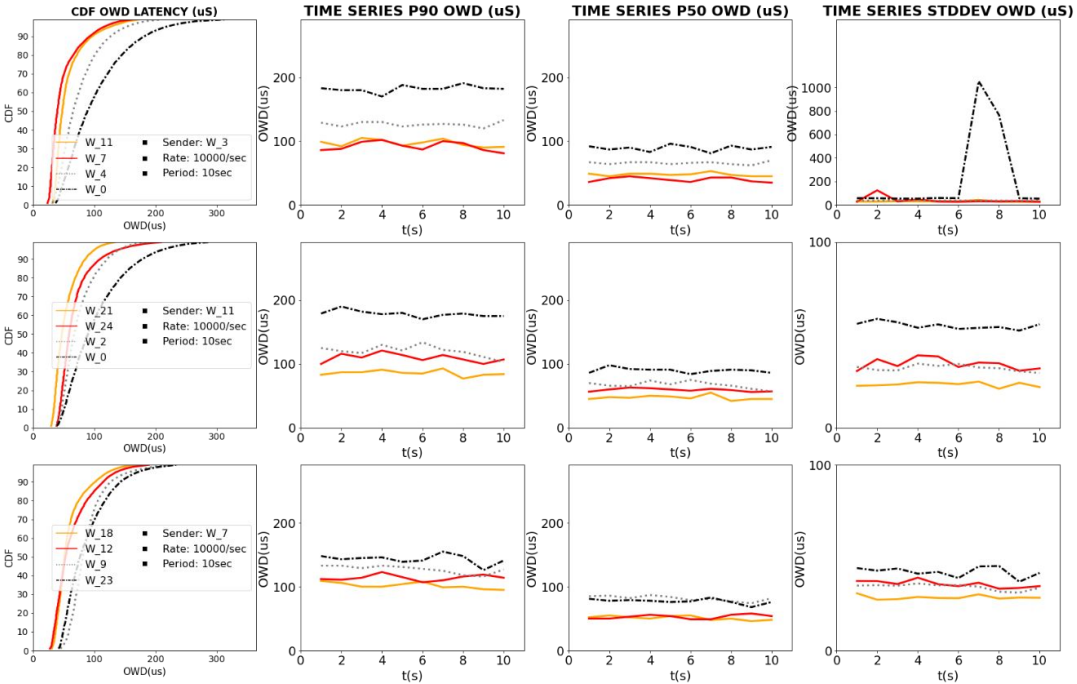
	99(%)	90(%)	50(%)	MEAN	STD	RX(%)
W_11	174.0	97.0	48.0	58.82	31.71	100.0
W_7	163.0	93.0	40.0	51.76	50.24	100.0
W_4	210.0	127.0	66.0	76.62	38.3	100.0
W_0	314.0	182.0	89.0	125.8	417.0	100.0

BEST-heuristic-W_3 STAGE 2

	99(%)	90(%)	50(%)	MEAN	STD	RX(%)
W_21	134.0	86.0	47.0	54.73	23.52	100.0
W_24	209.0	110.0	59.0	70.18	34.59	100.0
W_2	186.0	121.0	67.0	76.88	32.63	100.0
W_0	286.0	179.0	90.0	104.7	55.16	100.0

BEST-heuristic-W_3 STAGE 3

	99(%)	90(%)	50(%)	MEAN	STD	RX(%)
W_18	165.0	102.0	52.0	61.08	28.83	100.0
W_12	189.0	115.0	53.0	65.58	36.0	100.0
W_9	227.0	127.0	82.0	89.55	34.55	100.0
W_23	234.0	143.0	77.0	89.32	42.66	100.0



Results

BEST HEURISTIC: STAGES 4-7

N=15, D=3, F=2, K=4, POOL=24, Rate=10000 p/sec, Duration=10 sec

BEST-heuristic-W_3 STAGE 4

	99(%)	90(%)	50(%)	MEAN	STD	RX(%)
W_13	129.0	74.0	43.0	50.21	20.42	100.0
W_8	192.0	107.0	51.0	63.26	34.1	100.0
W_17	207.0	118.0	61.0	71.37	37.47	100.0
W_6	297.0	169.0	87.0	154.9	874.1	100.0

