# DEVS Tutorial--
# DEVS and Distributed DEVS

Ming Zhang, PhD
Paradise Research Laboratory
School of Information Technology and Engineering
University of Ottawa

# Outline

- **How to Find DEVS Materials:**
- **Google: ACIMS**
- DEVS Key Concepts
- DEVS Tools and Distributed DEVS Tools:
- *DEVS/Grid, DEVS/P2P, DEVS/SOA…*
- DEVS/RMI—A Reconfigurable Distributed DEVS Framework
- Solving large-scale simulation models using DEVS/RMI
- DEVS in the near future

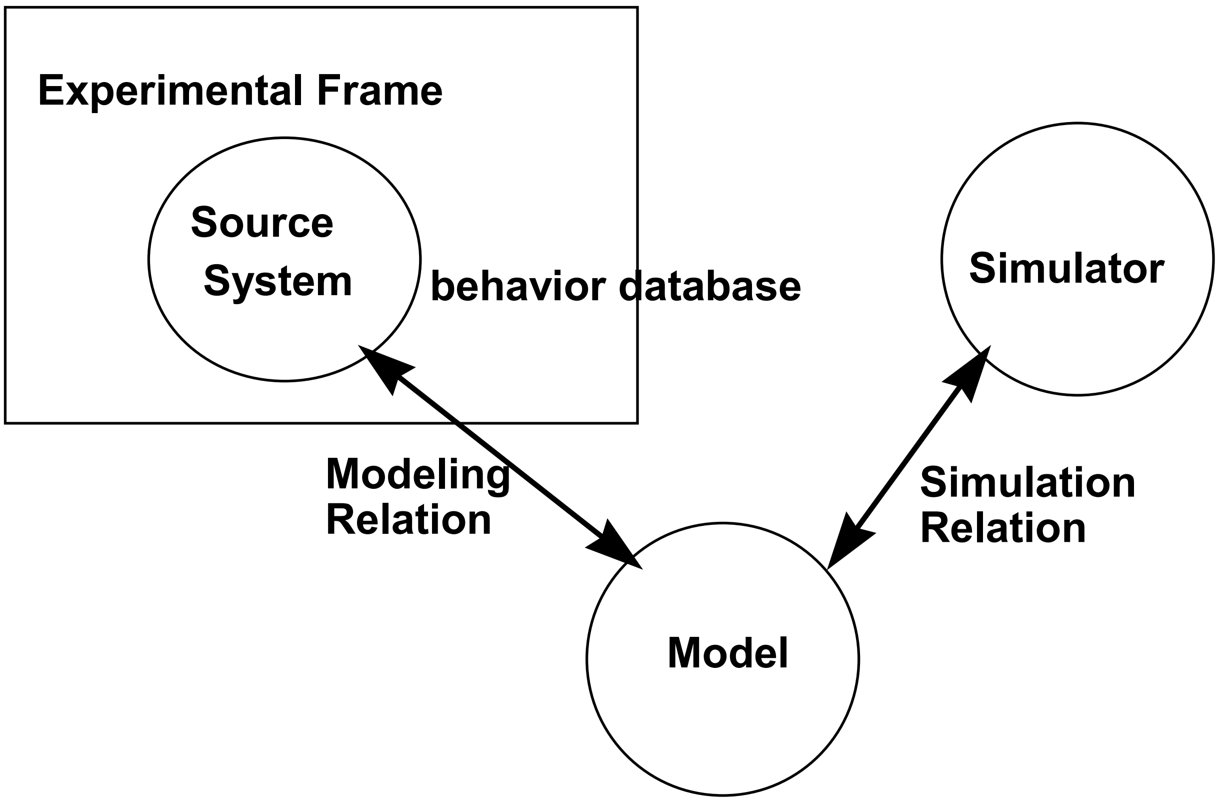
# What is and Why Use DEVS

- **DEVS—Discrete Event System Specification**
- Strictly Defined Modeling and Simulation Framework based on DEVS Formalism.
- Flexible Hierarchical Modeling and Simulation Structure.
- Support Model Reuse by Model Repository.
- Support both discrete event and continuous system modelling and simulation.
- Can be used for formalized design and system design verification and validation.
- Can be used for agent-based simulation.
- And more...

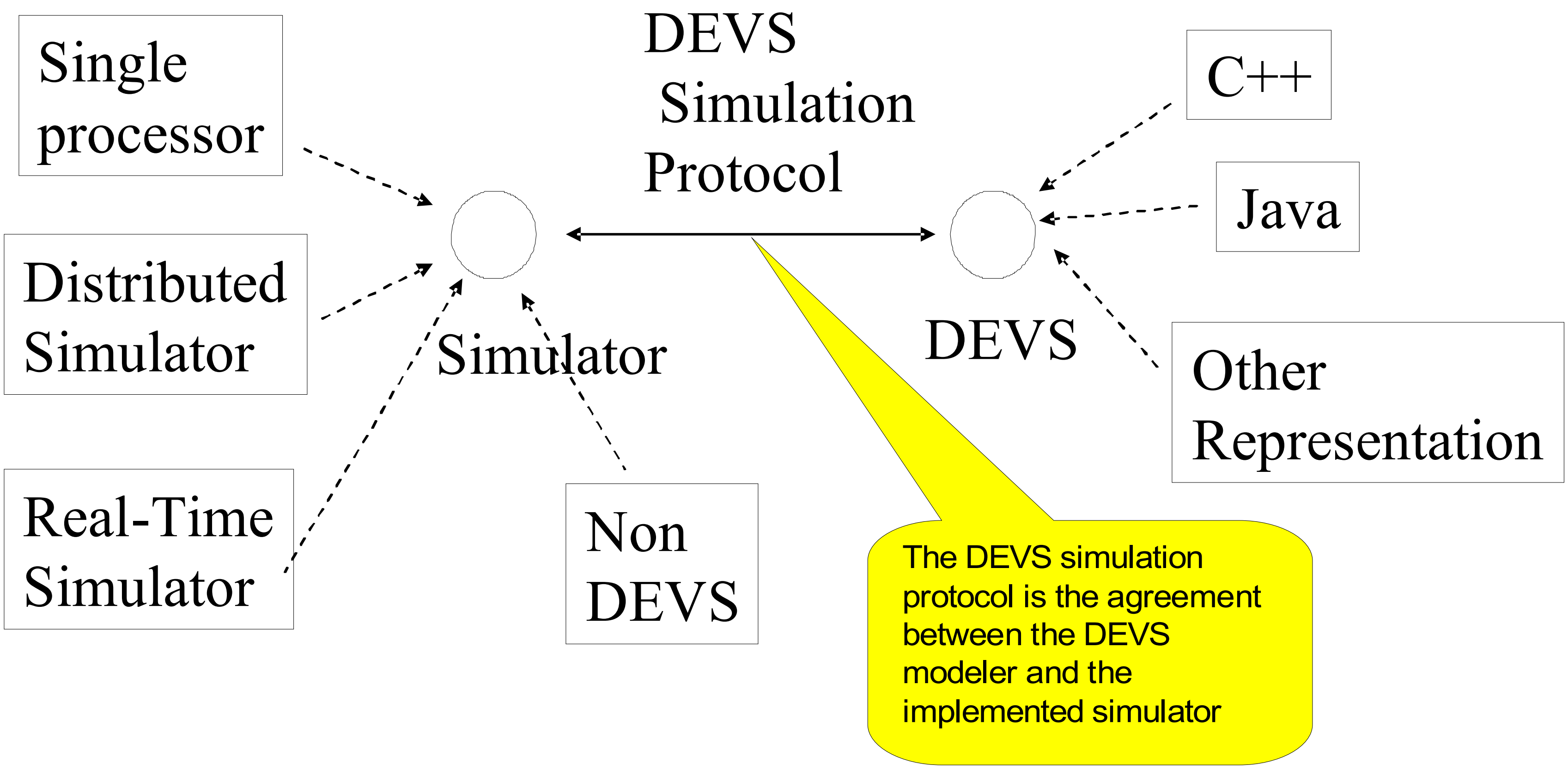
# DEVS and Non-DEVS based Simulation

- DEVS is formalized.
- DEVS is hierarchical.
- Clearly separating modeling and simulation framework.
- Models' behavior and their inter-relation are separated.

