On Choosing a Task Assignment Policy for a Distributed Server System

21.05.2012

ferhat elmas
214805

Outline

- Introduction
- Motivation
- Assumptions
- Performance Metrics
- System Parameters
- Simulation
- Results
- Confidence Intervals
- Conclusion
- Q&A

Introduction

- A task assignment policy for a distributed server system that composed of multiple hosts which process tasks in FCFS.

1. **Random**: One of the hosts is randomly chosen. Each of host has an equal probability.
2. **Round Robin**: Tasks are assigned to hosts in cyclic fashion.
3. **Dynamic**: The task is assigned to the host that has minimum expected waiting time.
4. **Size Based**: Task size is partitioned and tasks that are in a certain range are assigned to a particular host.

Motivation

- Assignment policies are studied extensively
  - Modeled as exponential, poor model
- Real task values show heavy tail property
  - Very small fraction of task makes nearly half of the load
  - Pr{X > x} ~ x^-a where 0 <= a <= 2
  - The lower ***a***, the more variable the distribution

Task Description	a
Unix CPU requirements at BellCore	[1, 1.25]
Unix CPU requirements at Berkeley	~ [1]
Size of files transferred through Web	[1.1, 1.3]
Size of FTP transfers in the internet	[0.9, 1.1]

Assumptions

- Task Size is known in advance
- Service time is equal to size of task, no scheduling overhead
- CPU is the only resource
- Hosts in the server are identical
- Task is assigned to a host as soon as it arrives.
- Execution is in FCFS and non-preemptive.

System Model

- Task arrival rate is poisson.
- Task size is bounded pareto distribution.
- Defined as *B(k, p, a)*
  - f (x) = a * k^a * x^(-a-1) / 1 - (k/p)^a
  - k and p are smallest and largest task sizes
  - a controls the variance and smaller a means higher variance
- a varies from 1 to 2

Performance Measures

- Average Waiting Time
- Average SlowDown
- Throughput
- Server Utilization

System Parameters

- Number of Servers
- Load (directly affects arrival rate)
- *a,* task size variance
  - while a is changing, k is adjusted to keep the mean same

Schedule

- Random:
  - Task t is assigned to host i with 1/number_of_hosts probability
- Round Robin:
  - Task t is assigned to host i,
  - if the_number_of_t mod number_of_host == i
- Dynamic:
  - Task t is assigned to host i,
  - if min(host_expected_waiting_time) == i
- Size Based:
  - Task t is assigned to host i,
  - size_of_t is in [host_i_lower, host_i_higher)

Task Ranges

- Ranges are defined such that expected task size within range is equal in each range
- Each range has a mean of overall mean is divided by number of hosts
- Cut off values
  - $k = x_0 < x_1 < x_2 < x_3 < .... < x_{h-1} < x_h = p$
  - if size is in range of $[x_{i-1}, x_i)$, host i is chosen

Simulation

- To study variance task size variability
  - a varies from 1.1 to 1.9 with step 0.1
- p is 100000, mean is 300
  - k is adjusted to keep mean at 300 all time
- Load is fixed at 0.8

Simulation - 2

- To study the number of hosts
  - varies between 1 and 10

- To study arrival rate
  - utilization(load) is changed
  - rate = numberOfServers * load / meanTaskSize
  - varies between 0.1 and 0.9

Simulation - 3

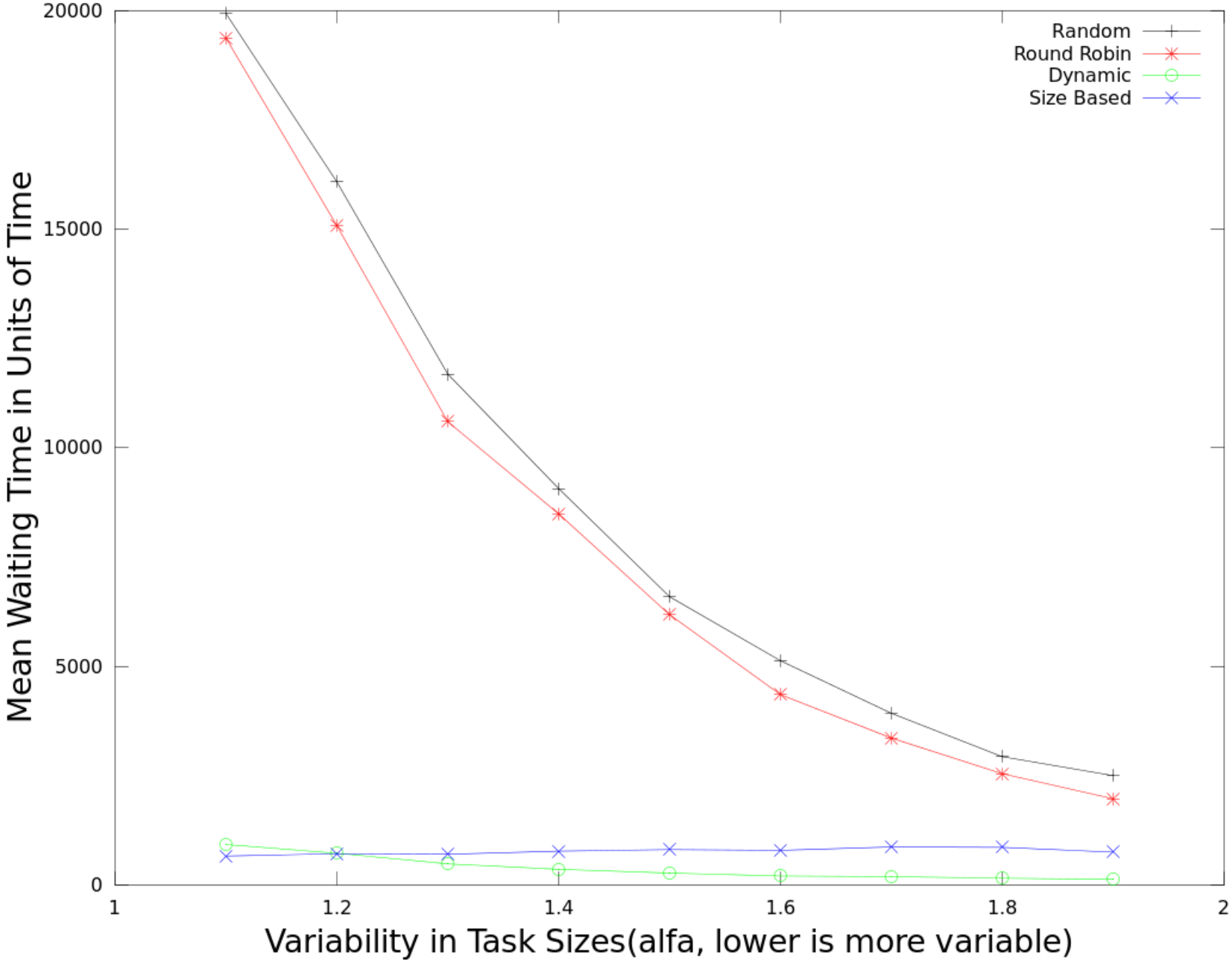
- Number of arrival tasks = 100000
- Number of runs = 100
- Steady state after 20000 task arrival
  - important, sufficiently large number of heavy tasks are needed otherwise statistics are opportunistic for policies

