A Java Grande Benchmark to compare Java and C++ performance for a discrete event simulation application

Ana Bosque Arbiol

Contents

- A Java Grande Benchmark to compare… what???
- Ns, a network simulator
- How the scheduler works
- The project
- Outstanding work
- Summary

Java Grande Benchmark?

Java = programming language

Grande = application with large memory, network or computational requirements

Java wasn't built for being used in these kinds of applications but...

Java important features

Java has a number of important features:

- Portability
- Nonexistence of pointers
- No need of complex Makefiles
- Remote method invocation (RMI)
- Remote file access
- Database access
- Good software engineering environments

Lacks of Java

Java has also some disadvantages:

- Lack of some features for scientific programmers
  (multidimensional arrays, complex numbers, …)
- Absence of the familiar parallel programming models
  (MPI and OpenMP)
- Performance

Java Grande Benchmark

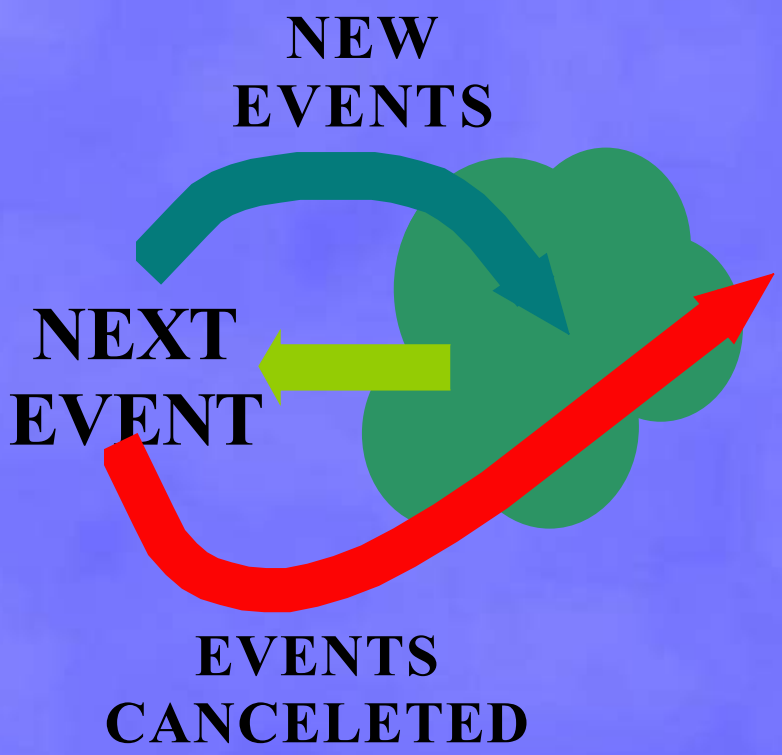
The aim is to provide the community a standard benchmark suite which can be used to:

- Demonstrate the use of Java for Grande applications.
- Provide metrics for comparing Java execution environments
- Expose features of the environments critical for Grande Applications

Further information:
www.epcc.ed.ac.uk/research/publications/journal/javabenchcpe.ps.gz
www.epcc.ed.ac.uk/research/publications/conference/jgflangcomp_final.ps.gz

A discrete event simulator?

A discrete event simulation application is a simulator which works with a set of initial events generating new events or cancelating the old ones in the execution of each event.



Ns, a network simulator

Ns is a network simulator that simulates variety of IP networks. It implements network protocols such as almost all variants of TCP and UDP, several forms of multicast, traffic source behaviour such as FTP, Telnet, Web, ... www.isi.edu/nsnam/ns

The ns project is now a part of the VINT project. The aim of this project is to build a network simulator that will allow the study of scale and protocol interaction in the context of current and future network protocols. www.isi.edu/nsnam/vint/index.html

Different parts of ns

- Tcl interface
- Network component object libraries
- Network setup module libraries

} Event Generator

- Event scheduler → Event treatment

Event scheduler

In ns, an event scheduler keeps track of simulation time and fires all the events in the event queue scheduled for the current time invoking appropriate network components.

