

1 Benchmarks

The following requirements apply to all benchmarks. If a benchmark program spans multiple machines, each machine must satisfy all of these requirements.

- Times are measured in real time, with a resolution of no worse than one second.
- The benchmark program and all related processes may use no more than 75% of the machine's physical memory, and in no event more than two gigabytes of physical memory.
- The benchmark program and all related processes may use no more than ten gigabytes of disk space.
- The benchmark program must produce the correct answer.
- The benchmark program must not encounter an error or prematurely terminate during or after the benchmark.
- The benchmark program must be in a usable condition after the benchmark.
- The benchmark program may not degrade or disable the user interface, or in other manner cause less work to be done than as expected by normal user interaction.

1.1 Test Environment

Processor	Intel Pentium 4 stepping D1 3.2 GHz clock speed 800 MHz front side bus speed
Cache	8 KB L1 data 12 KB L1 instruction 512 KB L2 unified
Memory	1 GB DDR 2700 RAM
Disk	80 GB capacity 100 MB/s interface bandwidth 49.3 MB/s average transfer rate 19.2 ms access time
Operating System	Microsoft Windows XP Service Pack 1a

Table 1: Machine Configuration

The processor HyperThreading extension was enabled during the benchmarks although this does not significantly affect performance. Almost all of the benchmarking work is done on a single thread.

Execution time was determined by the system high performance timer. The high performance timer has a resolution of one processor clock cycle. The runtime environment was configured to terminate the benchmark program if more than 75% (768 MB) of physical memory was allocated from the heap. During benchmark execution, the peak heap allocation size was recorded. Note that this value may be more than

Program	JigCell version 04302004
Compiler	IBM Jikes Java compiler version 1.19
Optimizer	Sun Java HotSpot Client VM version 1.4.2.03-b02, mixed mode
Runtime	Sun Java Runtime Environment SE version 1.4.2.03-b02

Table 2: Software Configuration

the amount of memory required by the benchmark as freed objects need not be immediately reclaimed by the runtime environment if memory is available. Additionally, a lower bound on the memory required was obtained by decreasing the amount of heap memory available to the runtime environment until the benchmark was unable to complete due to memory exhaustion.

1.2 Model Test Data

A model consists of the following elements:

- Model name
- List of compartments. Each compartment has a name and description.
- List of reactions. Each reaction has a name, description, kinetic information, and parameters. The kinetic equation for reaction i is $\frac{dS_{i+1}}{dt} = k_i * S_i - k_{i+1} * S_{i+1}$ starting with $i = 0$. Each parameter has a name and description. The value of parameter i is $\frac{1}{i+2}$.
- List of species used in reactions. Each species has a name and description.

The size of a model is determined by the number of reactions.

1.3 Model Loading Performance

Model loading performance measures the size of models supported by the Model Builder. A model is supported if the loading time is less than 300 seconds. The Model Builder must support a model containing 1000 reactions to satisfy the model building performance requirements.

The procedure for measuring model loading performance is:

1. Create a test model containing N reactions
2. Write test model to disk
3. Read test model from disk

Step 3 of the procedure is timed.

As of May 1st, the JigCell Model Builder does not pass this benchmark. The time to load a model containing 1000 reactions exceeds 626 seconds using less than 720 MB of memory and 686 KB of disk space.

1.4 Reaction Entry Performance

Reaction entry performance measures the time required to add a new reaction to the model. The Model Builder must support adding a reaction to a model containing 200 reactions in 120 seconds to satisfy the model building performance requirements.

The procedure for measuring reaction entry performance is:

1. Create a test model containing N reactions
2. Write test model to disk
3. Read test model from disk
4. Add species $N + 1$
5. Add reaction $N + 1$
6. Add parameter $N + 1$
7. Write test model to disk

Steps 3 through 7 of the procedure are timed.

As of May 1st, the JigCell Model Builder has not been run for this benchmark.