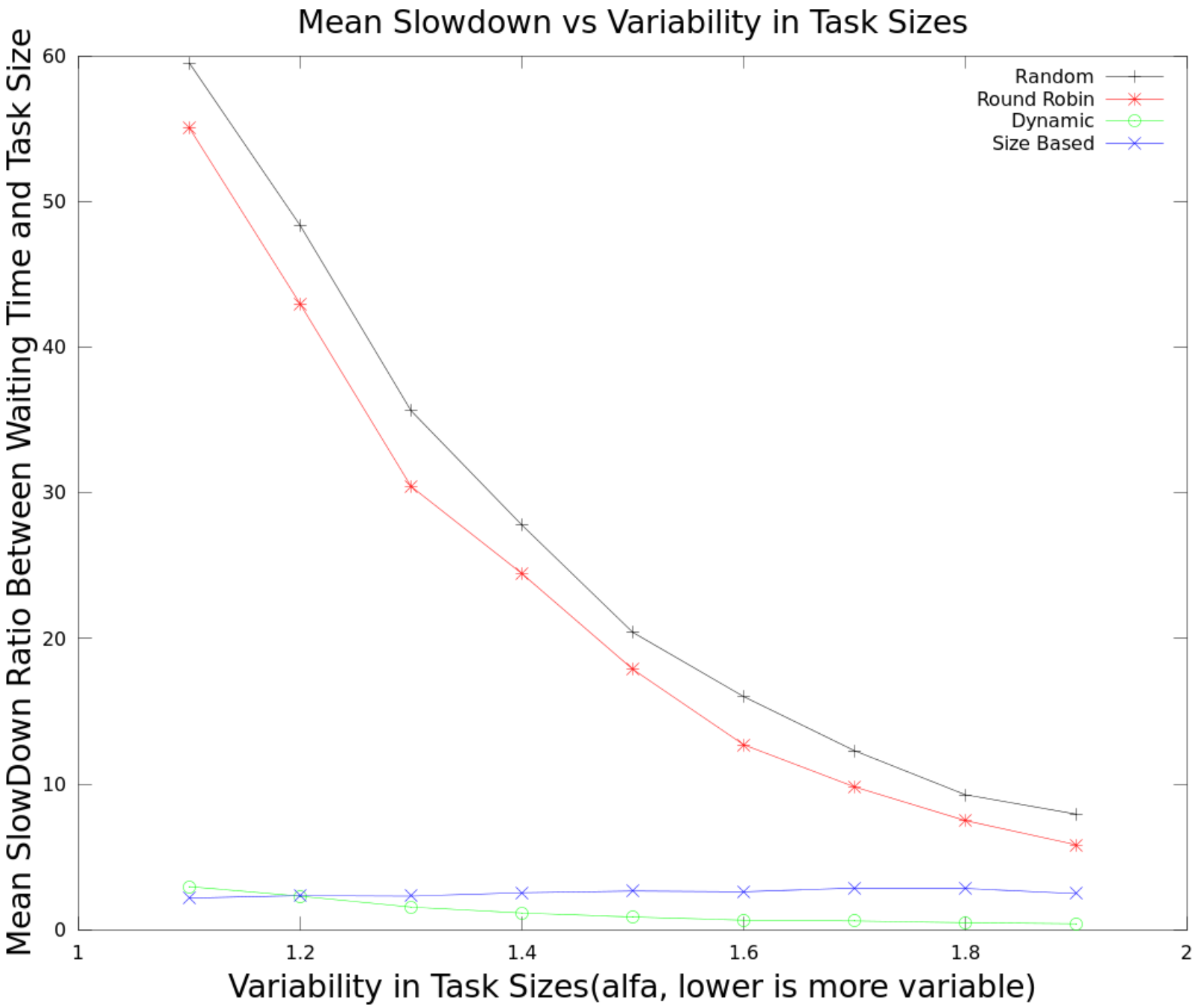
Results

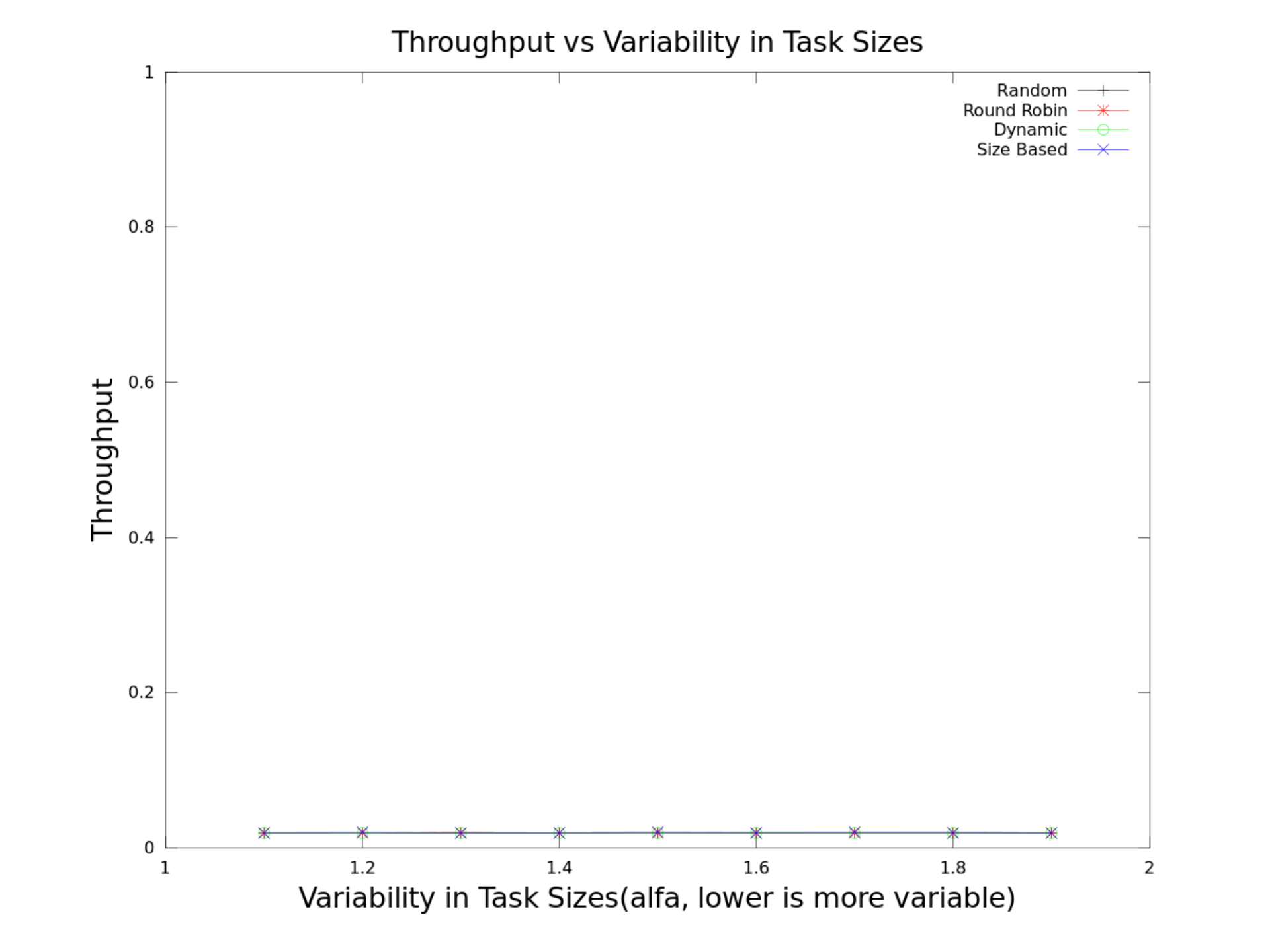
- Variability
- Number Of Servers
- Load