# Concept View of Entity Relationship[*]



# Separate Model and Simulator [*]



# DEVS Formalism

- DEVS stands for Discrete Event System Specification.
- DEVS Formalism is used to strictly define the model component behaviour.
- DEVS Formalism has many extensions to satisfy the emerging requirements.

# Basic DEVS Formalism[*]

A *Discrete Event System Specification* (*DEVS*) is a structure

$$M = \langle X, S, Y, \delta_{int}, \delta_{ext}, \delta_{con}, \lambda, ta \rangle$$

where

$X$ is the set of input values

$S$ is a set of states,

$Y$ is the set of output values

$\delta_{int}: S \rightarrow S$ is the *internal transition* function

$\delta_{ext}: Q \times X^b \rightarrow S$

is the *external transition* function, where

$Q = \{(s,e) \mid s \in S, 0 \leq e \leq ta(s)\}$ is the *total state* set

$e$ is the *time elapsed* since last transition

$X^b$ denotes the collection of bags over X

(sets in which some elements may occur more than once).

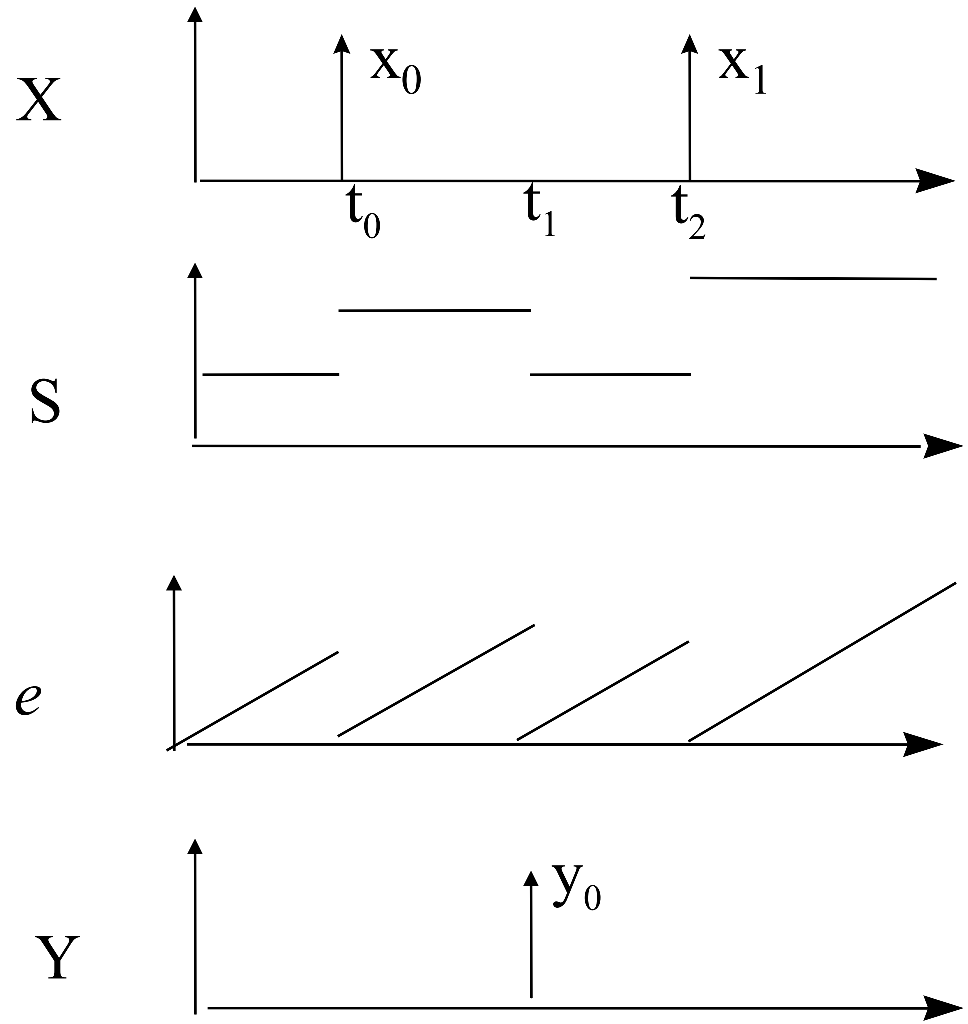
$\delta_{con}: Q \times X^b \rightarrow S$

is the *confluent transition* function,

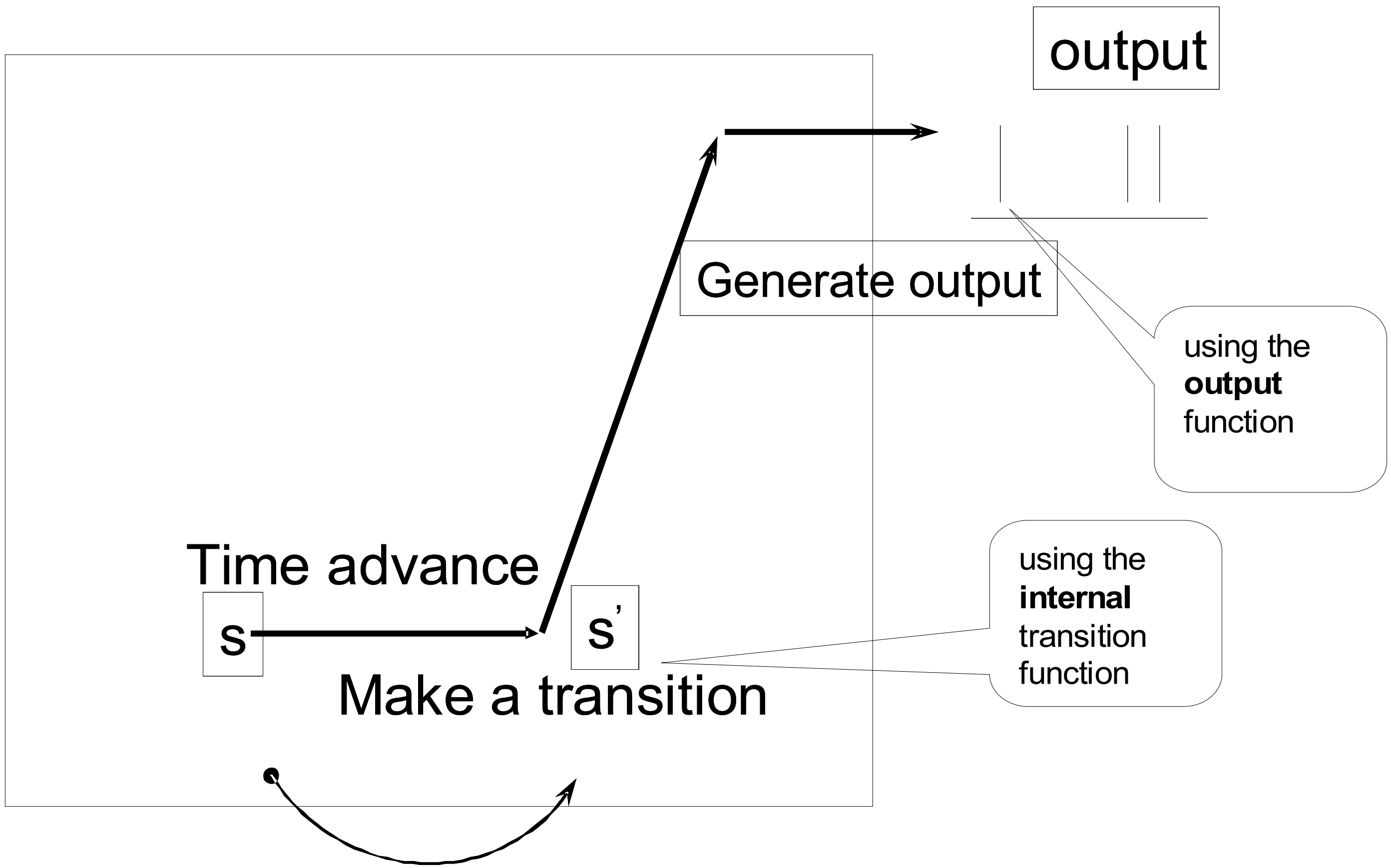
$\lambda: S \rightarrow Y^b$ is the output function