BEST-heuristic-W_3 STAGE 5

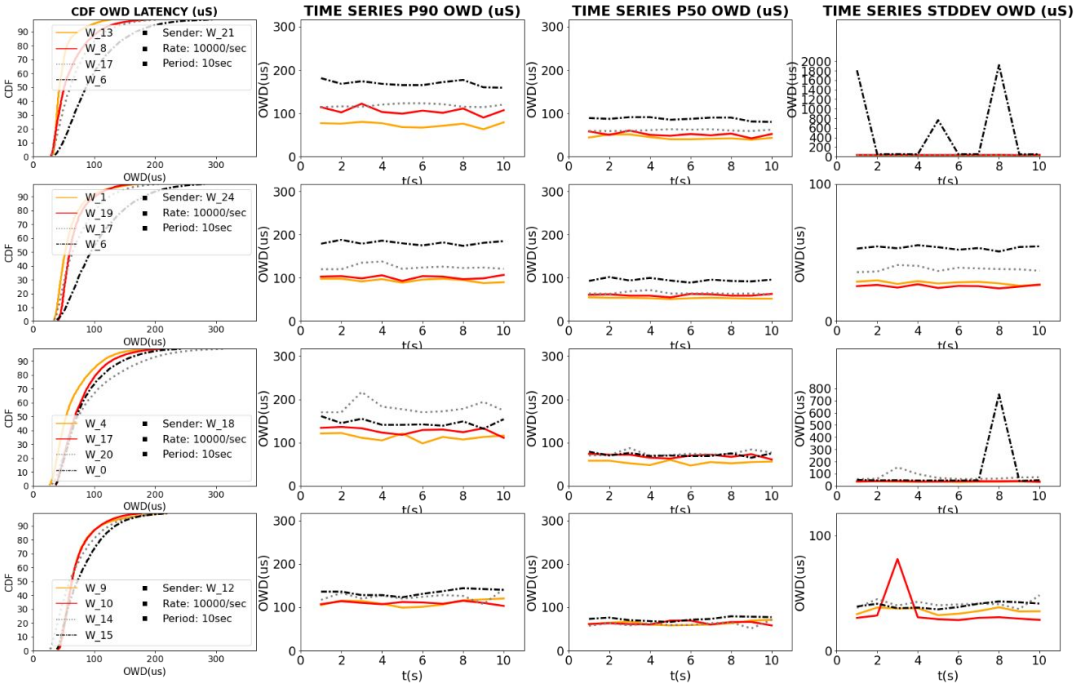
	99(%)	90(%)	50(%)	MEAN	STD	RX(%)
W_1	172.0	94.0	53.0	61.67	27.98	100.0
W_19	161.0	102.0	60.0	68.36	25.58	100.0
W_17	206.0	126.0	65.0	76.48	38.17	100.0
W_6	281.0	181.0	95.0	108.4	53.58	100.0

BEST-heuristic-W_3 STAGE 6

	99(%)	90(%)	50(%)	MEAN	STD	RX(%)
W_4	193.0	114.0	54.0	66.18	35.66	100.0
W_17	211.0	128.0	69.0	79.47	37.46	100.0
W_20	315.0	181.0	74.0	96.92	80.06	100.0
W_0	245.0	146.0	72.0	92.87	242.7	100.0

BEST-heuristic-W_3 STAGE 7

	99(%)	90(%)	50(%)	MEAN	STD	RX(%)
W_9	219.0	111.0	64.0	73.63	34.85	100.0
W_10	177.0	110.0	64.0	73.11	36.84	100.0
W_14	222.0	125.0	60.0	72.53	41.23	100.0
W_15	219.0	135.0	73.0	83.8	39.51	100.0