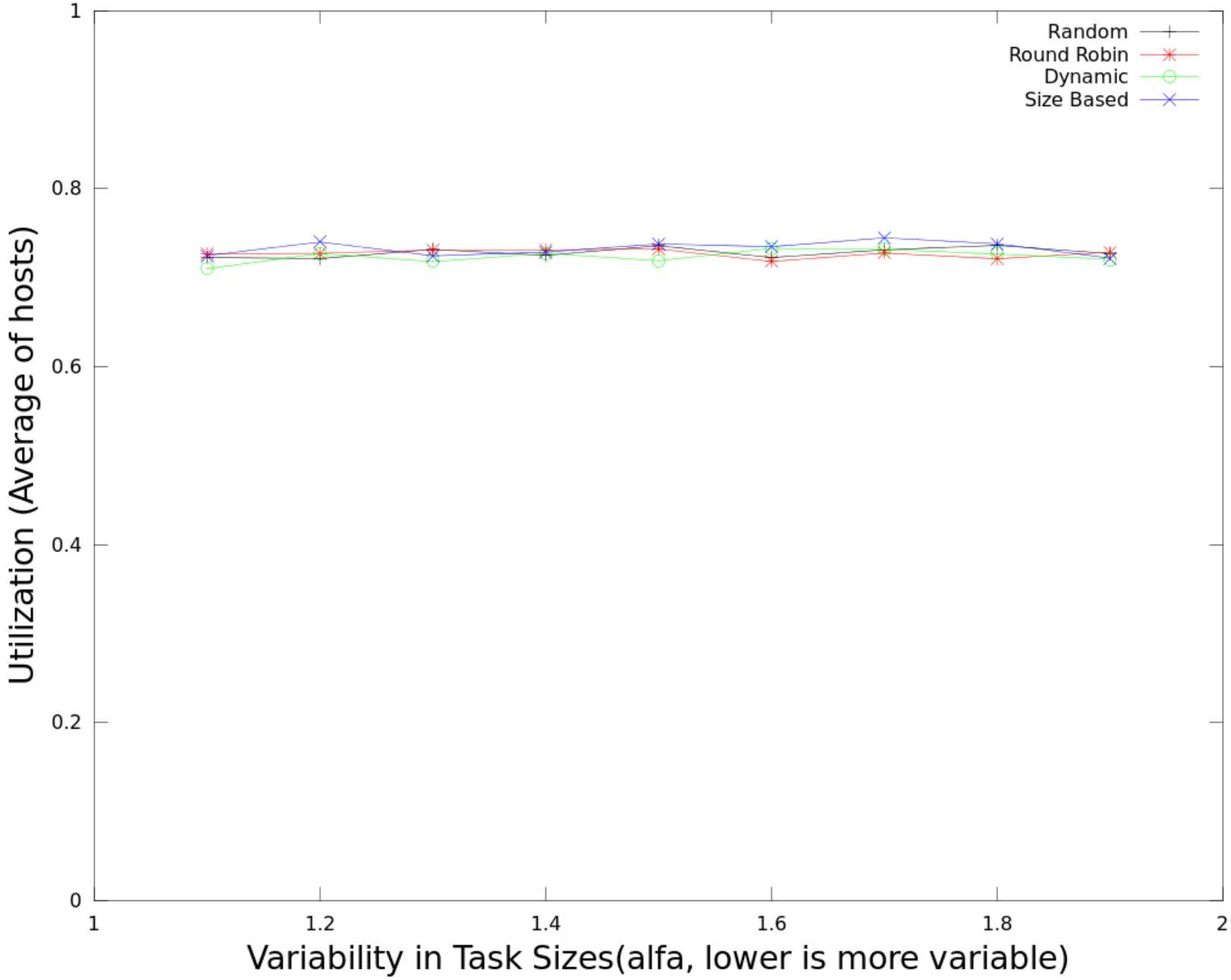
Variability

- a is changed
  - [1.1, 1.9]
- load is fixed at 0.8


Mean Waiting Time vs Variability in Task Sizes
Mean Waiting Time in Units of Time
Variability in Task Sizes(alfa, lower is more variable)
Random
Round Robin
Dynamic
Size Based
20000
15000
10000
5000
0
1
1.2
1.4
1.6
1.8
2

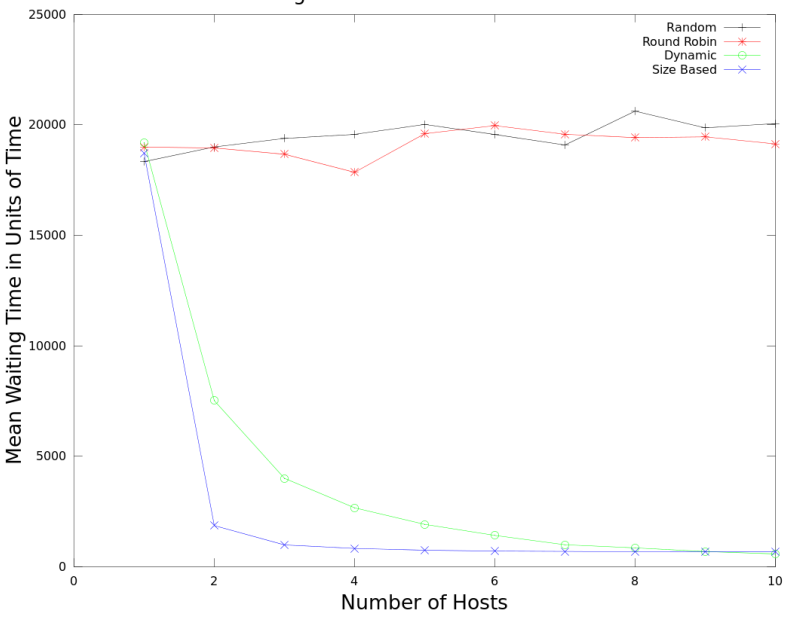

Mean Slowdown vs Variability in Task Sizes
Mean SlowDown Ratio Between Waiting Time and Task Size
Variability in Task Sizes(alfa, lower is more variable)
Random
Round Robin
Dynamic
Size Based
0
10
20
30
40
50
60
1
1.2
1.4
1.6
1.8
2

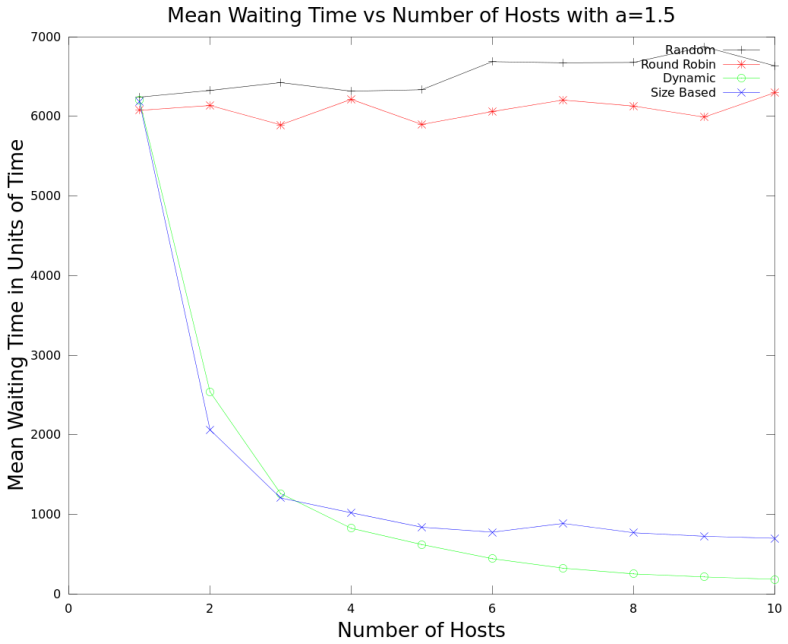

Throughput vs Variability in Task Sizes
Random
Round Robin
Dynamic
Size Based
Throughput
1
0.8
0.6
0.4
0.2
0
1
1.2
1.4
1.6
1.8
2
Variability in Task Sizes(alfa, lower is more variable)