$ta: S \rightarrow \mathbf{R}^{+}_{0,\infty}$ is the *time advance* function

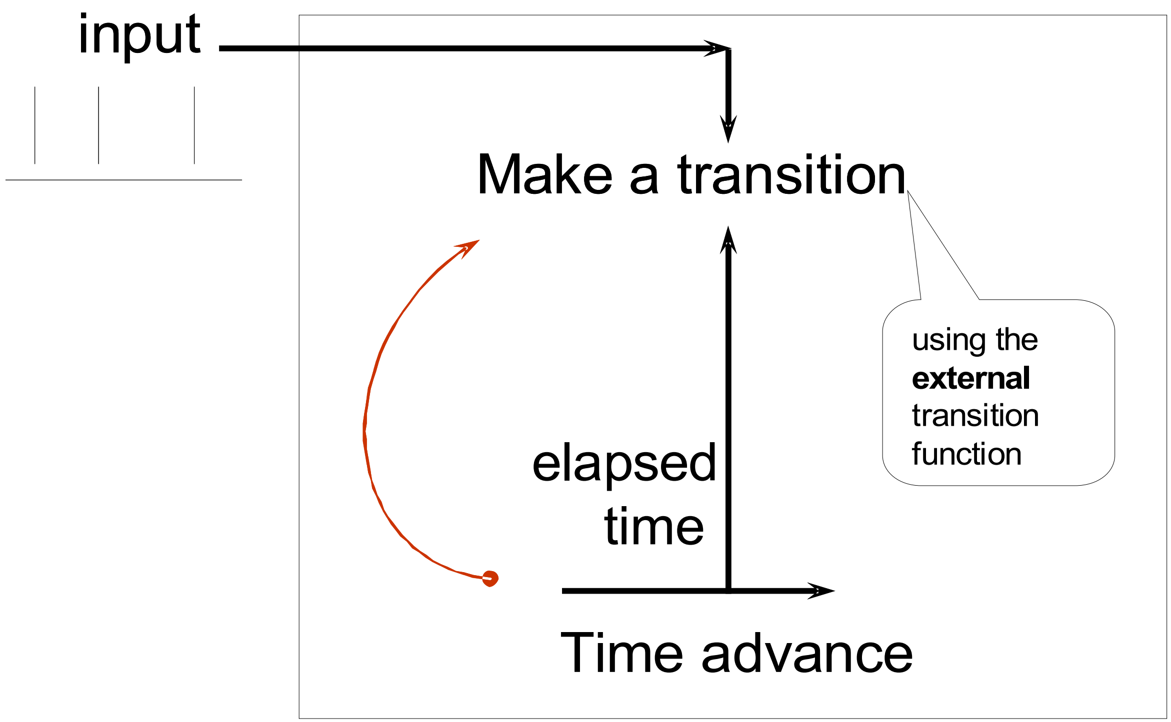
# Basic Concept in DEVS[*]



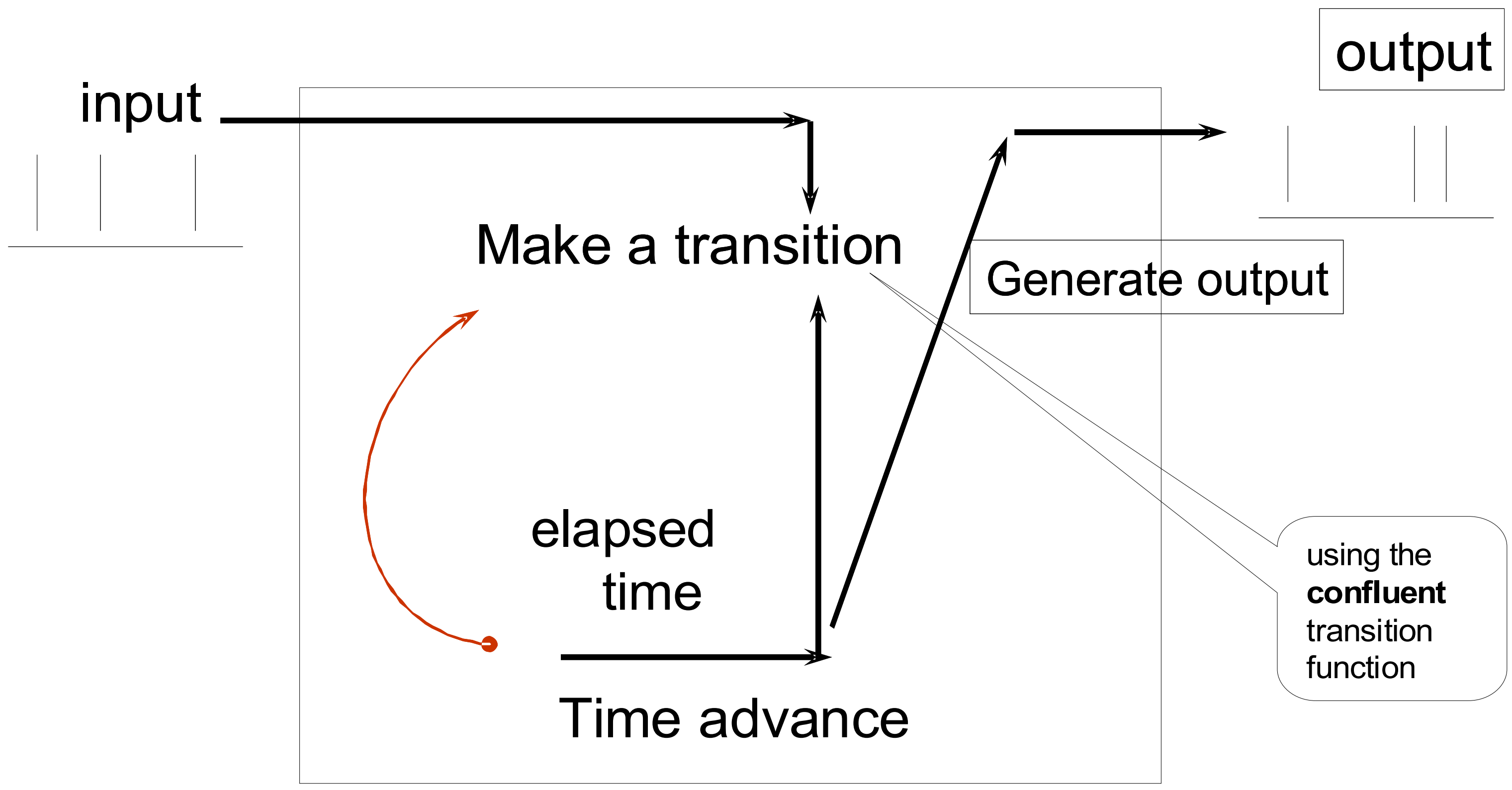
# Internal Transition[*]



# External Transition[*]



# Confluent Function[*]



# Basic DEVS Formalism Example[*]

$DEVS_{scuba} = (X, Y, S, \delta_{ext}, \delta_{int}, \lambda, ta)$

where

$X = \{\}$

$Y = \{1\}$

$S = \{\text{"sixty"}, \text{"forty"}, \text{"five"}, \text{"surface1"}, \text{"surface2"}\} \times R_{0,\infty}^{+}$