1.5 Run Test Data

1.6 Run Set Loading Performance

Run set loading performance measures the size of run sets, parameter sets, and initial condition sets supported by the Model Builder. A set is supported if the loading time is less than 300 seconds. The Model Executor must support a run set containing 10000 runs, 500 parameter sets, and 500 initial condition sets to satisfy the model execution performance requirements. The Model Executor does not need to support these limits simultaneously.

There are three procedures for measuring run set loading performance. The first procedure measures run set loading performance with the maximum run set size:

1. Create a test model containing 200 reactions
2. Write test model to disk
3. Create 1 test parameter set
4. Create 1 test initial condition set
5. Create a test run set containing N runs
6. Write test run set to disk
7. Read test run set from disk

The second procedure measures run set loading performance with the maximum parameter set size:

1. Create a test model containing 200 reactions
2. Write test model to disk
3. Create N test parameter sets
4. Create 1 test initial condition set
5. Create a test run set containing N runs
6. Write test run set to disk
7. Read test run set from disk

The third procedure measures run set loading performance with the maximum initial condition set size:

1. Create a test model containing 200 reactions
2. Write test model to disk
3. Create 1 test parameter set
4. Create N test initial condition sets
5. Create a test run set containing N runs
6. Write test run set to disk
7. Read test run set from disk

Step 7 of each of these procedures is timed.

As of May 1st, the JigCell Run Manager has not been run for these benchmarks.

1.7 Run Entry Performance

Run entry performance measures the time required to add a new run to a run set. The Model Executor must support adding a run to a run set containing 2000 runs in 300 seconds to satisfy the model execution performance requirements.

1. Create a test model containing 200 reactions
2. Write test model to disk
3. Create 100 test parameter sets
4. Create 100 test initial condition sets
5. Create a test run set containing N runs

6. Write test run set to disk
7. Read test run set from disk
8. Add run $N + 1$
9. Write test run set to disk

Steps 7 through 9 of the procedure are timed.

As of May 1st, the JigCell Run Manager has not been run for this benchmark.

1.8 Run Execution Performance

Run execution performance measures the time required to execute an existing run. The Model Executor must support executing a run for a working model from a run set containing 2000 runs in 60 seconds to satisfy the model execution performance requirements.

The procedure for measuring run execution performance is:

1. Create a test model containing 200 reactions
2. Write test model to disk
3. Create 100 test parameter sets
4. Create 100 test initial condition sets
5. Create a test run set containing N runs
6. Write test run set to disk
7. Read test run set from disk
8. Execute run N

Steps 7 and 8 of the procedure are timed.

As of May 1st, the JigCell Run Manager has not been run for this benchmark.

1.9 Run Modification Performance

Run modification performance measures the time required to modify an existing run. The Model Executor must support modifying a run for a working model from a run set containing 2000 runs in 60 seconds to satisfy the model execution performance requirements.

The procedure for measuring reaction modification performance is:

1. Create a test model containing 200 reactions
2. Write test model to disk
3. Create 100 test parameter sets

4. Create 100 test initial condition sets
5. Create a test run set containing N runs
6. Write test run set to disk
7. Read test run set from disk
8. Set the parameter for reaction 200 in parameter set 100 to 0
9. Set the initial condition for species 200 in initial condition set 100 to 0
10. Write test run set to disk

Steps 7 through 10 of the procedure are timed.

As of May 1st, the JigCell Run Manager has not been run for this benchmark.

1.10 Experiment Set Test Data

An experiment set consists of experiments. Each experiment has the following elements:

- Experiment name
- Globally unique identifier
- Experimental observations. The experimental value is a two variable time series containing 4 observations. The first variable of experiment observation i is the integer i and the second variable is a finite, normalized real number.
- Experimental type. The type must be compatible with the time series format of the experimental value.
- Experiment description

The size of an experiment set is determined by the number of experiments.

1.11 Experiment Loading Performance

Experiment loading performance measures the size of experiment sets supported by the Model Analyzer. An experiment set is supported if the loading time is less than 300 seconds. The Model Analyzer must support an experiment set containing 10000 experiments to satisfy the model analysis performance requirements.