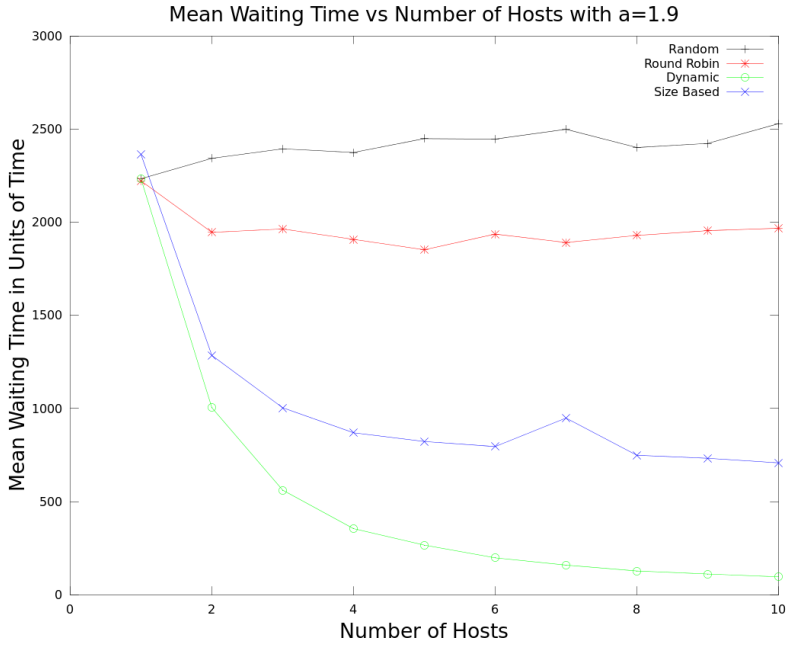

Utilization vs Variability in Task Sizes
Random
Round Robin
Dynamic
Size Based
Utilization (Average of hosts)
0
0.2
0.4
0.6
0.8
1
1
1.2
1.4
1.6
1.8
2
Variability in Task Sizes(alfa, lower is more variable)

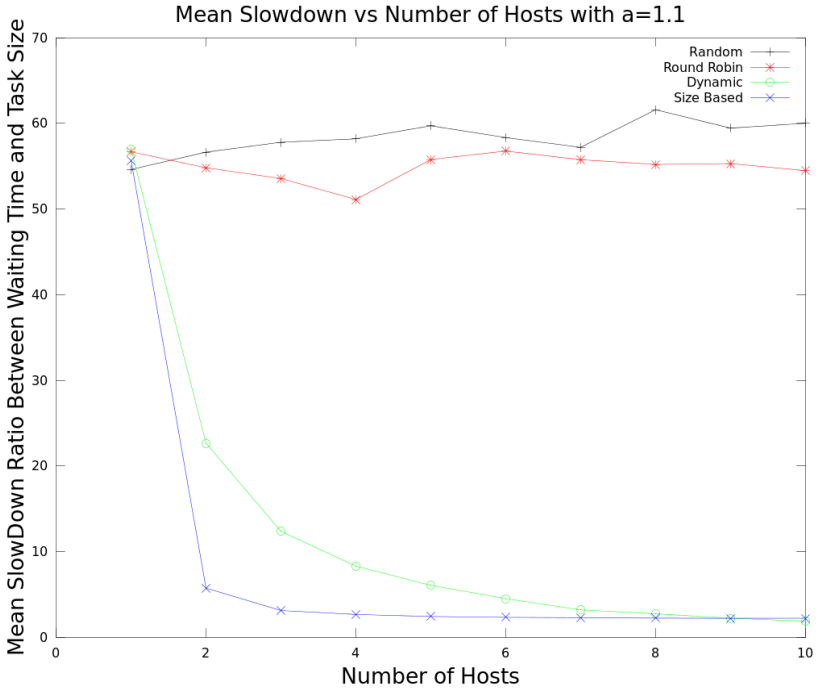
Number Of Hosts

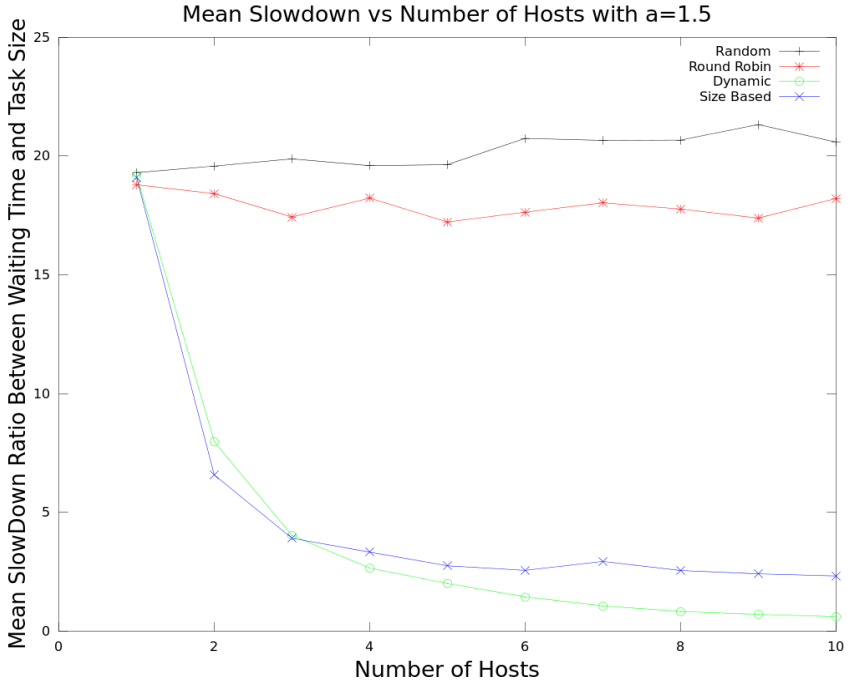
- Number of hosts changes
  - [1, 10]
- Load is fixed at 0.8
- a has 3 different values,
  - 1.1, 1.5, 1.9

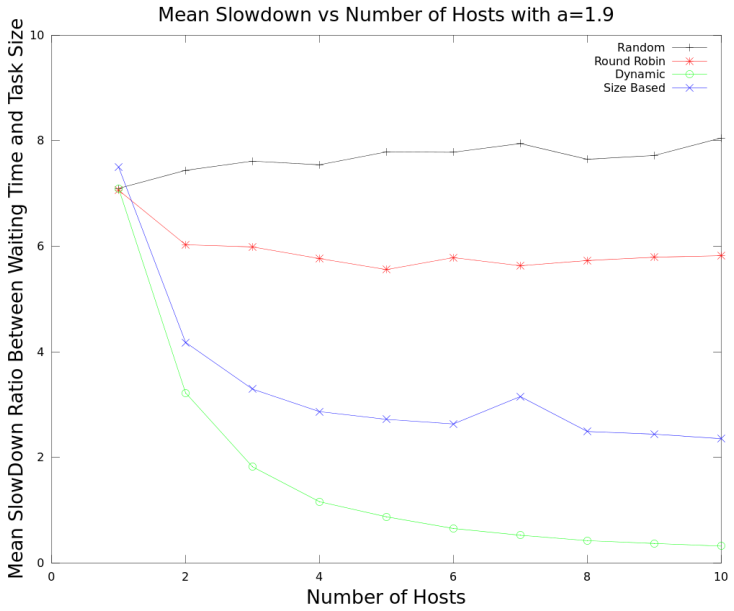

Mean Waiting Time vs Number of Hosts with a=1.1
Mean Waiting Time in Units of Time
Number of Hosts
Random
Round Robin
Dynamic
Size Based
0
5000
10000
15000
20000
25000
0
2
4
6
8
10