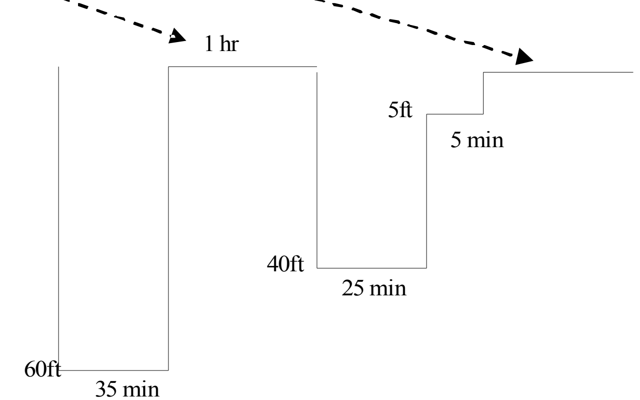
**Dive Plan**

$\delta_{int}(\text{"sixty"}, \sigma) = (\text{"surface1"}, 60)$

$\delta_{int}(\text{"surface1"}, \sigma) = (\text{"forty"}, 25)$

$\delta_{int}(\text{"forty"}, \sigma) = (\text{"five"}, 5)$

$\delta_{int}(\text{"five"}, \sigma) = (\text{"surface2"}, \infty)$




$\lambda(phase, \sigma) = 1$

$ta(phase, \sigma) = \sigma$

**Emergency Phone Call Response**

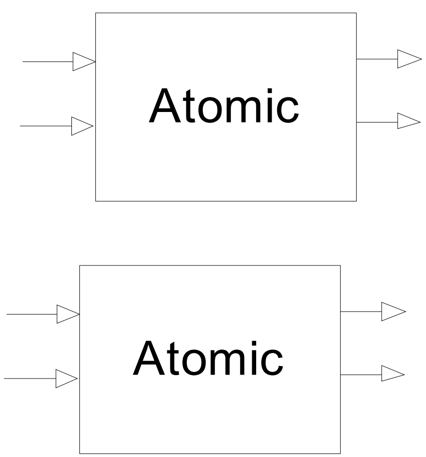
$\delta_{ext}(\text{phase}, \sigma, \text{e}, \text{x}) = (\text{"five"}, 5)$ if phase != "surface1" , "surfac2",or "5"

$= (phase, \sigma - e)$ *otherwise*

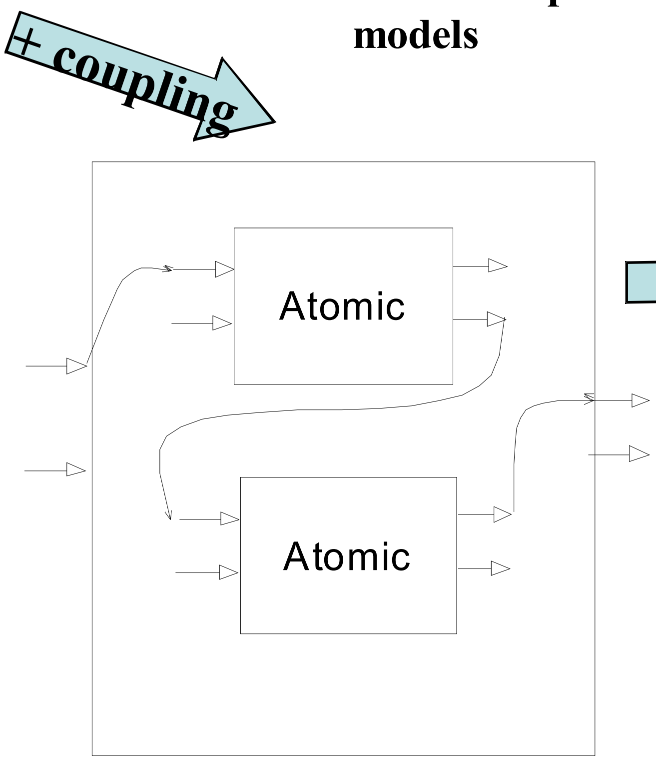
$\delta_{con}(\text{phase}, \sigma, \text{e}, \text{x}) = \delta_{ext}(\text{phase}, \sigma, \text{e}, \text{x})$ //pay attention to external event (call)

# Hierarchical Model Construction[*]

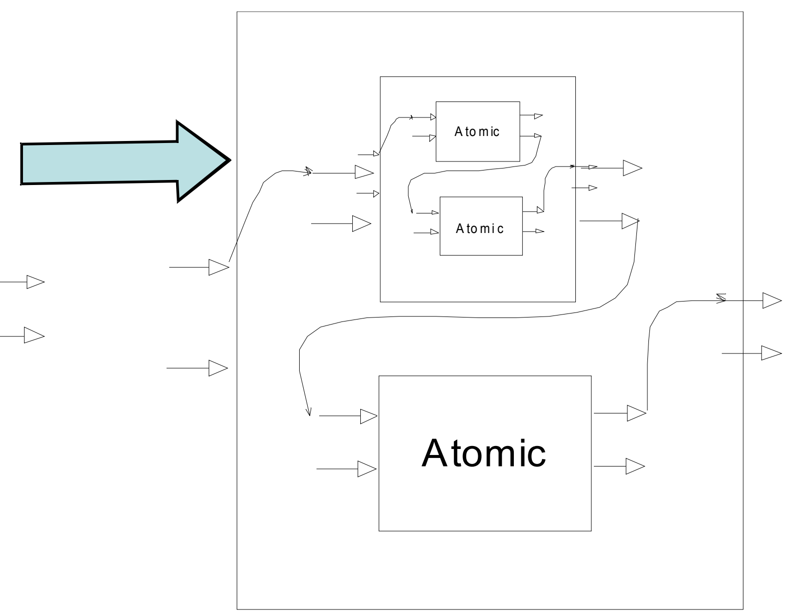

**Atomic: lowest level model, contains structural dynamics -- model level modularity**




**Coupled: composed of one or more atomic and/or coupled models**




**hierarchical construction**



# DEVS Tools

- ADEVS (by Dr. Nutaro, ORNL)
- CD++ (by Dr. Wainer, Carleton Univ.)
- Tools Developed at ACIMS, Univ. of Arizona:
- DEVSJAVA
- DEVS/CORBA
- DEVS/Grid
- DEVS/P2P
- DEVS/SOA
- DEVS/HLA
- DEVS/RMI

# DEVSJAVA

- Java Implementation of DEVS
- Support discrete and continuous system modeling and simulation
- Support As-Fast-As-Can, Real Time DEVS simulation.
- Support Distributed Simulation, but with limited functionalities.
- Support dynamic structure changes through “variable structure”.

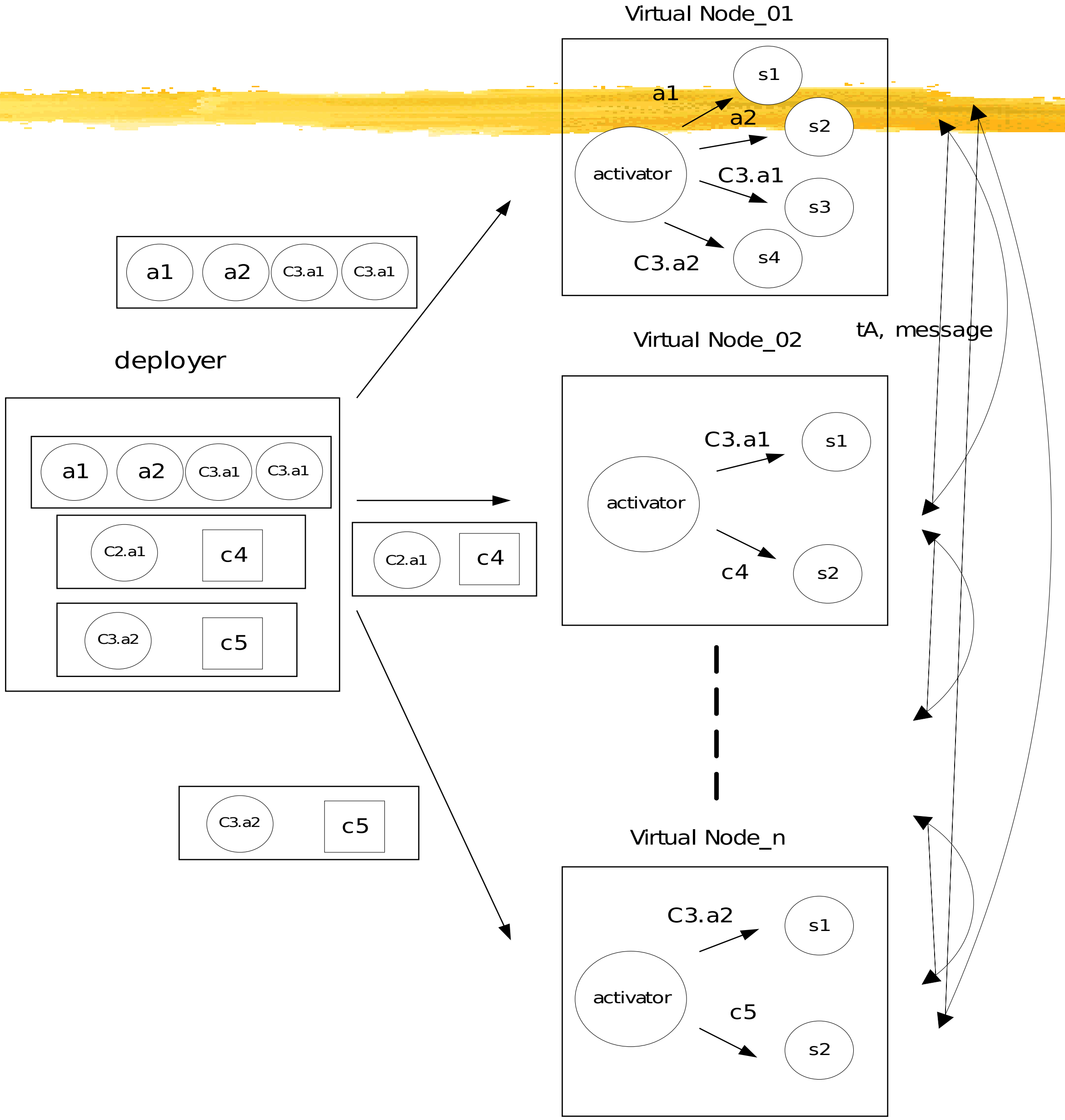
# Distributed DEVS--Why?