Results

EVALUATION COMPARISON - BEST ITERATIONS

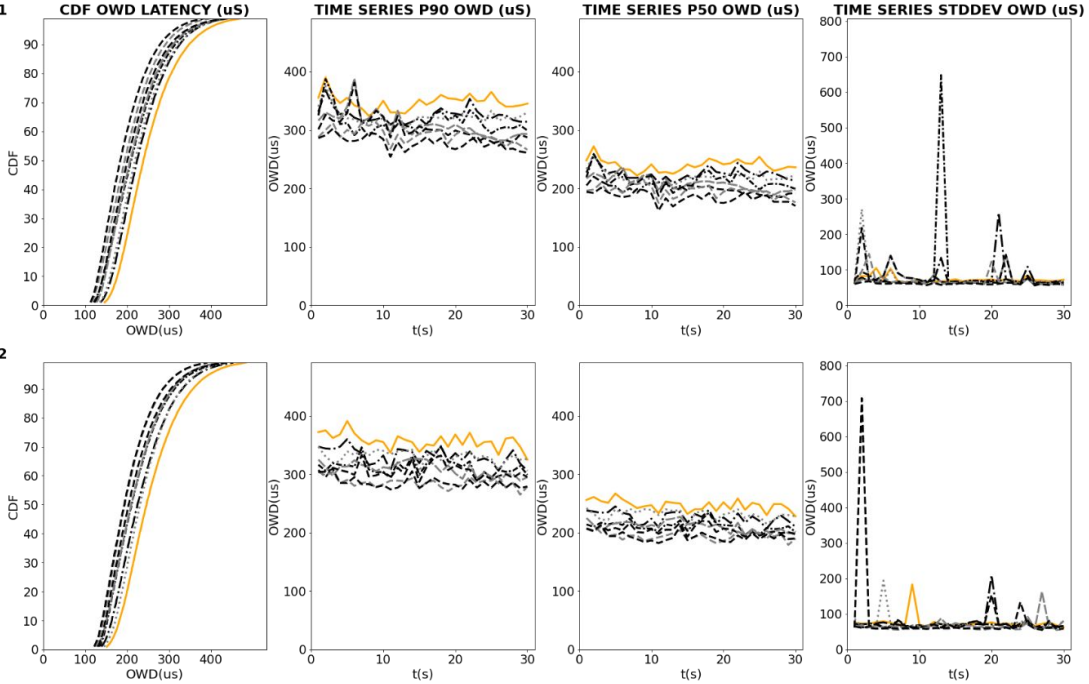
N=15, D=3, F=2, K=4, POOL=24, Rate=10000 p/sec, Duration=30 sec

BEST-heuristic-W_5 RESULTS 1

	99(%)	90(%)	50(%)	MEAN	STD	RX(%)
W_9	472.0	348.0	239.0	253.5	74.31	100.0
W_16	438.0	326.0	224.0	237.7	85.13	100.0
W_11	442.0	323.0	219.0	233.4	84.27	100.0
W_14	435.0	315.0	210.0	223.9	137.3	100.0
W_19	442.0	310.0	208.0	221.6	76.69	100.0
W_13	440.0	304.0	201.0	216.2	85.51	100.0
W_17	399.0	292.0	195.0	206.5	72.45	100.0
W_8	385.0	279.0	184.0	196.7	61.73	100.0

BEST-heuristic-W_5 RESULTS 2

	99(%)	90(%)	50(%)	MEAN	STD	RX(%)
W_9	483.0	358.0	247.0	261.0	78.95	100.0
W_11	454.0	333.0	225.0	239.5	78.93	100.0
W_16	443.0	331.0	231.0	244.2	76.52	100.0
W_19	439.0	314.0	211.0	224.9	68.74	100.0
W_14	420.0	310.0	210.0	222.8	72.15	100.0
W_13	434.0	305.0	204.0	218.5	72.16	100.0
W_17	390.0	286.0	195.0	209.2	143.0	100.0
W_8	392.0	286.0	194.0	207.8	144.8	100.0