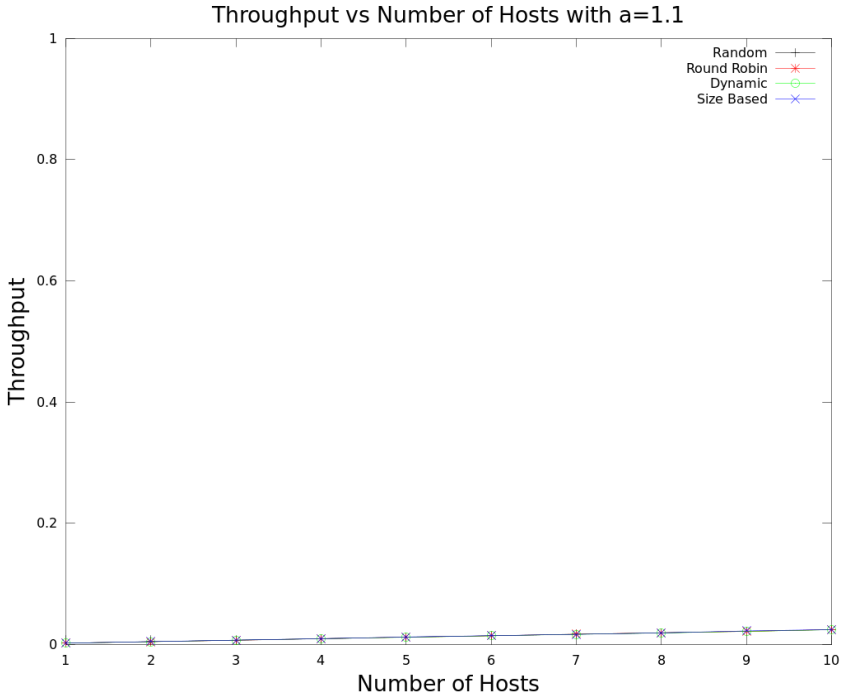

Mean Waiting Time vs Number of Hosts with a=1.5
Mean Waiting Time in Units of Time
Number of Hosts
Random
Round Robin
Dynamic
Size Based
0
1000
2000
3000
4000
5000
6000
7000
0
2
4
6
8
10

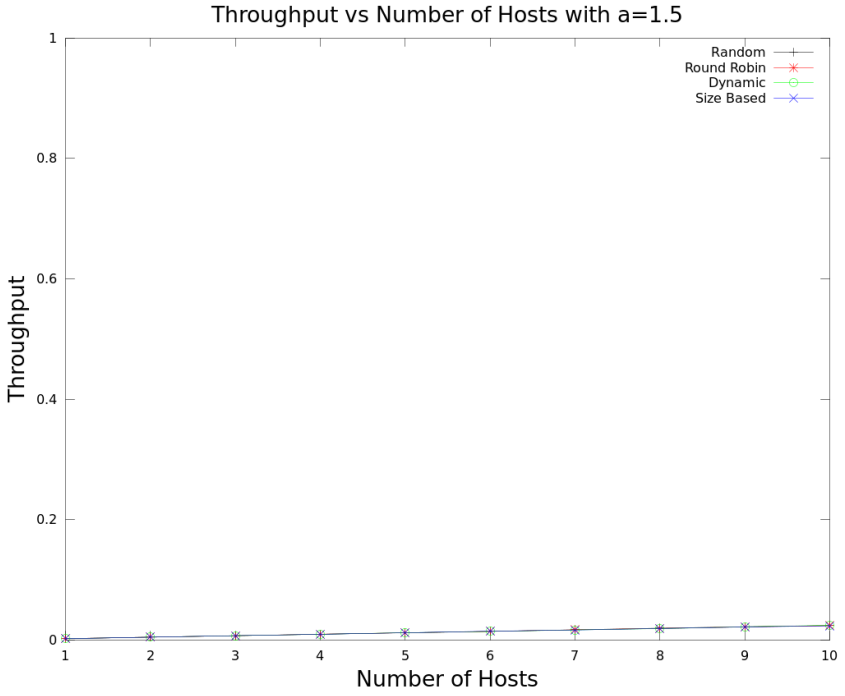

Mean Waiting Time vs Number of Hosts with a=1.9
Mean Waiting Time in Units of Time
Number of Hosts
Random
Round Robin
Dynamic
Size Based
0
500
1000
1500
2000
2500
3000
0
2
4
6
8
10

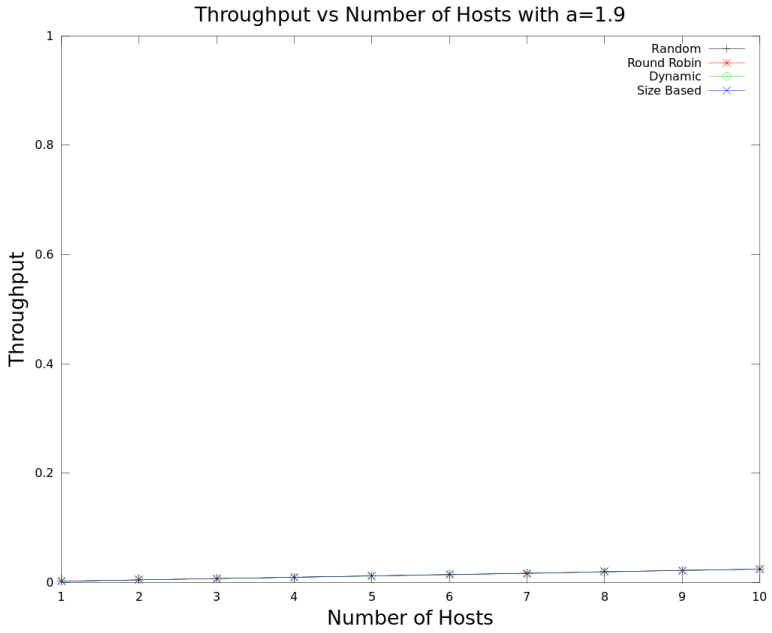

Mean Slowdown vs Number of Hosts with a=1.1
Mean SlowDown Ratio Between Waiting Time and Task Size
Number of Hosts
Random
Round Robin
Dynamic
Size Based
0
10
20
30
40
50
60
70
0
2
4
6
8
10

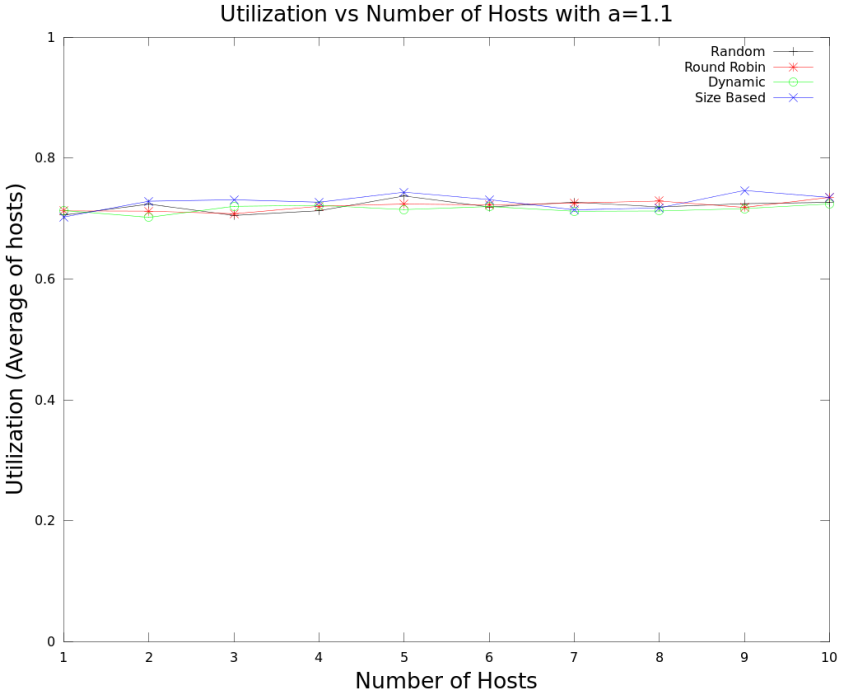

Mean Slowdown vs Number of Hosts with a=1.5
Mean SlowDown Ratio Between Waiting Time and Task Size
Number of Hosts
Random
Round Robin
Dynamic
Size Based
0
5
10
15
20
25
0
2
4
6
8
10

