



FAKULTÄT FÜR
INFORMATIK

Comparing Systems

How to Compare Systems

Perform the same number of replications for both systems and record the desired result sequences Y_1 and Y_2 , then compute confidence intervals.

Approach 1:

- Compute C.I. for each sequence separately
- If the two C.I. do not overlap, evaluate positions

Y_1	Y_2
1	3
4	5
2	1
...	...

Problem: difficult to interpret combined level of confidence

Approach 2:

- Compute single C.I. for the difference sequence
- If C.I. does not include the origin, evaluate position

$Y_1 - Y_2$
-2
-1
1
...

Correlated Sampling – Why does it help?

Approach 2 is easier to compute and potentially more efficient

- C.I. width depends on data variance
- So, for A2 it depends on the variance of the *differences* :

$$\text{var}(Y_1 - Y_2) = \frac{\sigma_1^2}{R} + \frac{\sigma_2^2}{R} - \frac{2\rho_{12}\sigma_1\sigma_2}{R}$$

Variance of Y_1

Variance of Y_2

Correlation coefficient between Y_1 and Y_2

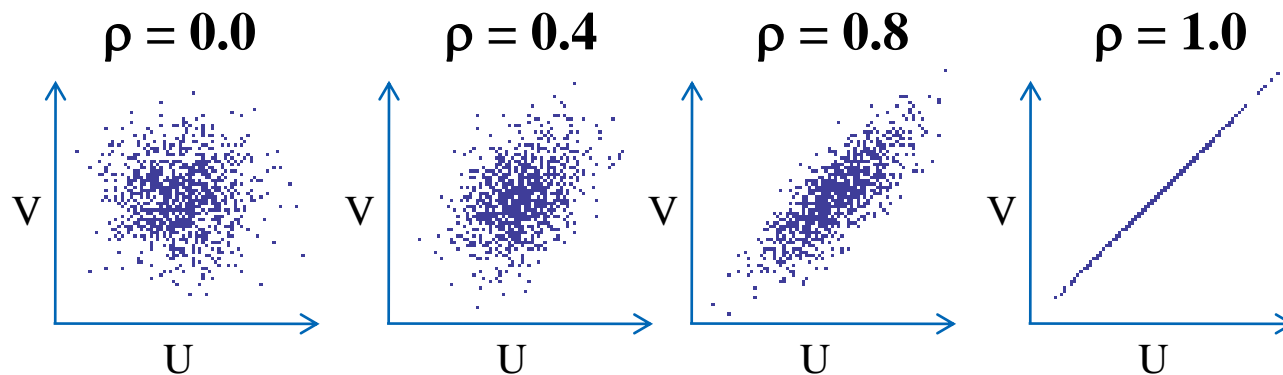
→ If results between both models correlate, variance and width of difference C.I. get smaller

Consequence: Fewer replications for desired level of accuracy

Review: Correlation

The correlation coefficient ρ describes the statistical relationship between two data sequences U and V

- $\rho > 0$ for a positive linear relationship between U and V



How do we create a positive correlation?

- Ensure that the models to be compared behave similarly in the same replication
- Do **NOT** manipulate the models or re-order the results!

Y_1	Y_2
100	105
345	314
57	64
...	...

Correlated Sampling – Achieving a Positive Correlation

Idea:

- Use the same random numbers in both models

More precisely:

- For each model, each replication uses different random numbers
- In each replication, both models use the same RNs

Then,

- Different replications will be independent (Required for C.I.)
- Every pair of results Y_{r1} , Y_{r2} will be positively correlated