- **Multiple runs** perform similarly

Results

EVALUATION COMPARISON - BEST x REBUILDS

N=15, D=3, F=2, K=4, POOL=24, Rate=10000 p/sec, Duration=30 sec

BEST-heuristic-W_5

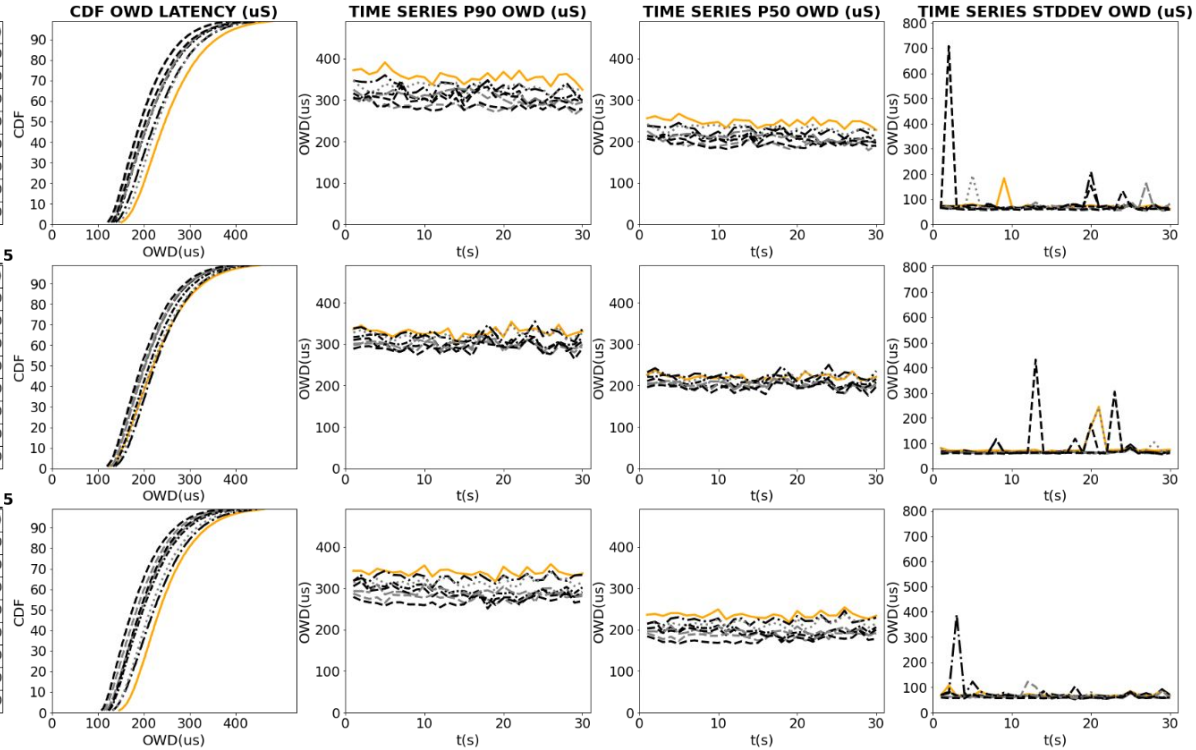
	99(%)	90(%)	50(%)	MEAN	STD	RX(%)
W_9	483.0	358.0	247.0	261.0	78.95	100.0
W_11	454.0	333.0	225.0	239.5	78.93	100.0
W_16	443.0	331.0	231.0	244.2	76.52	100.0
W_19	439.0	314.0	211.0	224.9	68.74	100.0
W_14	420.0	310.0	210.0	222.8	72.15	100.0
W_13	434.0	305.0	204.0	218.5	72.16	100.0
W_17	390.0	286.0	195.0	209.2	143.0	100.0
W_8	392.0	286.0	194.0	207.8	144.8	100.0

BEST-REBUILD-1-heuristic-W_5

	99(%)	90(%)	50(%)	MEAN	STD	RX(%)
W_12	454.0	330.0	221.0	234.4	88.58	100.0
W_16	435.0	324.0	225.0	237.0	68.87	100.0
W_11	442.0	323.0	219.0	232.8	86.16	100.0
W_19	424.0	310.0	212.0	224.5	65.09	100.0
W_13	415.0	299.0	203.0	215.8	63.53	100.0
W_14	407.0	299.0	201.0	214.2	108.5	100.0
W_17	403.0	297.0	201.0	213.3	83.73	100.0
W_8	397.0	290.0	195.0	207.9	82.78	100.0