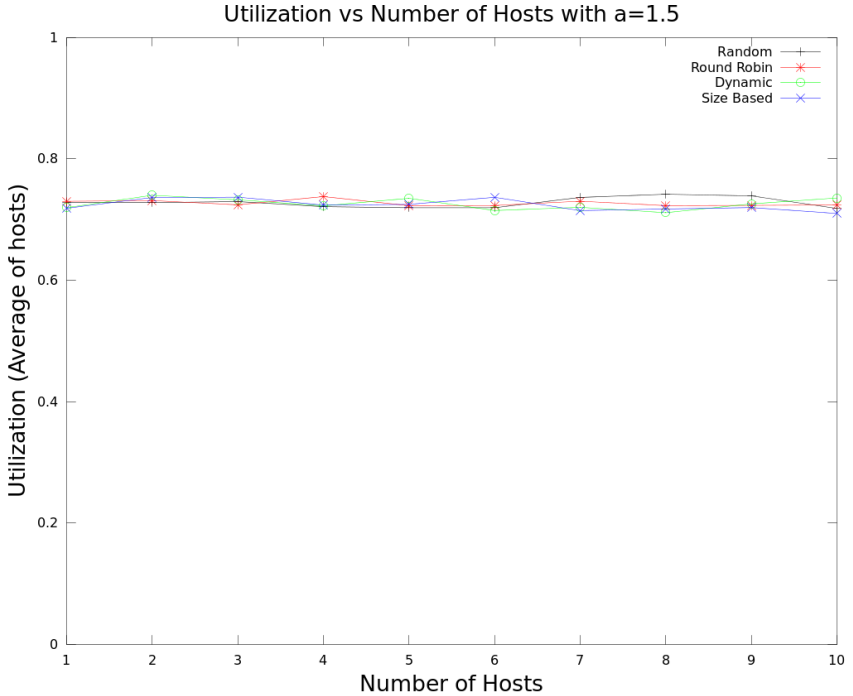

Mean Slowdown vs Number of Hosts with a=1.9
Mean SlowDown Ratio Between Waiting Time and Task Size
Number of Hosts
Random
Round Robin
Dynamic
Size Based
0
2
4
6
8
10
0
2
4
6
8
10

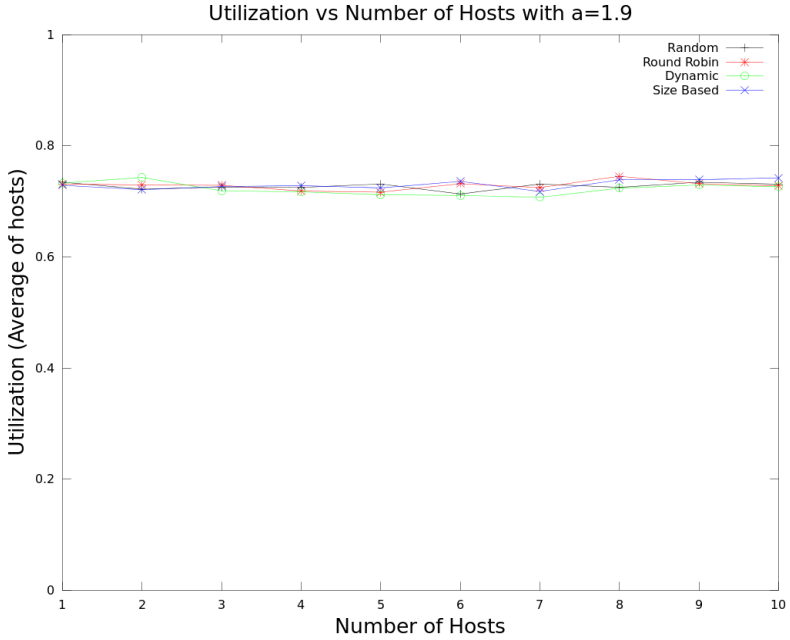

Throughput vs Number of Hosts with a=1.1
Random
Round Robin
Dynamic
Size Based
Throughput
Number of Hosts
0
0.2
0.4
0.6
0.8
1
1
2
3
4
5
6
7
8
9
10


Throughput vs Number of Hosts with a=1.5
Random
Round Robin
Dynamic
Size Based
Throughput
Number of Hosts
0
0.2
0.4
0.6
0.8
1
1
2
3
4
5
6
7
8
9
10


Throughput vs Number of Hosts with a=1.9
Random
Round Robin
Dynamic
Size Based
Throughput
Number of Hosts
0
0.2
0.4
0.6
0.8
1
1
2
3
4
5
6
7
8
9
10