The procedure for measuring experiment loading performance is:

1. Create a test experiment set containing N experiments
2. Write test experiment set to disk
3. Read test experiment set from disk

Step 3 of the procedure is timed.

As of May 1st, the JigCell Comparator passes this benchmark. The time to load an experiment set containing 10000 experiments is less than 5 seconds using less than 64 MB of memory and 5011 KB of disk space.

1.12 Experiment Entry Performance

Experiment entry performance measures the time required to add a new experiment to an experiment set. The Model Analyzer must support adding an experiment to an experiment set containing 2000 experiments in 120 seconds to satisfy the model analysis performance requirements.

The procedure for measuring experiment entry performance is:

1. Create a test experiment set containing N experiments
2. Write test experiment set to disk
3. Read test experiment set from disk
4. Add experiment $N + 1$
5. Write test experiment set to disk

Step 3 through 5 of the procedure is timed.

As of May 1st, the JigCell Comparator passes this benchmark. The time to add a new experiment to an experiment set containing 2000 experiments is less than 3 seconds using less than 64 MB of memory and 998 KB of disk space.

1.13 Transform Set Test Data

A transform set consists of transforms and a mapping function from experiment names to transforms. Each transform has the following elements:

- Transform name
- Globally unique identifier
- Transformation procedure. The transformation procedure is a functional specification of what the transform does.
- Parameters used by the transformation procedure
- Reference to the transform that provides input to this transform, if needed
- List of transforms contained in this transform

The procedure defined for each named experiment consists of two transforms and requires one parameter. The first transform received input from the second transform; the second transform does not require input. Neither transform contains any transforms. The output of the procedure is a time series of the same format as an experimental observation. The size of a transform set is determined by the number of mappings from experiment names to transforms.

1.14 Transform Loading Performance

Transform loading performance measures the size of transform sets supported by the Model Analyzer. A transform set is supported if the loading time is less than 300 seconds. The Model Analyzer must support a transform set for 10000 experiments, of which at least 500 of the transforms are different, to satisfy the model analysis performance requirements. When running this benchmark, all of the generated transforms are different.

The procedure for measuring loading performance is:

1. Create a test experiment set containing N experiments
2. Write test experiment set to disk
3. Create a test transform set containing N transforms
4. Write test transform set to disk
5. Read test experiment set from disk
6. Read test transform set from disk

Step 6 of the procedure is timed.

As of May 1st, the JigCell Comparator passes this benchmark. The time to load a transform set for 10000 experiments is less than 25 seconds using less than 256 MB of memory and 15971 KB of disk space.

1.15 Objective Set Test Data

An objective set consists of objectives and a mapping function from experiment names to objectives. Each objective has the following elements:

- Objective name
- Globally unique identifier
- Objective procedure. The objective procedure is a functional specification of what the objective does.
- Parameters used by the objective procedure

The procedure defined for each named experiment requires one parameter. The size of an objective set is determined by the number of mappings from experiment names to objectives.

1.16 Objective Function Loading Performance

Objective function loading performance measures the size of objective sets supported by the Model Analyzer. An objective set is supported if the loading time is less than 300 seconds. The Model Analyzer must support an objective set for 10000 experiments, of which at least 50 of the objective functions are different, to satisfy the model analysis performance requirements. When running this benchmark, all of the generated objective functions are different.

The procedure for measuring objective function loading performance is:

1. Create a test experiment set containing N experiments
2. Write test experiment set to disk
3. Create a test transform set containing N transforms
4. Write test transform set to disk
5. Create a test objective set containing N objectives
6. Write test objective set to disk
7. Read test experiment set from disk
8. Read test transform set from disk
9. Read test objective set from disk

Step 9 of the procedure is timed.

As of May 1st, the JigCell Comparator passes this benchmark. The time to load an objective set for 10000 experiments is less than 10 seconds using less than 256 MB of memory and 15169 KB of disk space.

1.17 Experiment Configuration Performance

Experiment configuration performance measures the time required to configure the transform and objective function for a new experiment. The Model Analyzer must support configuring an experiment in an experiment set containing 2000 experiments in 300 seconds to satisfy the model analysis performance requirements.