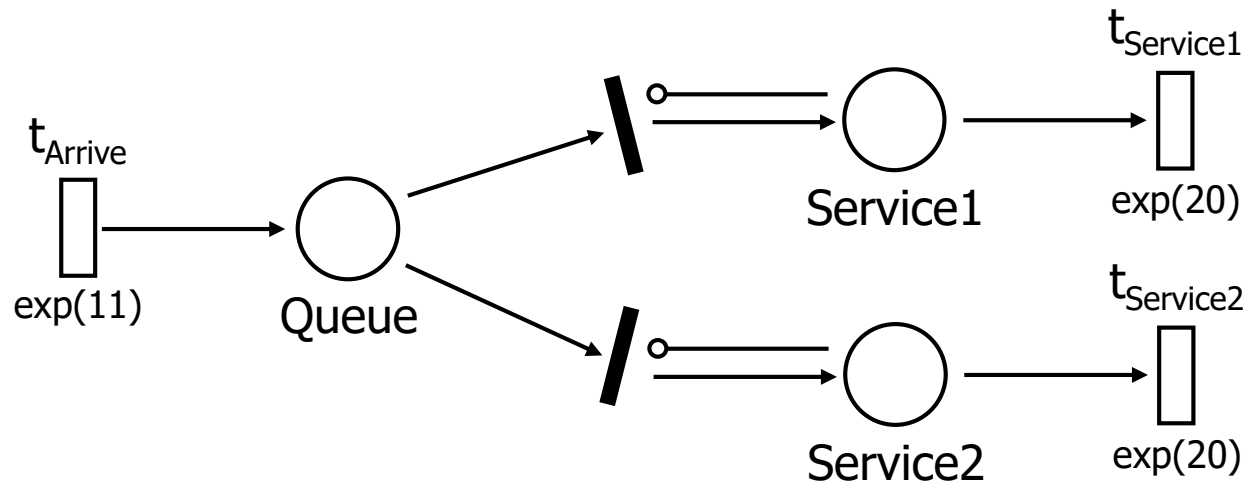
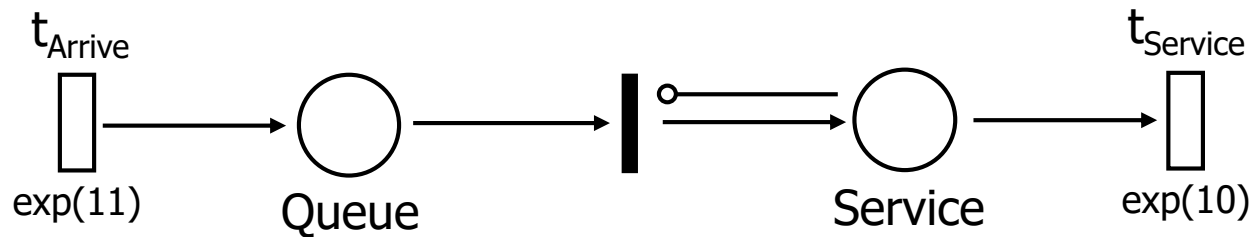
Correlated Sampling

Algorithm for comparing two systems:

- Compute $D_r = Y_{r1} - Y_{r2}$
- Compute $D = \frac{1}{R} \sum D_r$
- Compute $S^2 = \frac{1}{R-1} \sum (D_r - D)^2$
- Compute $\sigma = \frac{S}{\sqrt{R}}$
- Choose α and compute the confidence interval