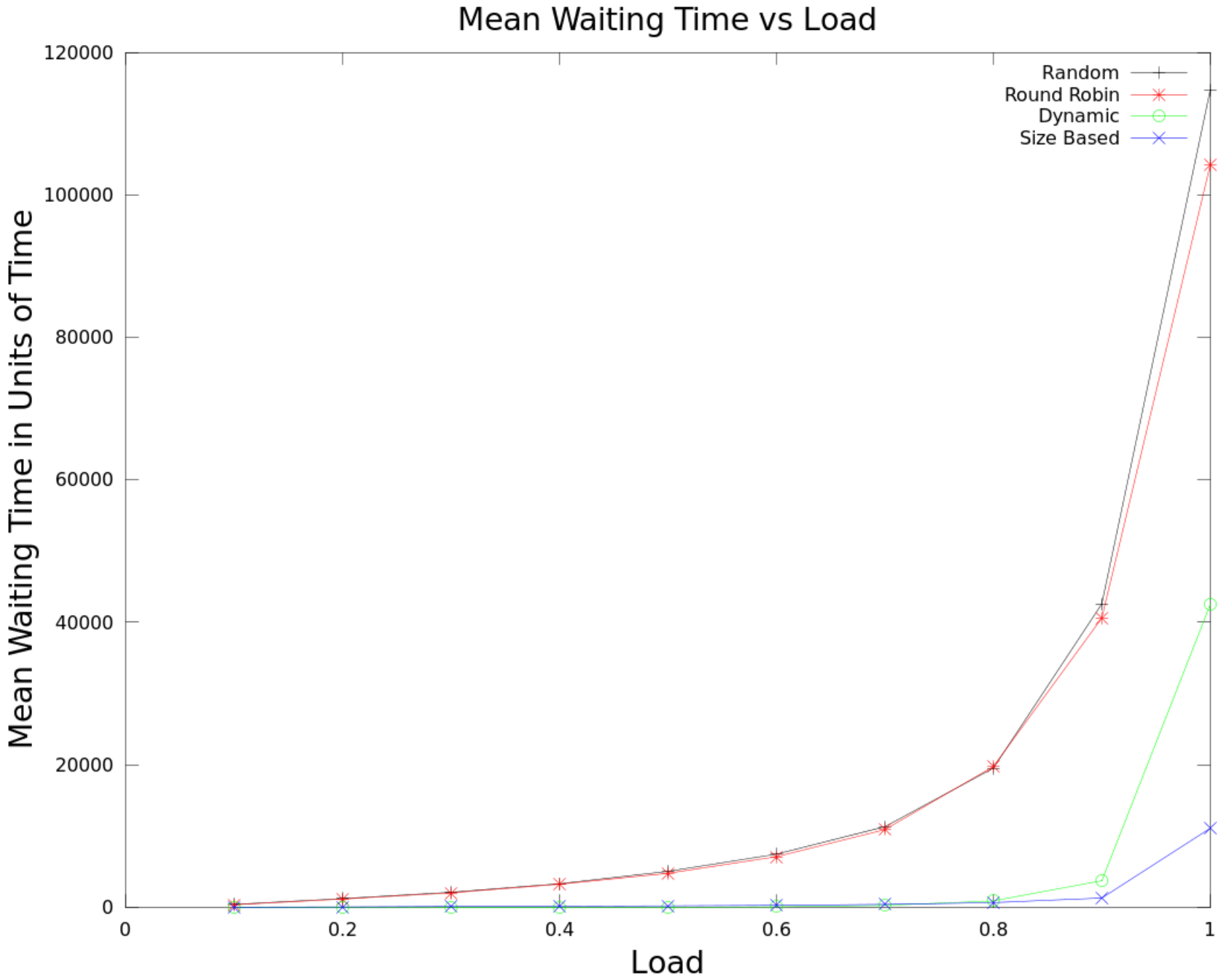

Utilization vs Number of Hosts with a=1.1
Random
Round Robin
Dynamic
Size Based
Utilization (Average of hosts)
Number of Hosts
0
0.2
0.4
0.6
0.8
1
1
2
3
4
5
6
7
8
9
10

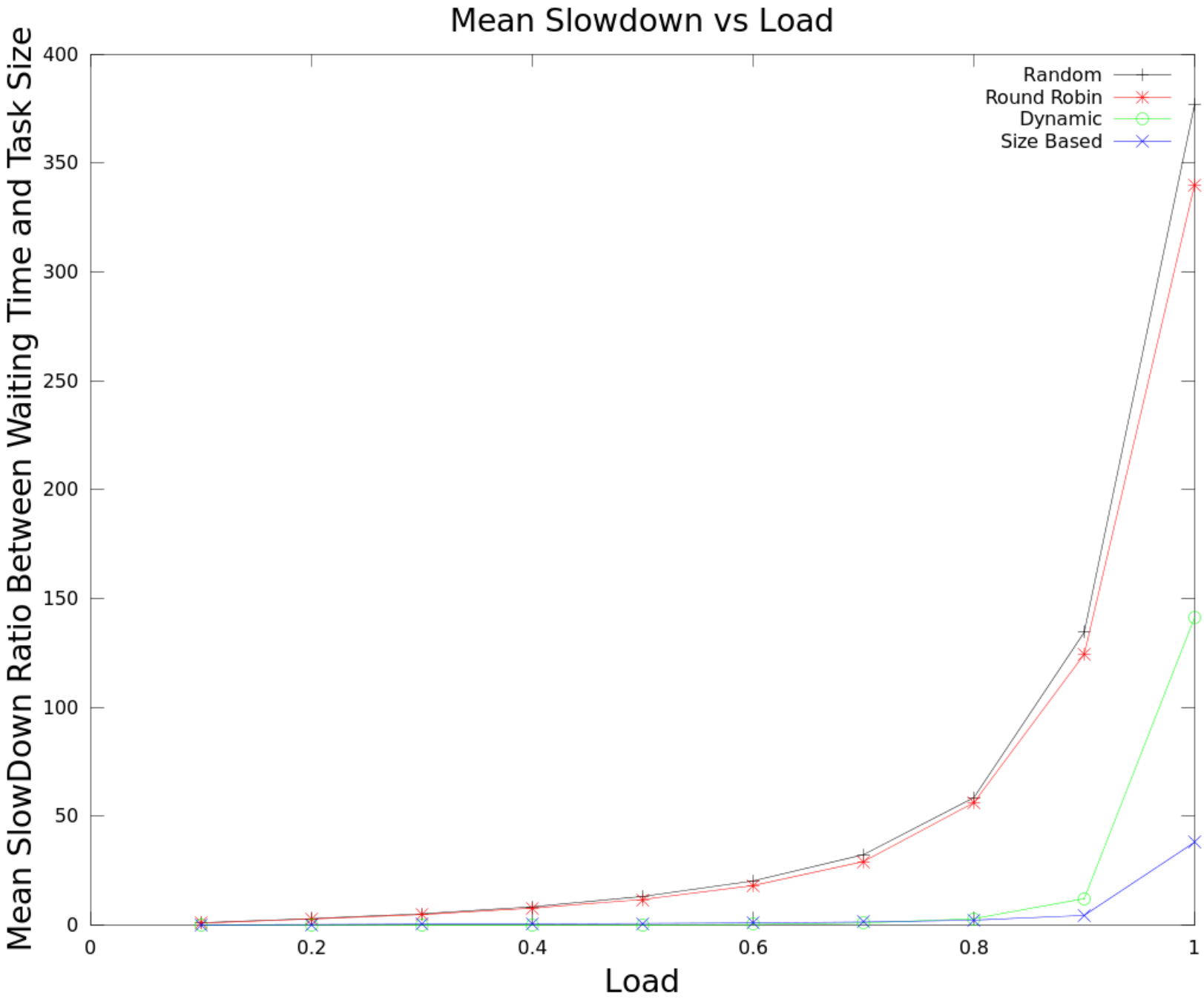

Utilization vs Number of Hosts with a=1.5
Random
Round Robin
Dynamic
Size Based
Utilization (Average of hosts)
Number of Hosts
0
0.2
0.4
0.6
0.8
1
1
2
3
4
5
6
7
8
9
10