The procedure for measuring experiment configuration performance is:

1. Create a test experiment set containing $N + 1$ experiments
2. Write test experiment set to disk
3. Create a test transform set containing N transforms
4. Write test transform set to disk
5. Create a test objective set containing N objectives

6. Write test objective set to disk
7. Read test experiment set from disk
8. Read test transform set from disk
9. Create transform $N + 1$
10. Add transform $N + 1$ to experiment $N + 1$
11. Write test transform set to disk
12. Read test objective set from disk
13. Create objective $N + 1$
14. Add objective $N + 1$ to experiment $N + 1$
15. Write test objective set to disk

Steps 7 through 16 of the procedure are timed.

As of May 1st, the JigCell Comparator passes this benchmark. The time to configure an experiment in an experiment set containing 2000 experiments is less than 20 seconds using less than 256 MB of memory and 5207 KB of disk space.

1.18 Summary of Results

As of May 1st, 6 of the 13 benchmarks have been completed and run. JigCell passes 5 of the benchmarks and fails 1 of the benchmarks.

#	Time(s)	Memory(MB)	Disk(KB)	Passes
1.3	626.28	512-720	686	
1.4				
1.6a				
1.6b				
1.6c				
1.7				
1.8				
1.9				
1.11	5.00	32-49	5011	•
1.12	2.89	32-41	998	•
1.14	24.03	128-187	15971	•
1.16	9.94	128-150	15169	•
1.17	19.45	32-256	5207	•