Basically, it's an structure in which are really important the next operations:

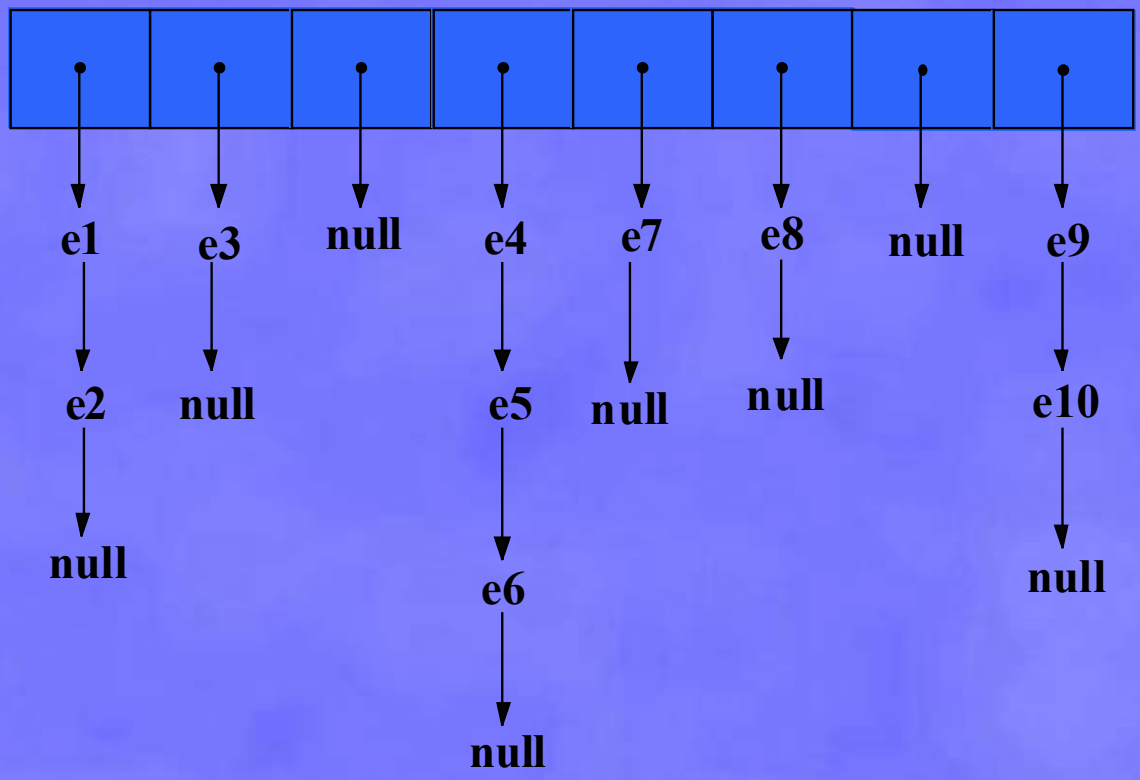
- Insertion
- Deletion
- Look for the next event to execute