- Reducing model execution time.
- Overcoming limited memory for a single machine to handle large models.
- Obtaining scalable performance.
- Handling geographically distributed users and/or resources (e.g., databases, specialized equipment).
- Integrating simulations running on different platforms.
- Dealing with fault tolerance.

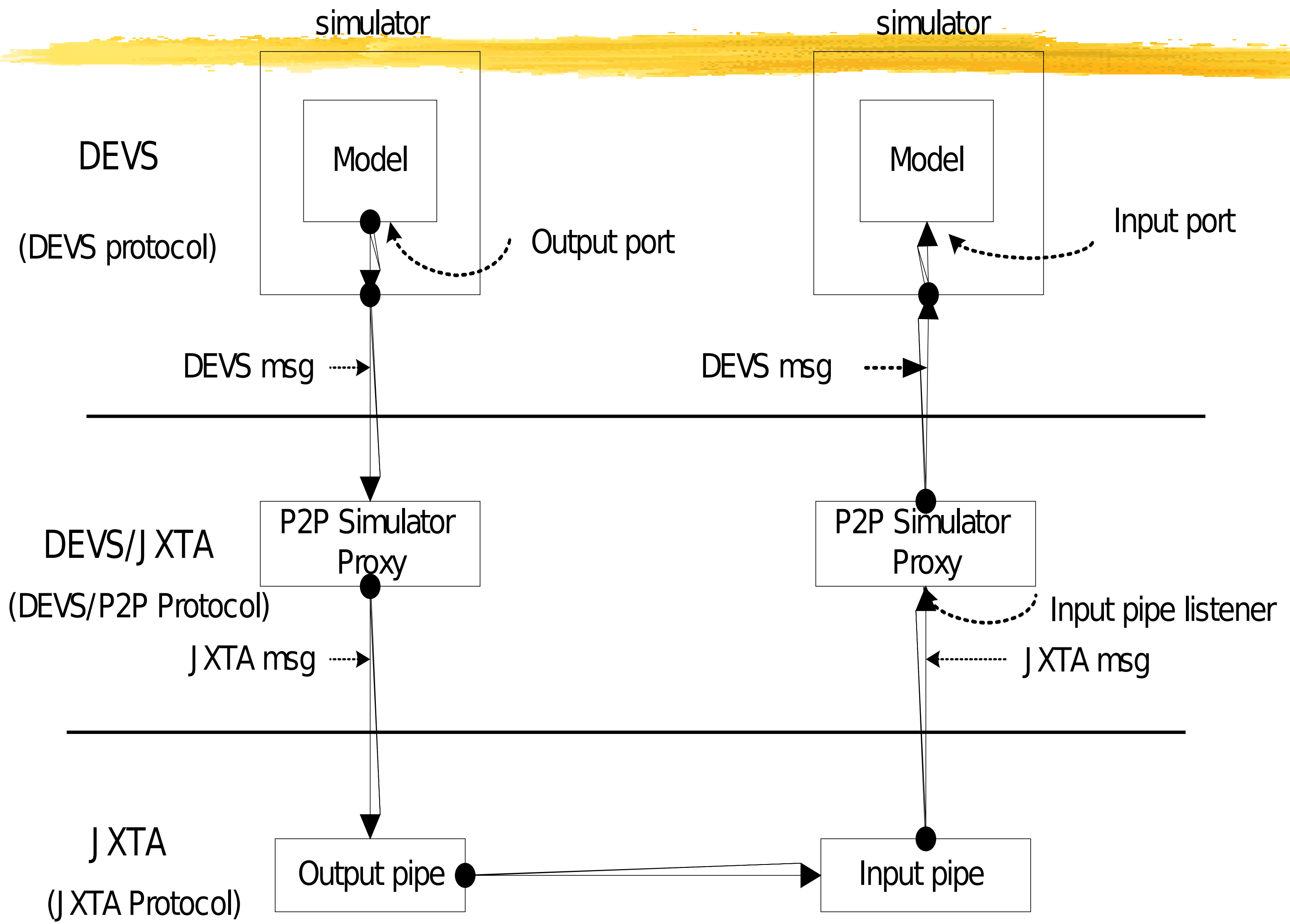
# DEVS/P2P[1]

- Use JXTA Pipe Interface as middleware to support distributed Execution of DEVS.
- Need additional layer for simulation time management.
- Prototype developed, not see application on complex and large-scale models.