Table 3: Benchmark Results Summary

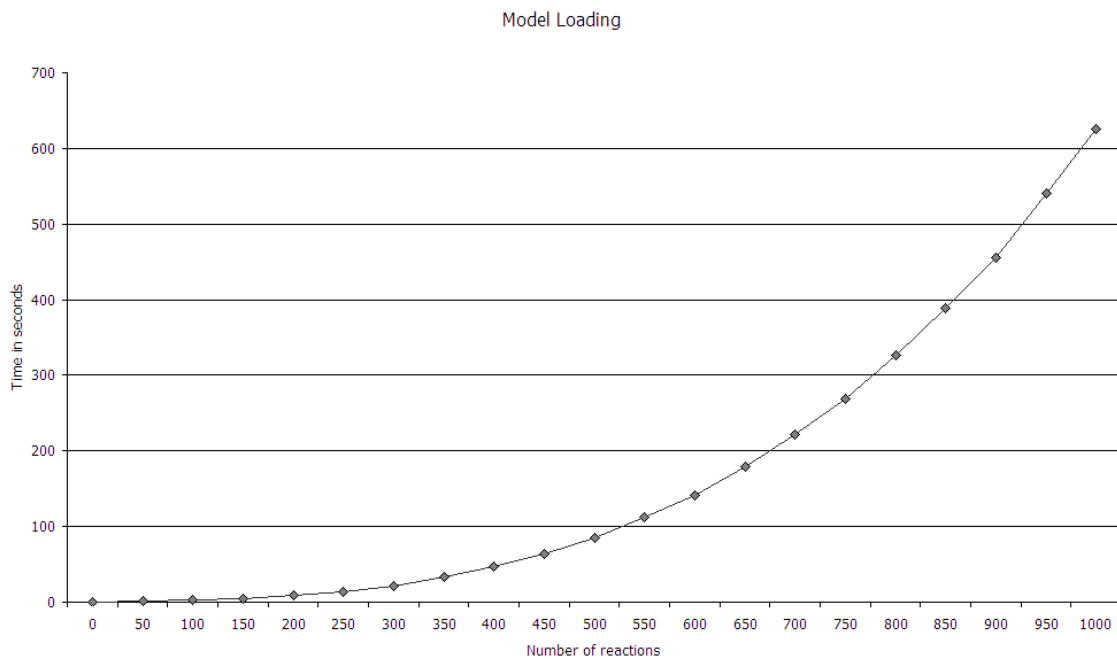


Figure 1: Benchmark 1.3

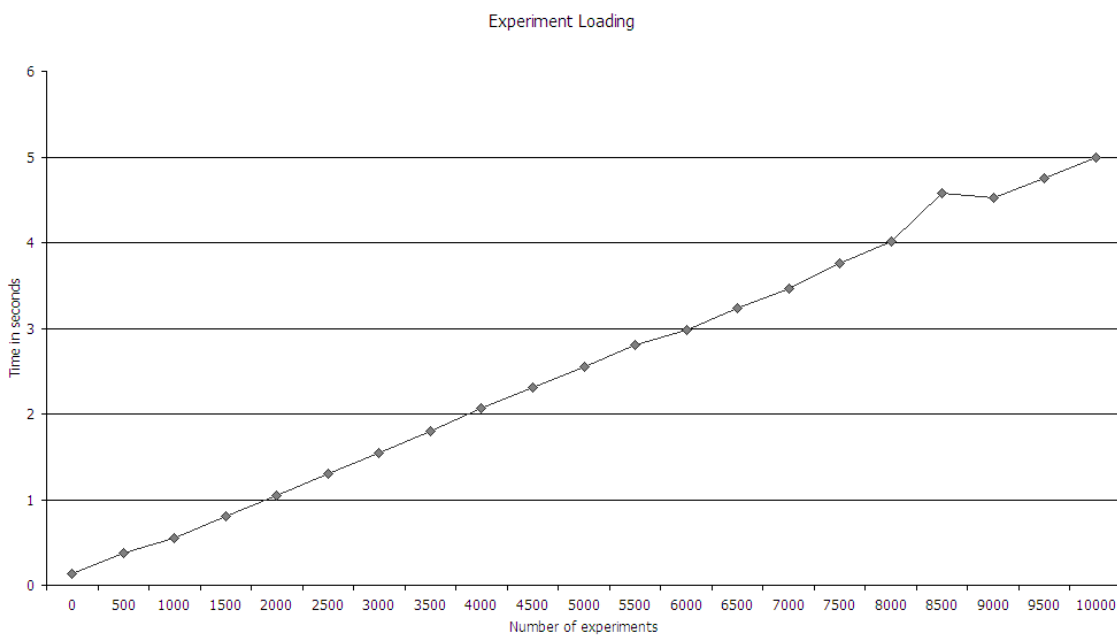