Different implementations of the scheduler

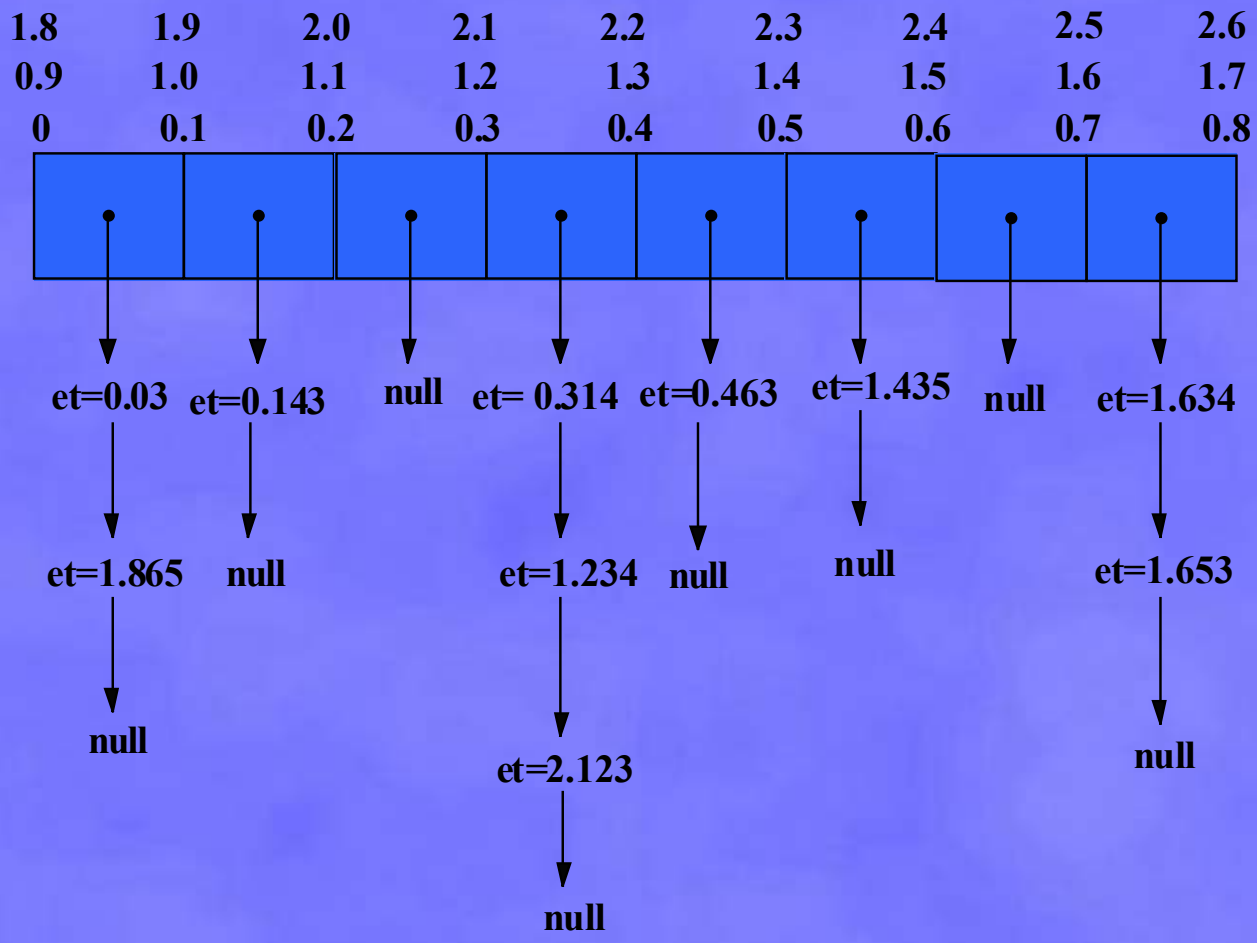
We can think in a lot of different data structures. In the ns project three different structures have been implemented:

- List
- Heap
- Calendar

Calendar Scheduler



Calendar Scheduler

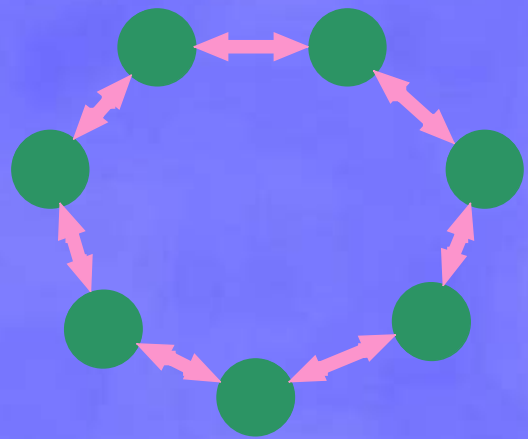


Calendar Scheduler execution

Models executed:

- 50x10Mb_simple_tcp.tcl

- ring-master.tcl



- number of nodes
- web clients per node
- ftp clients per node
- distance between servers

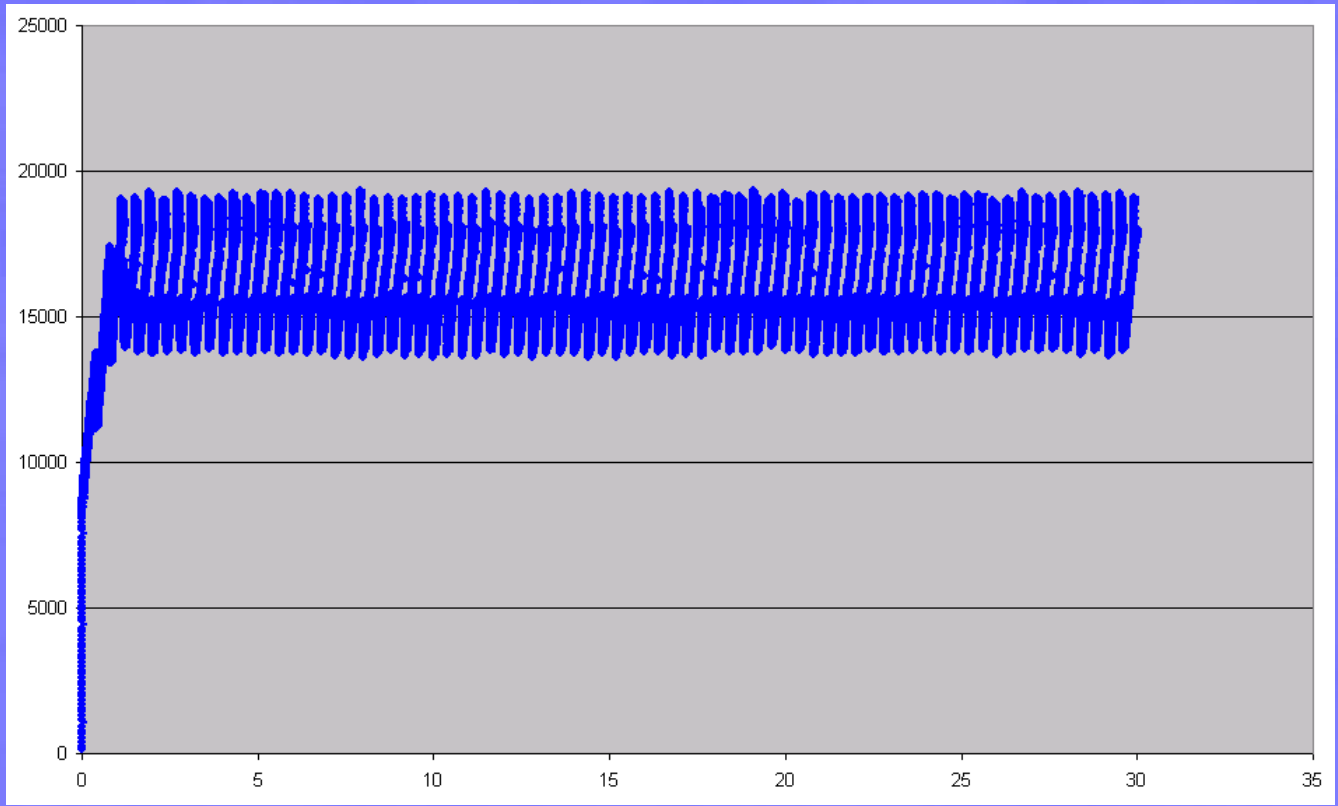
Calendar Scheduler execution

Scheduler time vs. Event execution time

NODES	
16	21.88%
32	22.25%
128	28.02%
256	34.23%