# DEVS/P2P-architecture



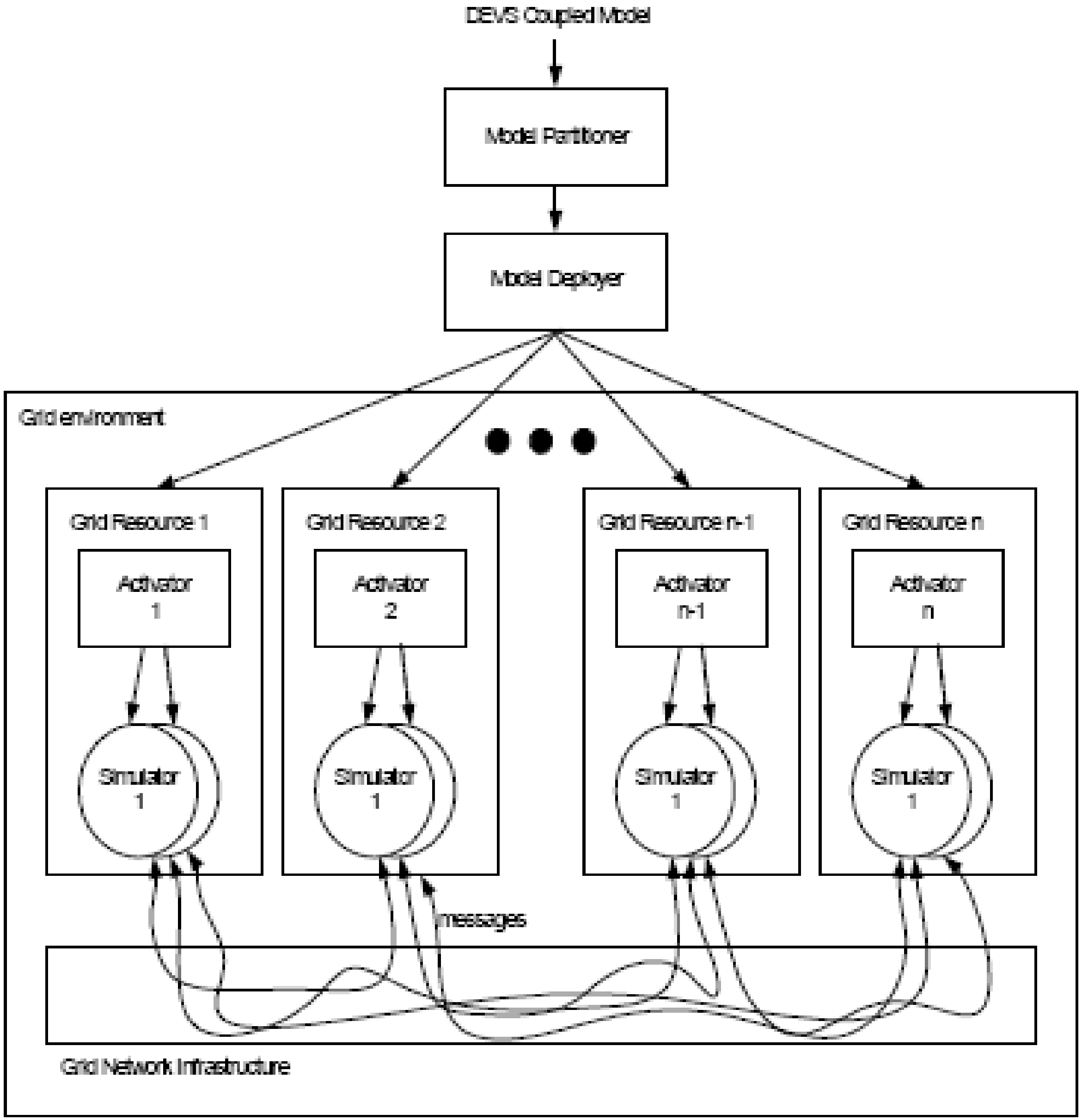
# DEVS/P2P-communication between simulators



# DEVS/Grid [2]

- Use Grid infrastructure to run DEVS in distributed fashion.
- Rely on existing Grid management framework, such as Globus.
- Not see application on solving real-world simulation models.

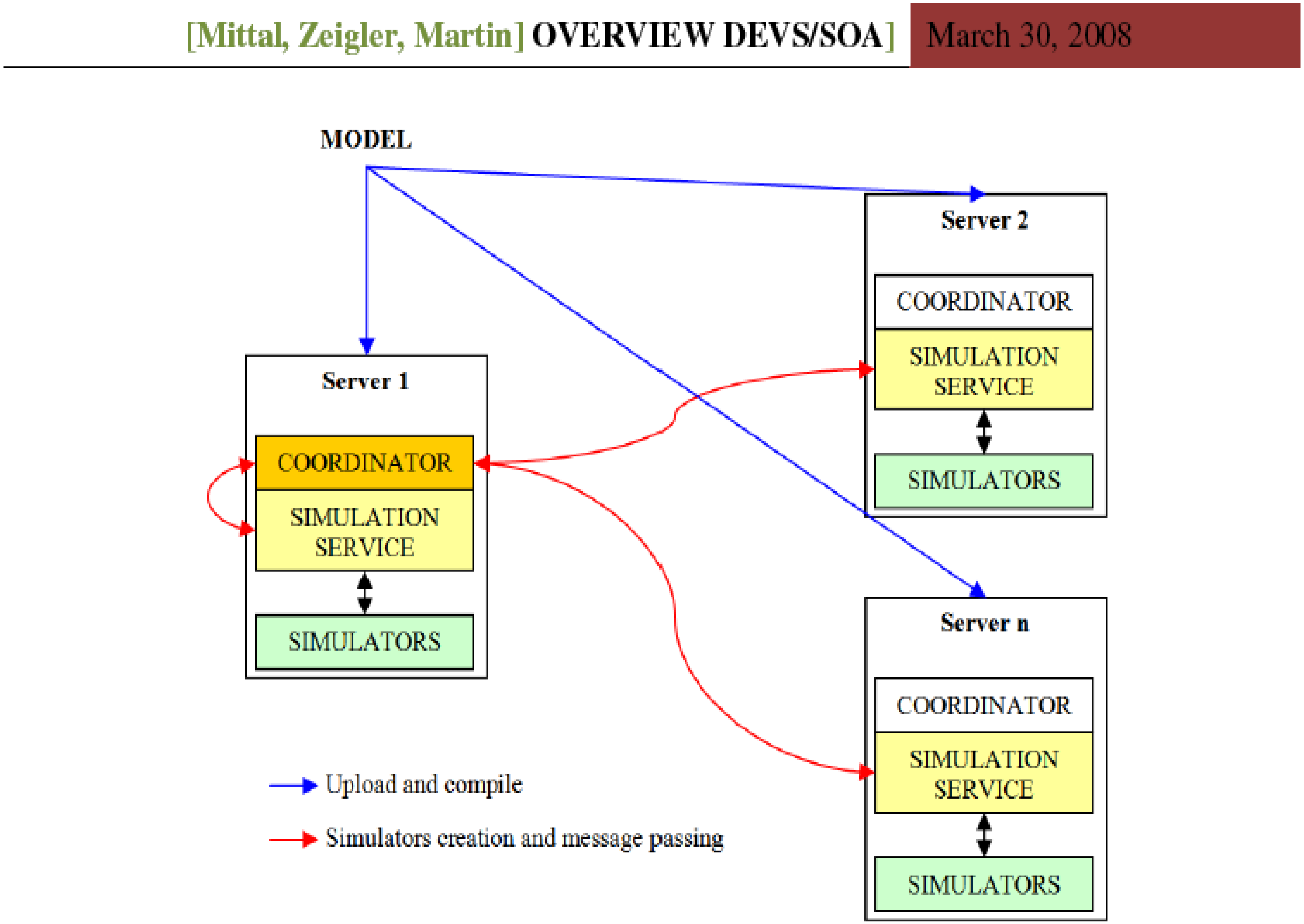
# DEVS/Grid [2]-architecture



# DEVS/SOA—recent advance [3]

- Use SOA as the basis architecture.
- Use most current web service technology.
- DEVS can be run on internet !
- Performance is the big issue?

# DEVS/SOA-architecture



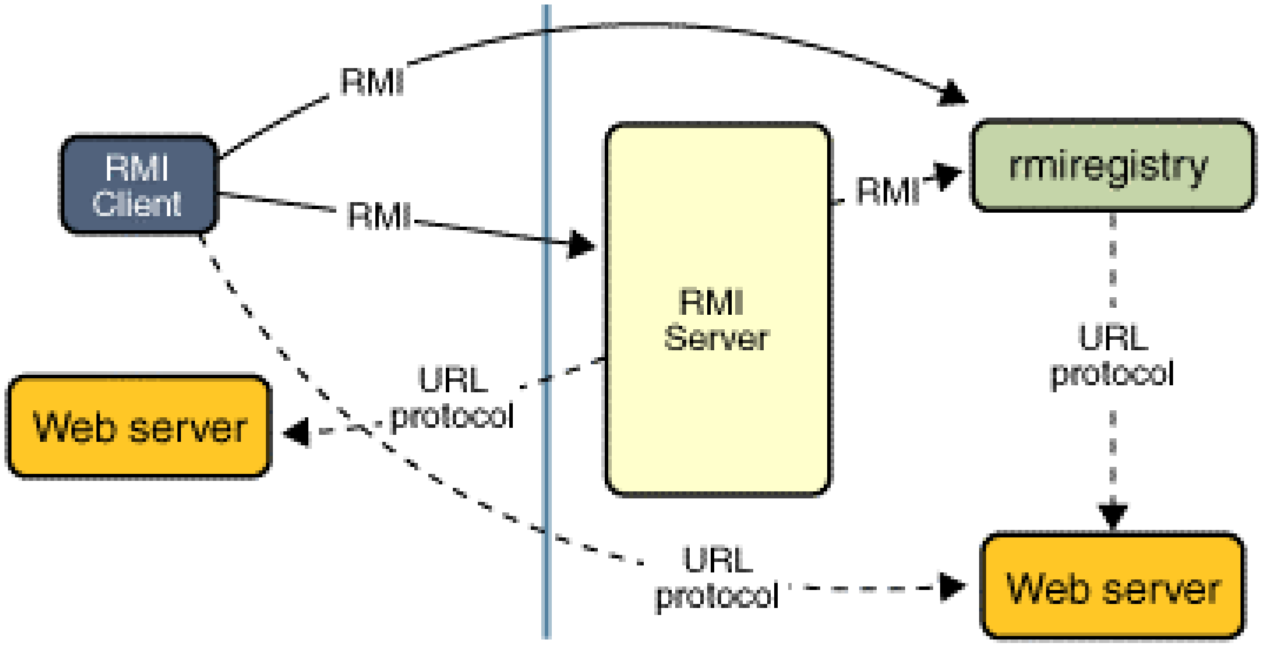
# DEVS/RMI--Motivation

- Portable distributed simulation framework
- Support dynamic re-configuration of simulation in a distributed environment
- Eliminate the model code change when mapping models to computing nodes
- Flexible to implement partition algorithm in a distributed environment
- Toward very large-scale distributed simulation.