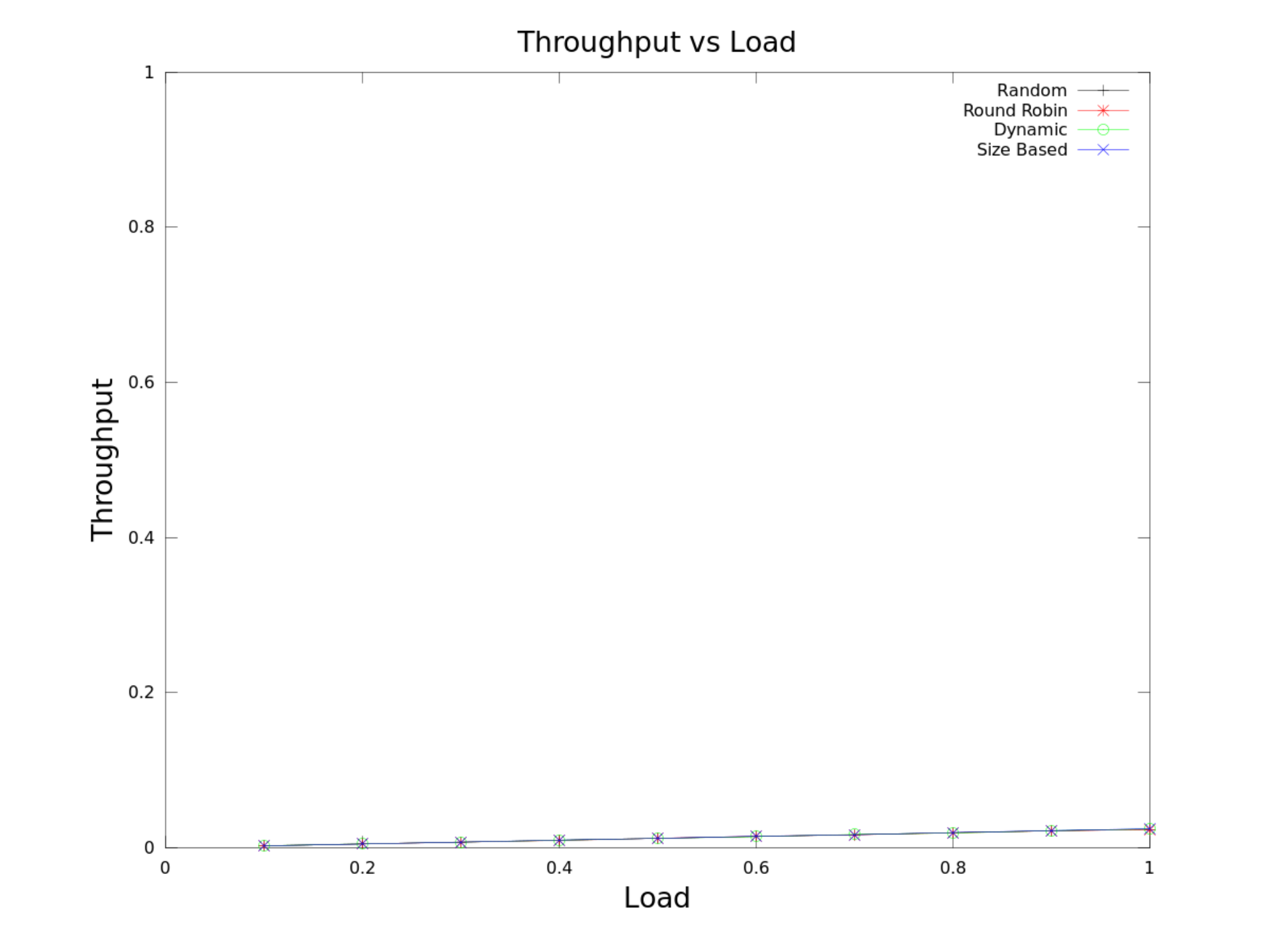

Utilization vs Number of Hosts with a=1.9
Random
Round Robin
Dynamic
Size Based
Utilization (Average of hosts)
Number of Hosts
0
0.2
0.4
0.6
0.8
1
1
2
3
4
5
6
7
8
9
10

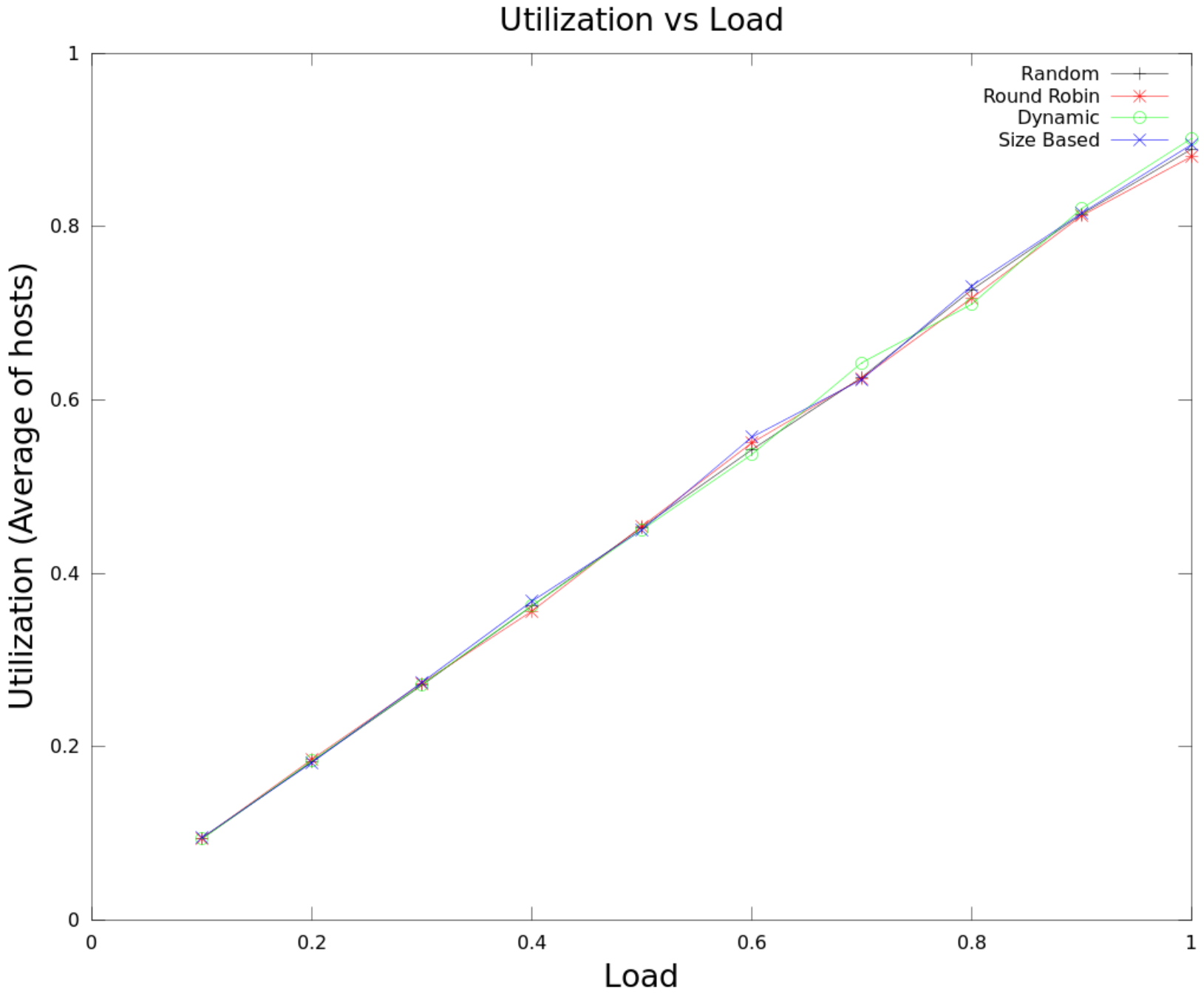
Load

- Varies between 0.1 and 1.0 with step 0.1
- Variability a = 1.1
- Number of host = 8


Mean Waiting Time vs Load
Mean Waiting Time in Units of Time
Load
Random
Round Robin
Dynamic
Size Based
120000
100000
80000
60000
40000
20000
0
0
0.2
0.4
0.6
0.8
1


Mean Slowdown vs Load
Mean SlowDown Ratio Between Waiting Time and Task Size
Load
Random
Round Robin
Dynamic
Size Based
0
50
100
150
200
250
300
350
400
0
0.2
0.4
0.6
0.8
1


Throughput vs Load
Throughput
Load
Random
Round Robin
Dynamic
Size Based
0
0.2
0.4
0.6
0.8
1


Utilization vs Load
Utilization (Average of hosts)
Load
Random
Round Robin
Dynamic
Size Based
0
0.2
0.4
0.6
0.8
1

Conclusion

- With high variance, size based policy does the best
- With ***a*** increases, variability decreases, dynamic policy outperforms others
- Size-based and dynamic always have much smaller waiting time and slowdown than random and round robin policies
- Size based policy isn't affected a lot by variability in task size
- Throughput and utilization are independent of the policy

Q & A