BEST-REBUILD-2-heuristic-W_5

	99(%)	90(%)	50(%)	MEAN	STD	RX(%)
W_22	462.0	339.0	235.0	248.8	71.55	100.0
W_16	447.0	326.0	223.0	236.1	101.3	100.0
W_11	433.0	316.0	215.0	228.0	67.82	100.0
W_19	410.0	297.0	199.0	211.9	66.22	100.0
W_14	397.0	291.0	195.0	206.7	64.04	100.0
W_13	401.0	287.0	193.0	205.7	63.6	100.0
W_17	381.0	279.0	185.0	197.2	65.66	100.0
W_8	373.0	269.0	176.0	188.6	62.43	100.0



- **REBUILD** on leaf layers provide inconsistent performance gains

Results

EVALUATION COMPARISON - BEST x REBUILDS

N=15, D=3, F=2, K=4, POOL=24, Rate=10000 p/sec, Duration=30 sec

BEST-heuristic-W_3

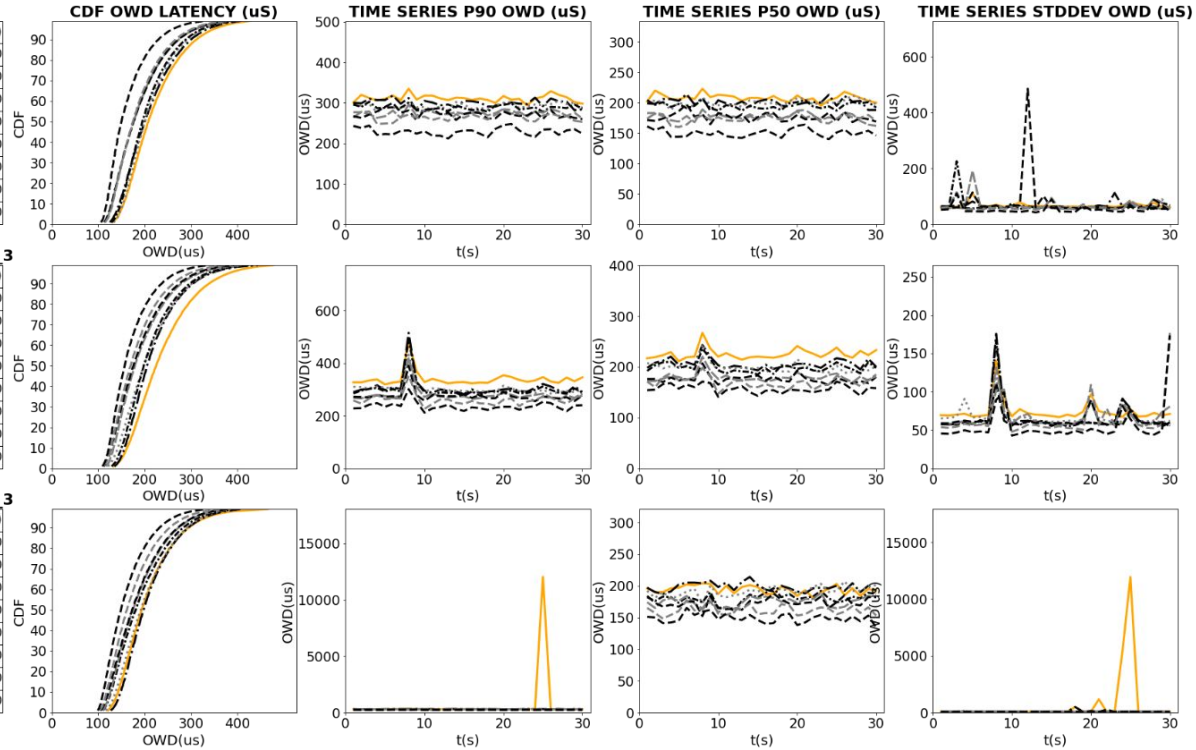
	99(%)	90(%)	50(%)	MEAN	STD	RX(%)
W_9	428.0	312.0	208.0	223.1	68.6	100.0
W_4	407.0	300.0	200.0	213.8	67.13	100.0
W_19	403.0	293.0	200.0	214.1	62.75	100.0
W_1	398.0	289.0	195.0	209.6	72.58	100.0
W_10	385.0	274.0	176.0	191.6	70.17	100.0
W_17	381.0	273.0	176.0	192.2	109.5	100.0
W_8	376.0	267.0	175.0	189.4	64.88	100.0
W_13	328.0	230.0	150.0	164.6	56.76	100.0