Figure 2: Benchmark 1.11

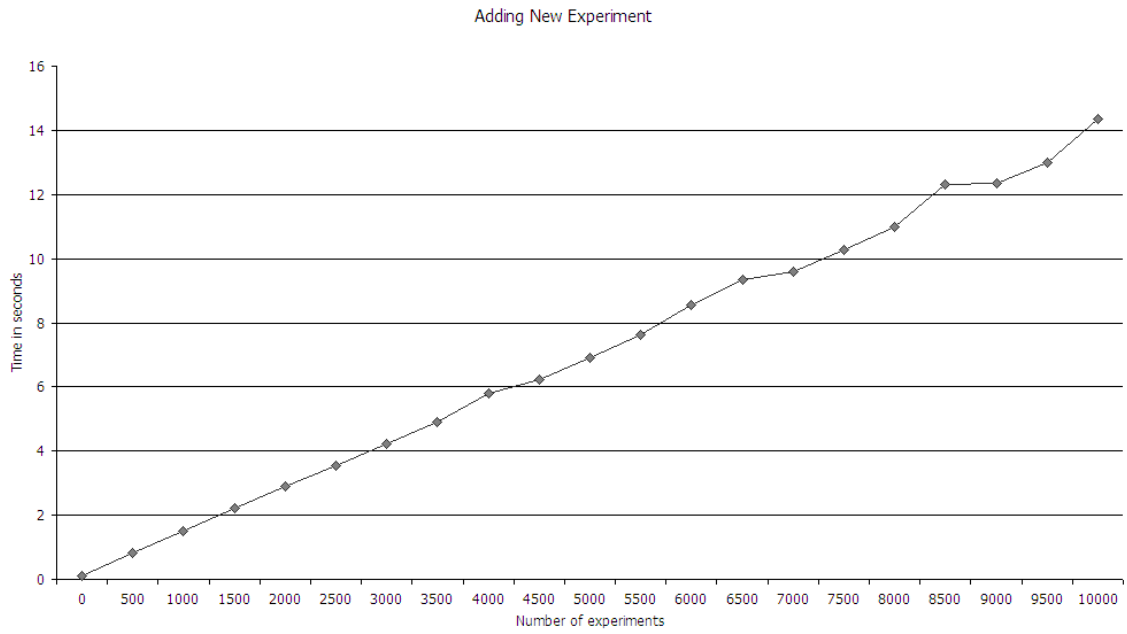


Figure 3: Benchmark 1.12

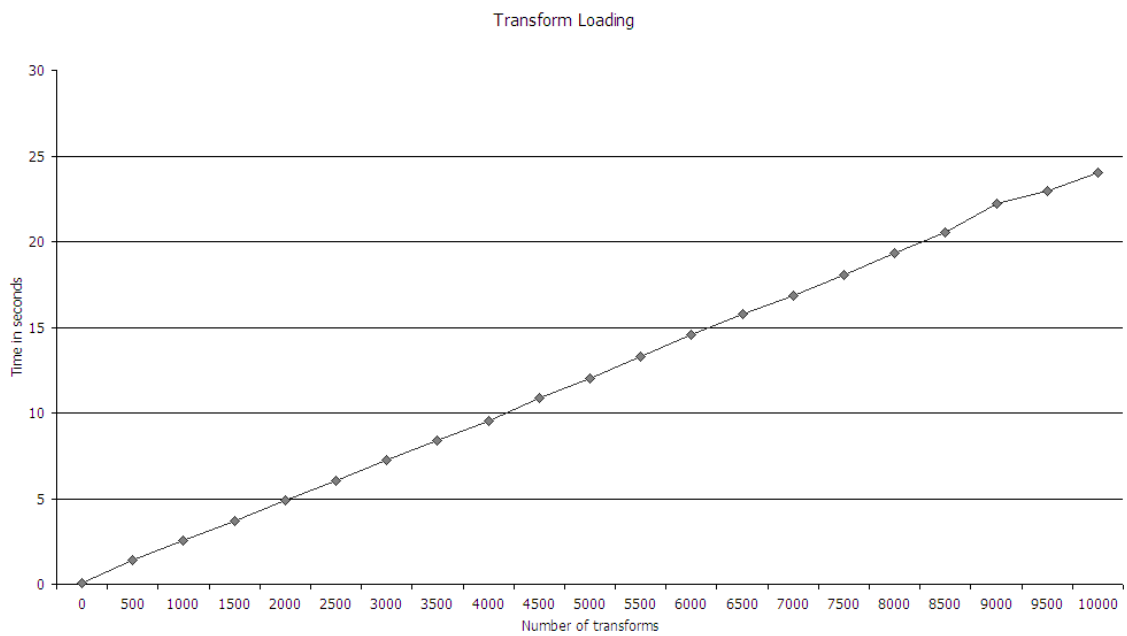


Figure 4: Benchmark 1.14