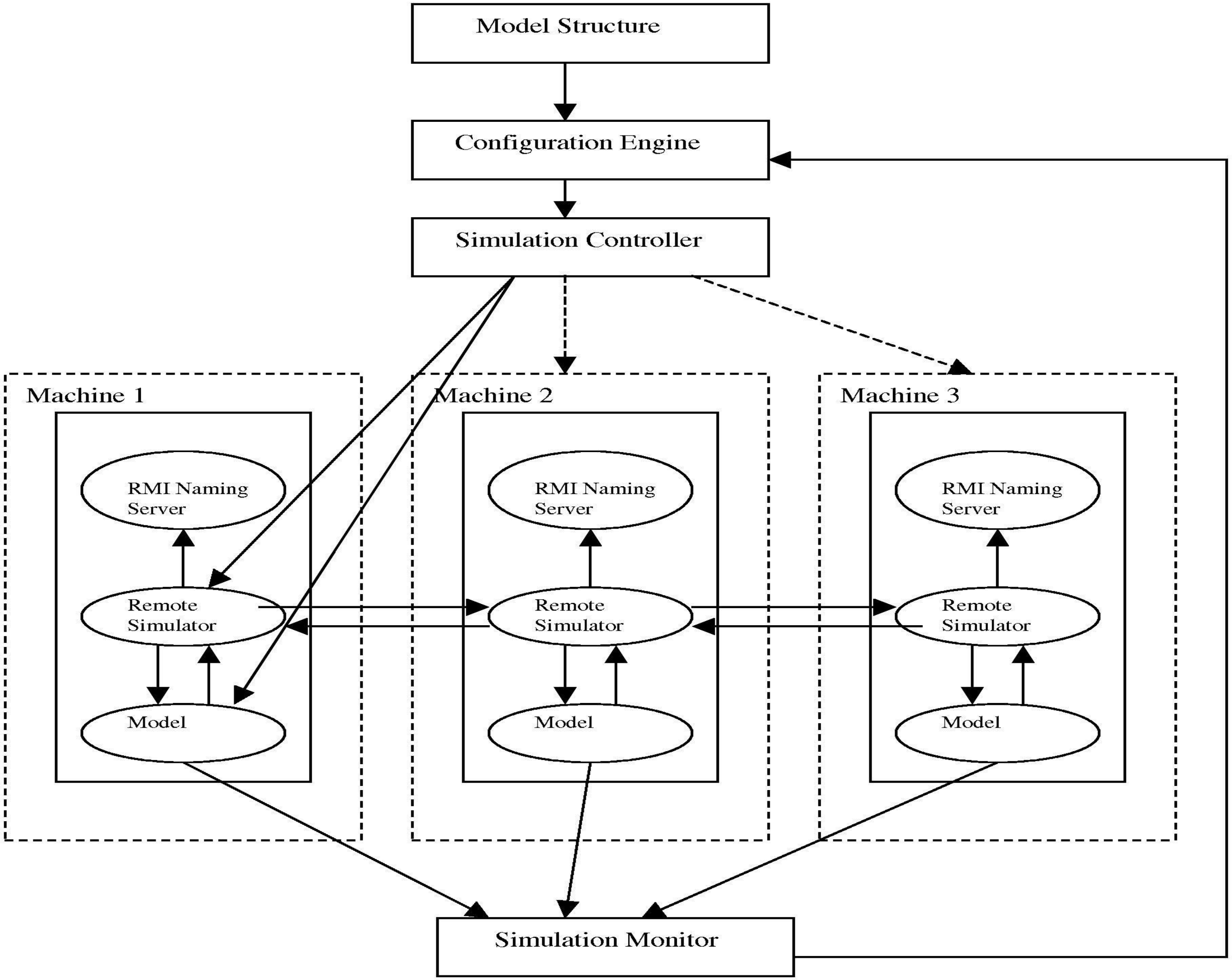
# Java RMI(1)

- Maintaining the original object architecture built for a single processor, which is important for building large-scale scalable system.
- Task or computing workload distribution is at object level, which helps on solving load-balance, fault-tolerance problems in distributed computing in an easier way.
- Make the design of highly dynamic and reconfigurable distributed framework easier; Systems integration can be performed to a higher degree.

# Java RMI(2)


RMI
RMI Client
RMI
RMI Server
RMI
rmiregistry
URL protocol
Web server
URL protocol
URL protocol
Web server