Example

Example: Compare one- and two-server models



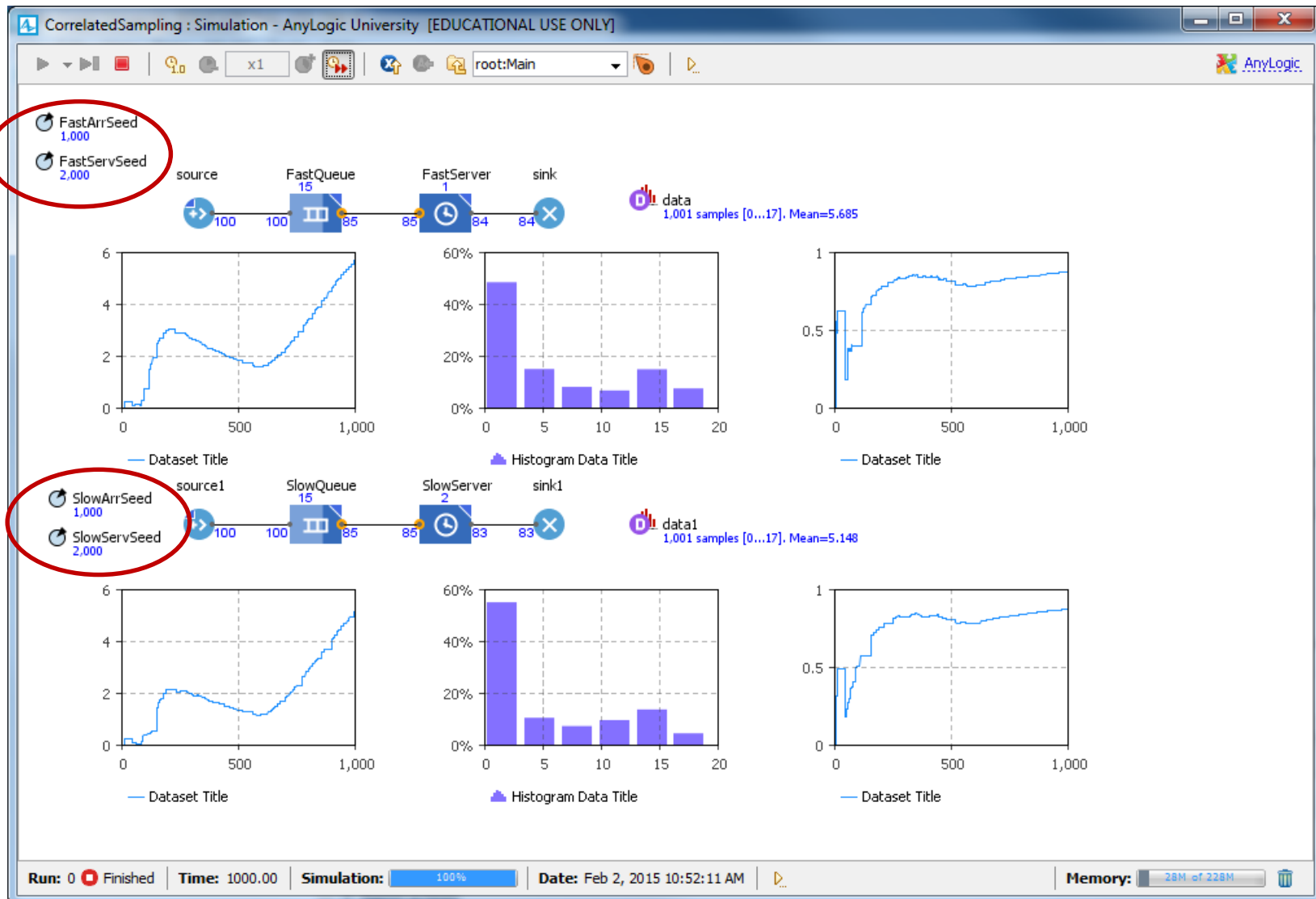
Example

Do a simulation study to make the comparison

We will do the following experiment:

- Compute average queue length at $T = 10,000$
- Use $R = 15$ replications

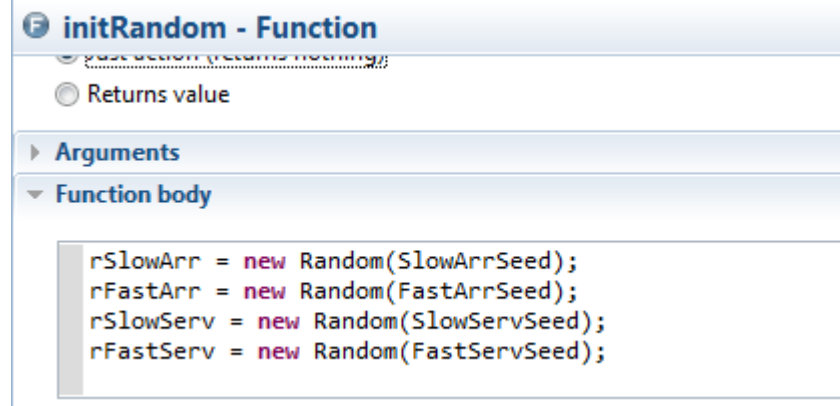
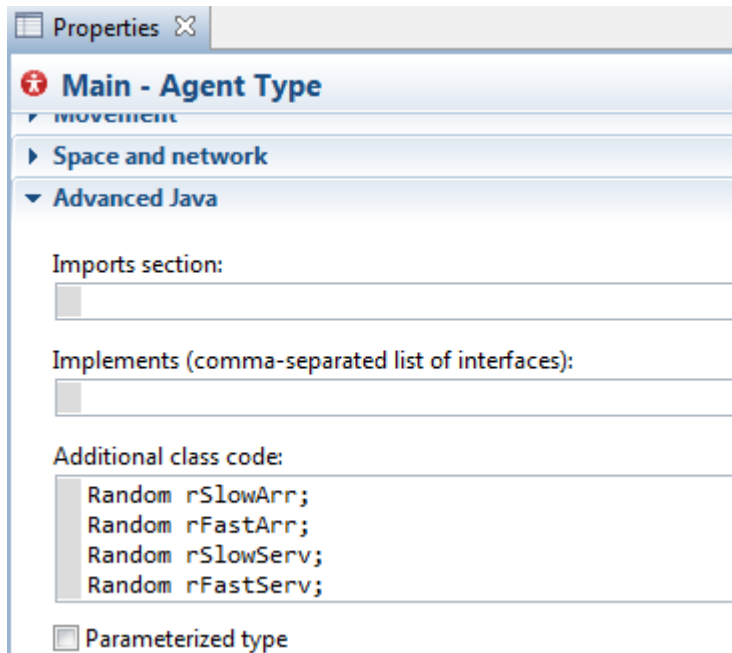
AnyLogic Model Using Correlated Sampling