1024	32.78%
2048	64.28%

Calendar Scheduler execution

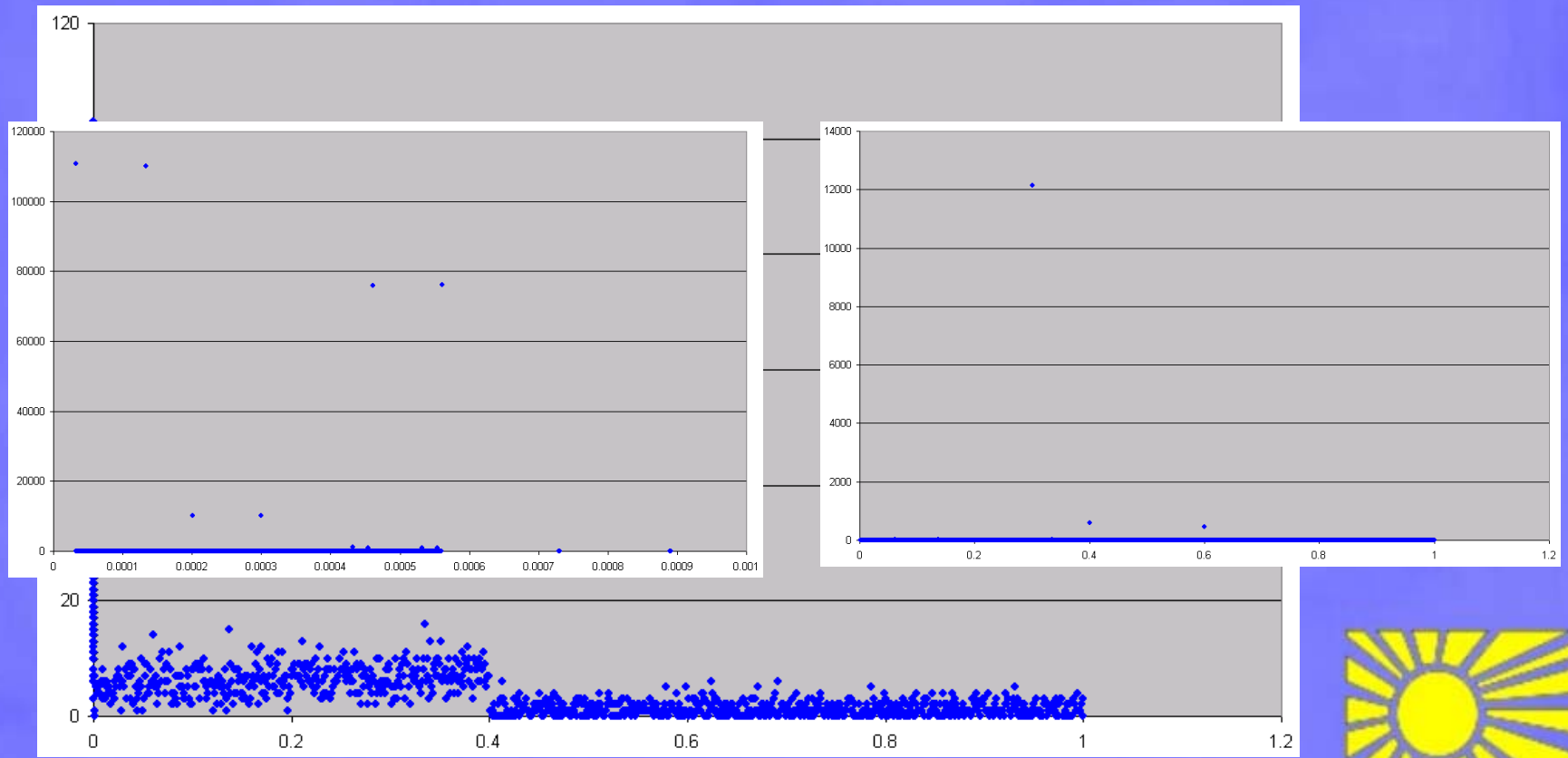
Load of the Scheduler



Number of nodes = 256

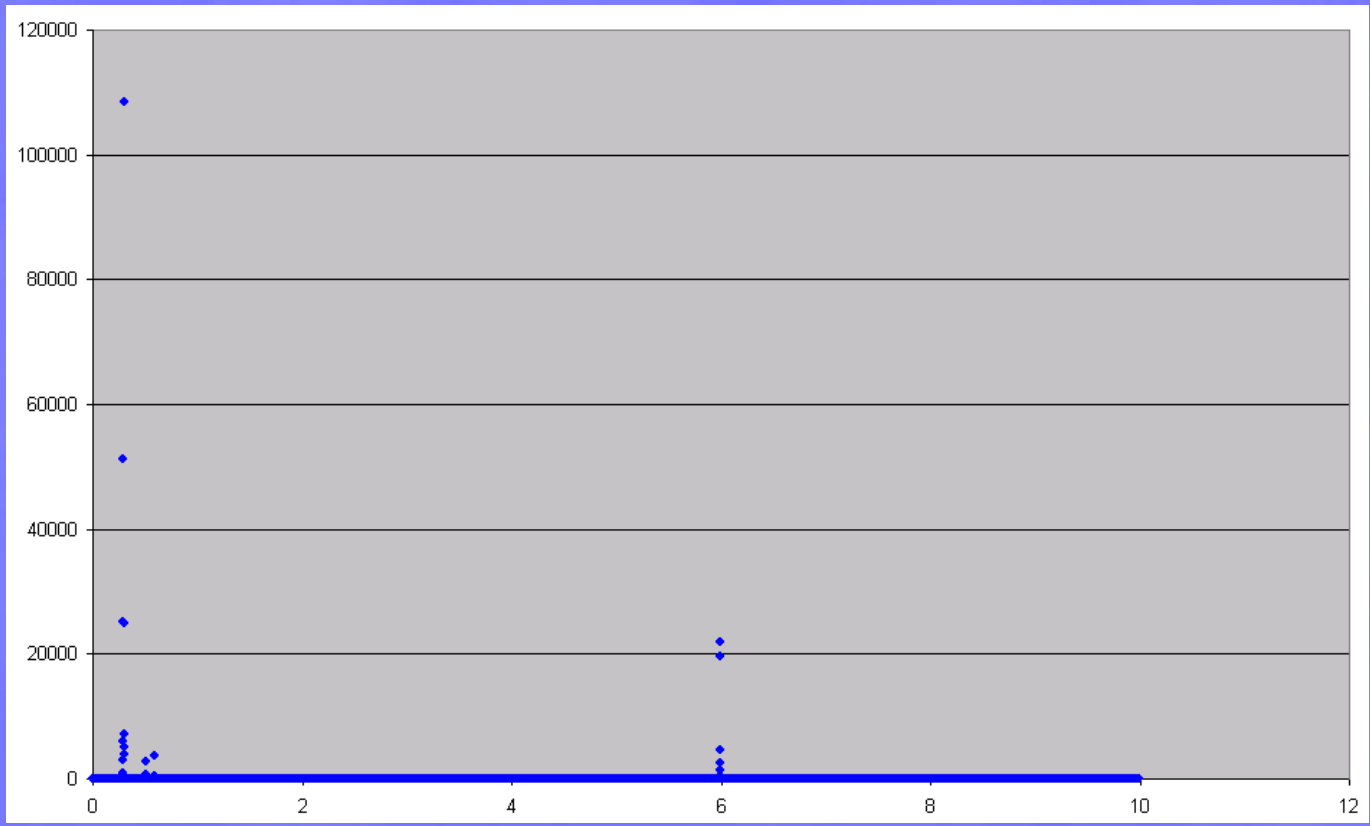
Calendar Scheduler execution

Insertion delay events distribution



Calendar scheduler execution