BEST-REBUILD-1-heuristic-W_3

	99(%)	90(%)	50(%)	MEAN	STD	RX(%)
W_6	477.0	339.0	225.0	240.6	76.58	100.0
W_19	444.0	304.0	205.0	221.5	70.47	100.0
W_4	424.0	302.0	198.0	213.1	69.83	100.0
W_1	432.0	298.0	198.0	214.5	69.33	100.0
W_10	409.0	279.0	179.0	195.9	68.02	100.0
W_17	393.0	275.0	176.0	191.1	64.59	100.0
W_8	377.0	262.0	170.0	184.9	71.96	100.0
W_13	339.0	237.0	157.0	170.8	62.47	100.0

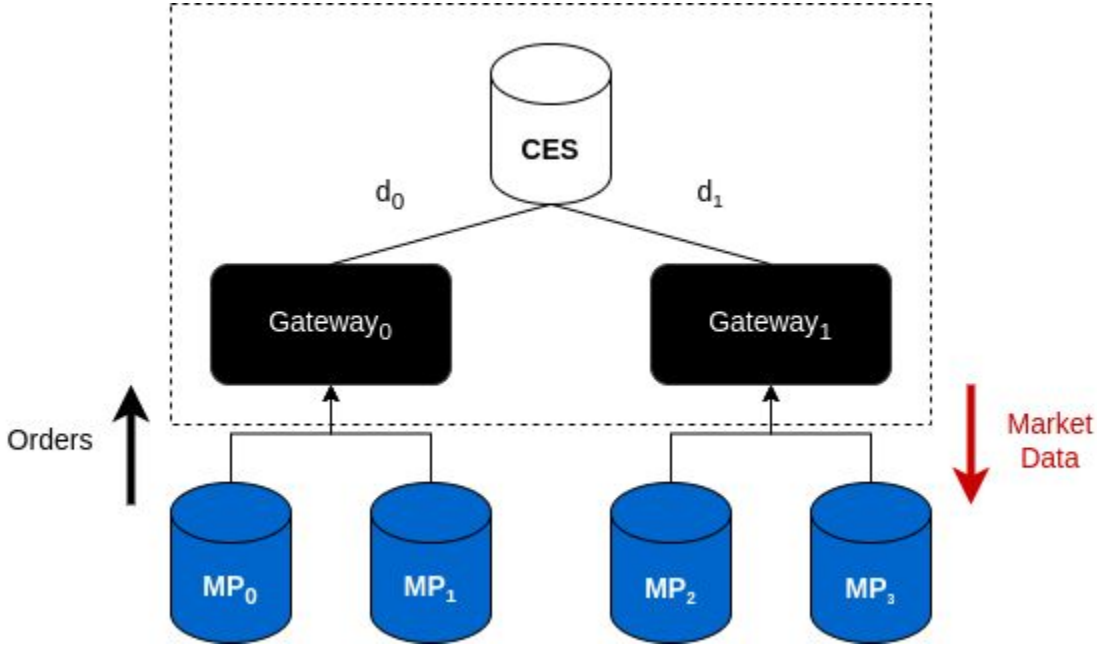
BEST-REBUILD-2-heuristic-W_3

	99(%)	90(%)	50(%)	MEAN	STD	RX(%)
W_1	465.0	295.0	194.0	368.1	2507.	100.0
W_19	406.0	294.0	198.0	213.1	110.1	100.0
W_4	397.0	291.0	192.0	205.2	66.06	100.0
W_2	394.0	283.0	181.0	195.5	69.08	100.0
W_10	385.0	273.0	175.0	189.6	65.98	100.0
W_17	379.0	272.0	173.0	187.5	70.49	100.0
W_8	362.0	254.0	162.0	176.9	75.72	100.0
W_13	331.0	232.0	149.0	163.1	71.2	100.0



- **REBUILD** on leaf layers provide inconsistent performance gains

Figures 1


CES
d_0
d_1
$Gateway_0$
$Gateway_1$
Orders
Market Data
MP_0
MP_1
MP_2
MP_3

Figures 1