Controlling Random Numbers in AnyLogic

Declare four separate random number streams

Initialize the random number streams in a function



Controlling Random Numbers in AnyLogic

Assign specific random number stream to each random variable

source - Source

Name: ☒ Show name ☐ Ignore

Arrivals defined by:

Interarrival time:

FastServer - Delay

Name: ☒ Show name ☐ Ignore

Type: ☒ Specified time
☐ Until stopDelay() is called

Delay time:

Capacity:

SlowServer - Delay

Name: ☒ Show name ☐ Ignore

Type: ☒ Specified time
☐ Until stopDelay() is called

Delay time:

Capacity:

Experiment Setup

CorrelatedSampling : ParametersVariation - AnyLogic University [EDUCATIONAL USE ONLY]

Run

Iteration: 1

Parameters

SlowArrSeed	0
SlowServSeed	0
FastArrSeed	0
FastServSeed	0

ArrFastQL {0}

CumFastQL 0

AvgFastQL 0

SumDiffSqFastQL 0

SsqFastQL 0

SigmaFastQL 0

HwidthFastQL 0

ArrSlowQL {0}

CumSlowQL 0

AvgSlowQL 0

SumDiffSqSlowQL 0

SsqSlowQL 0

SigmaSlowQL 0

HwidthSlowQL 0

ArrDiff {0}

CumDiff 0

AvgDiff 0

SumDiffSqDiff 0

SsqDiff 0

SigmaDiff 0

HwidthDiff 0

Queue Fast Server: [0,0]

Queue Slow Servers: [0,0]

Difference: [0,0]

i 1

CorrelateArrival False

CorrelateService False

correlation 0

☐ Correlate Arrivals

☐ Correlate Service Times

Run: 0 Idle Experiment: 0% Simulations: Stop time not set Memory: 20M of 228M 0.0 sec

Controlling Random Numbers in AnyLogic

Set seeds and initialize the random number streams from the experiment

- Use the same seeds for correlated runs
- Use different seeds for independent runs