Cancellation delay events distribution



Our model

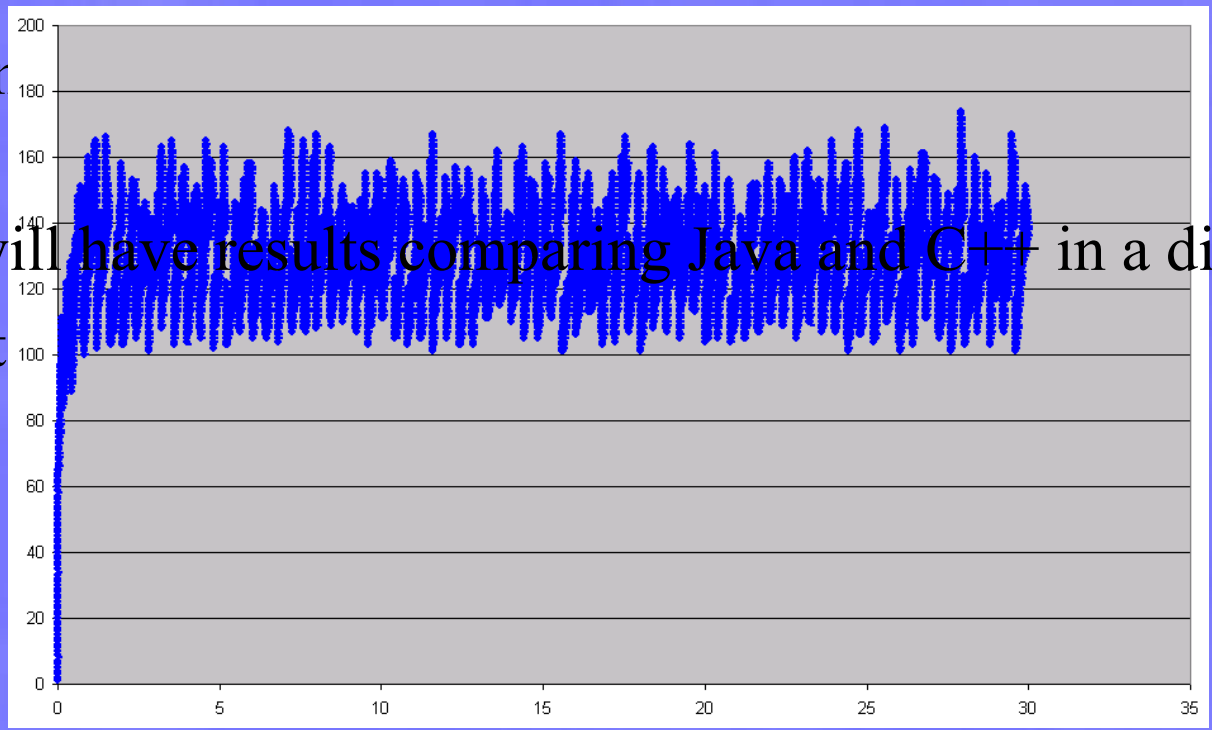
We use trace files instead of building the insertion and cancellation distributions.

Information in these trace files:

- Insertion: delay
- Cancellation: timestamp and id of the event
- Look up: id of the event

Outstanding work

- I have the model in Java and C++
- I am r
- We will have results comparing Java and C++ in a discrete event



Summary

- Why a Java Grande Benchmark?
- Why the ns simulator?
- How works the treatment of events in ns?
- How did I build my model?
- Where am I now?

Questions