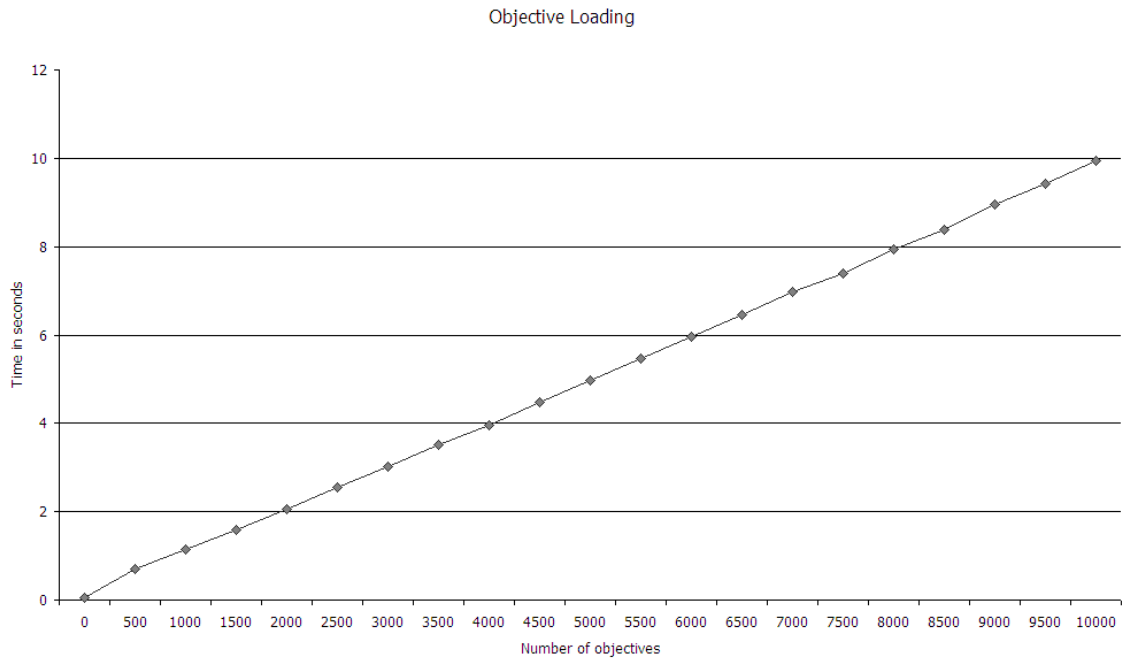


Figure 5: Benchmark 1.16
Configuring Experiment

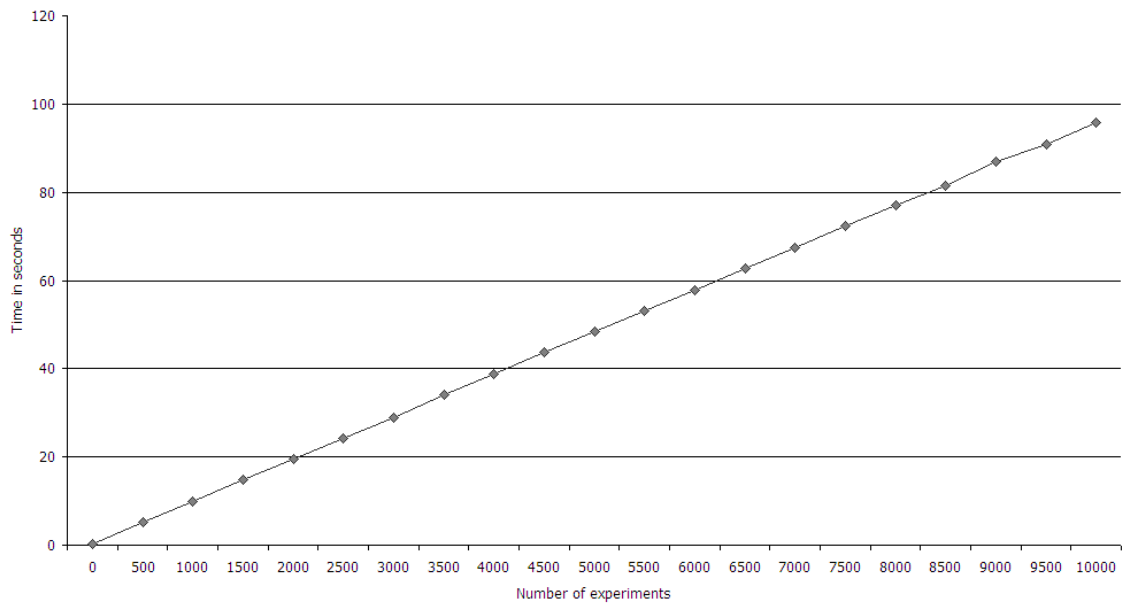


Figure 6: Benchmark 1.17