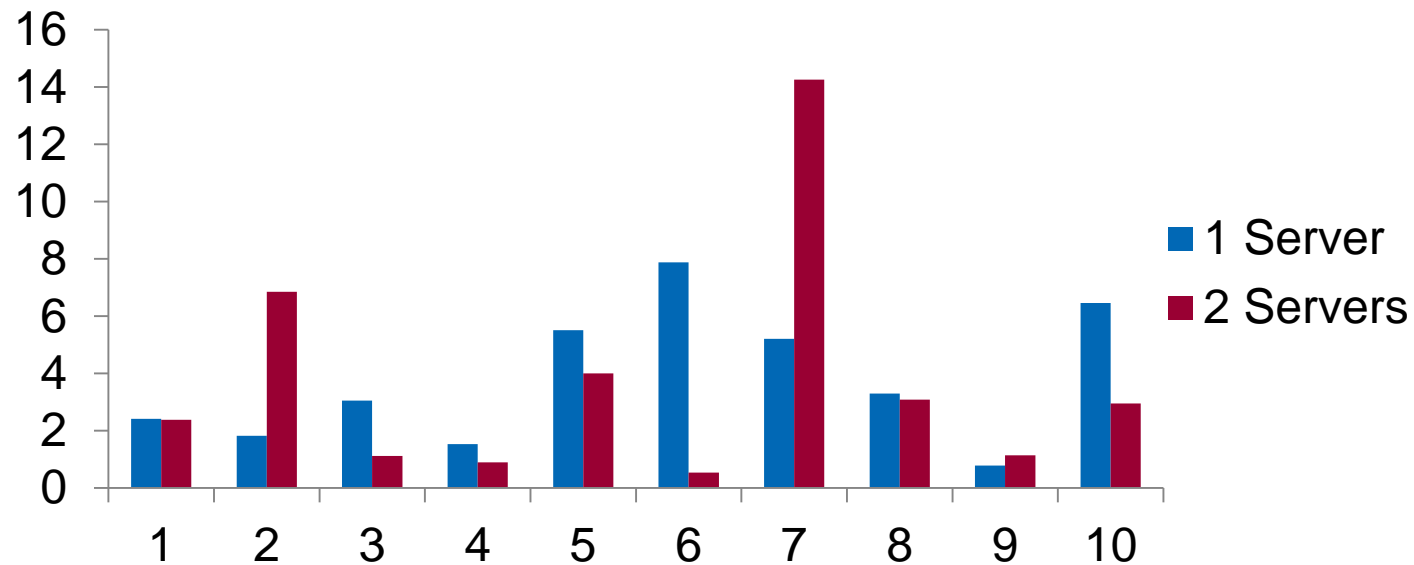
Before simulation run:

```
i+=1;  
root.SlowArrSeed = i;  
if(CorrelateArrival)    root.FastArrSeed = i;  
else                   root.FastArrSeed = i+10;  
root.SlowServSeed = i+20;  
if(CorrelateService) root.FastServSeed = i+20;  
else                   root.FastServSeed = i+30;  
root.initRandom();
```

Experimental Results

Individual replication results, unsynchronised

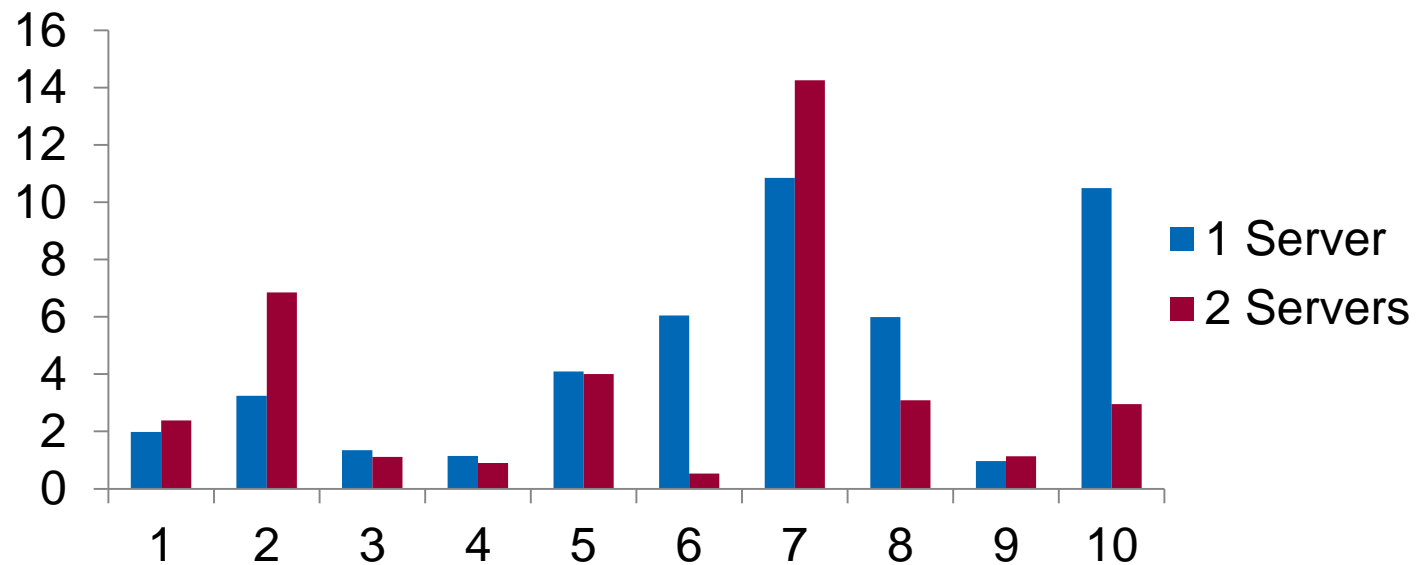
Unsynchronised



Experimental Results

Individual replication results, only arrivals synchronised

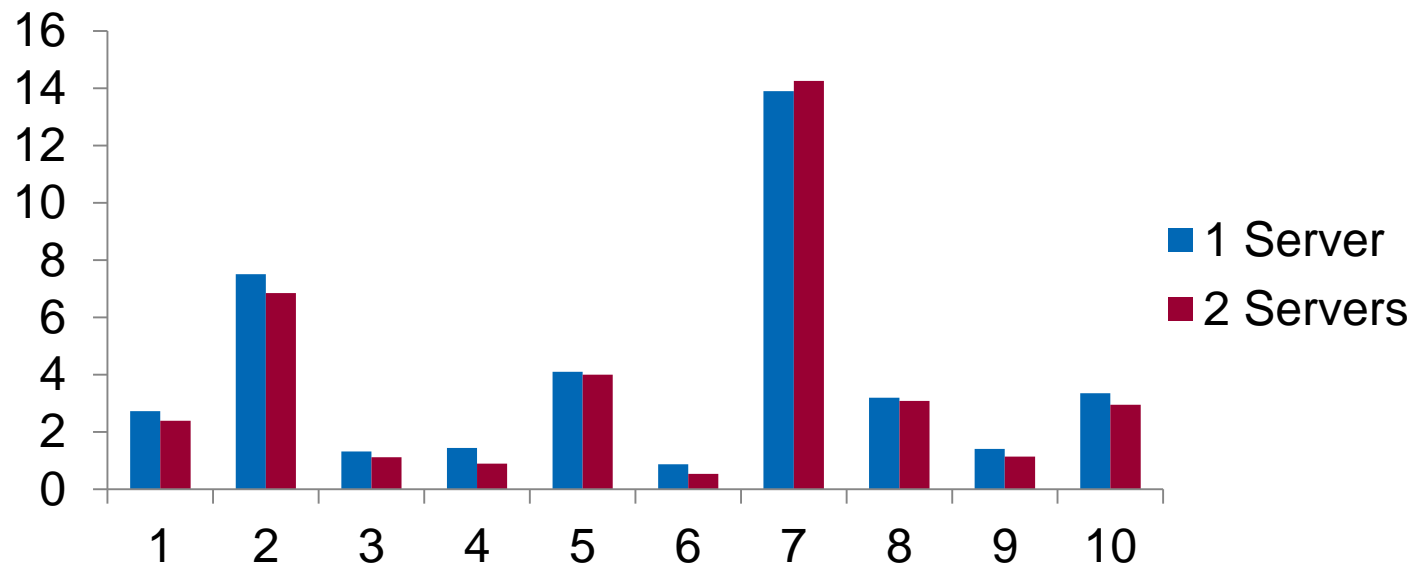
Arrivals Synchronised



Experimental Results

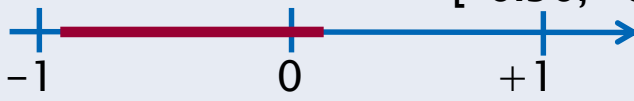
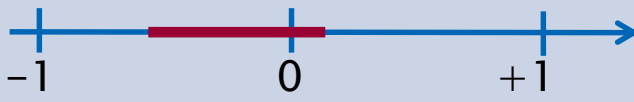

Individual replication results, fully synchronised

Fully Synchronised



Results

For 100 Replications:

RN	Sample		Width	C.I. ($\alpha = 0.05$)
	Corr.	Variance		Interval
Independent	0.02	6.46	1.01	 [-0.90, 0.11]
Arrivals Synchronized	0.54	2.75	0.66	 [-0.54, 0.12]
Fully Synchronized	0.99	0.05	0.09	 [-0.35, -0.26]