# DEVS/RMI Architecture

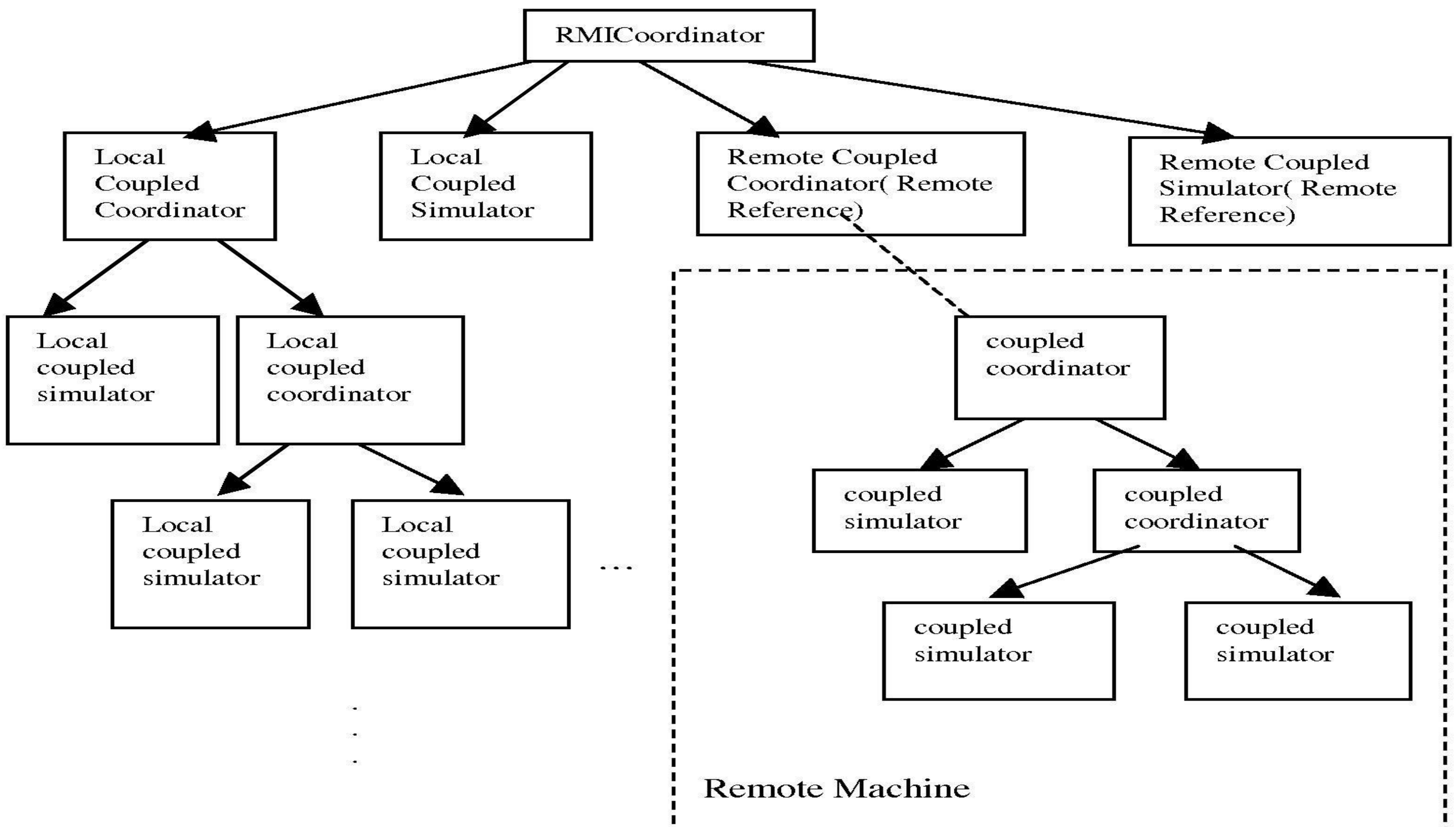



DEVS/RMI System Architecture

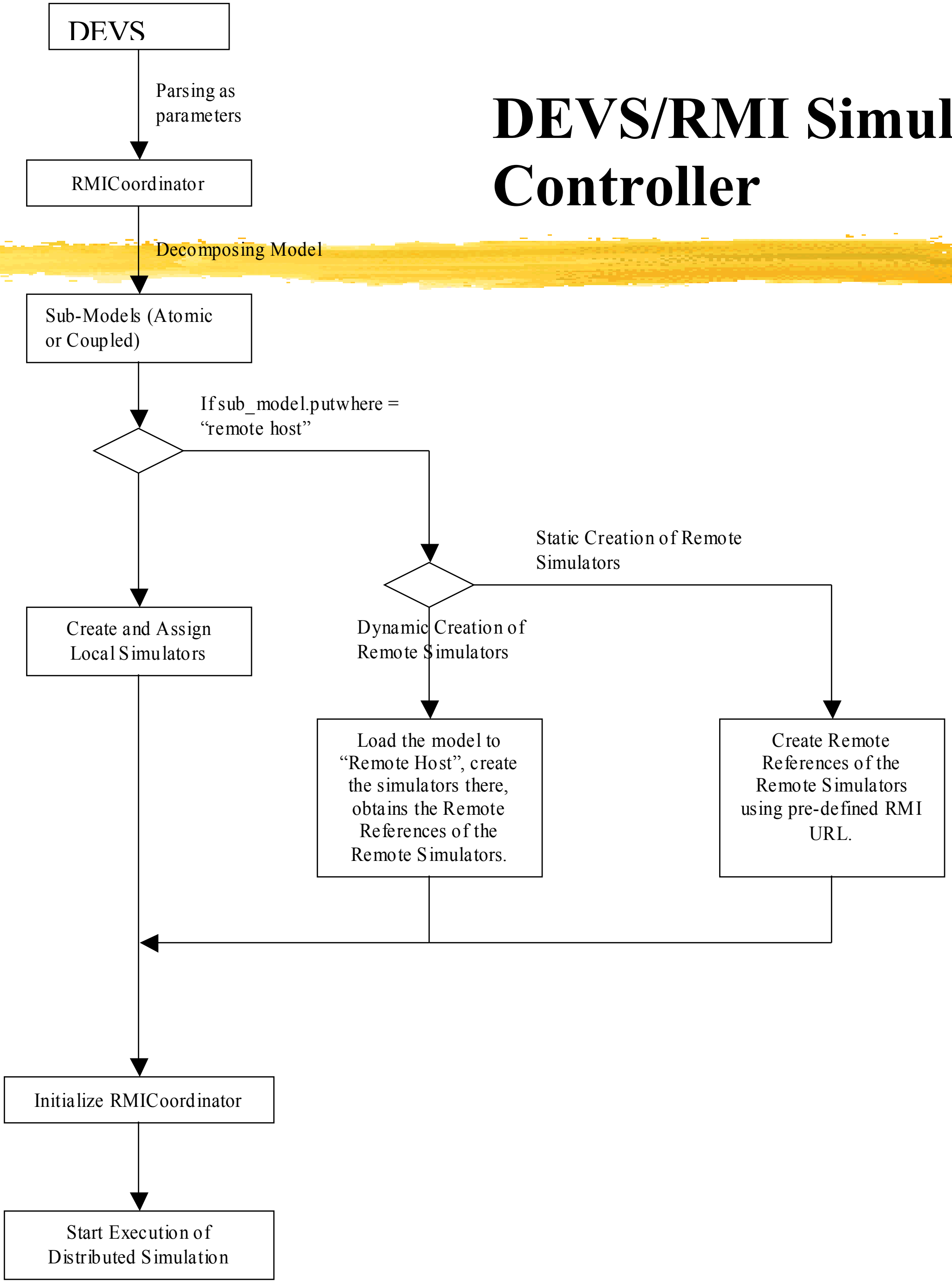
# DEVS/RMI-A Flexible Framework

- Integrate Java RMI to existing DEVS/JAVA objects framework.
- Using both local and remote simulators.
- No additional simulation time management.
- No model code change except adding a new attribute for model code to assign model to computing node.
- Flexible and dynamic re-configurable.

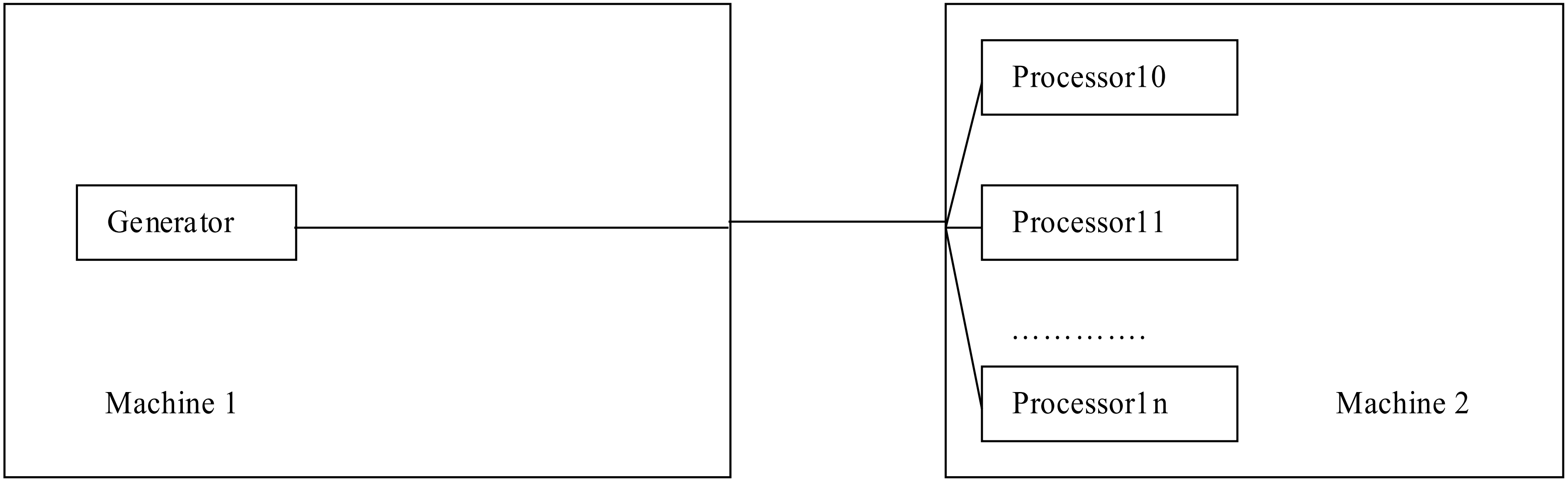
# DEVS/RMI--Simulator Relations



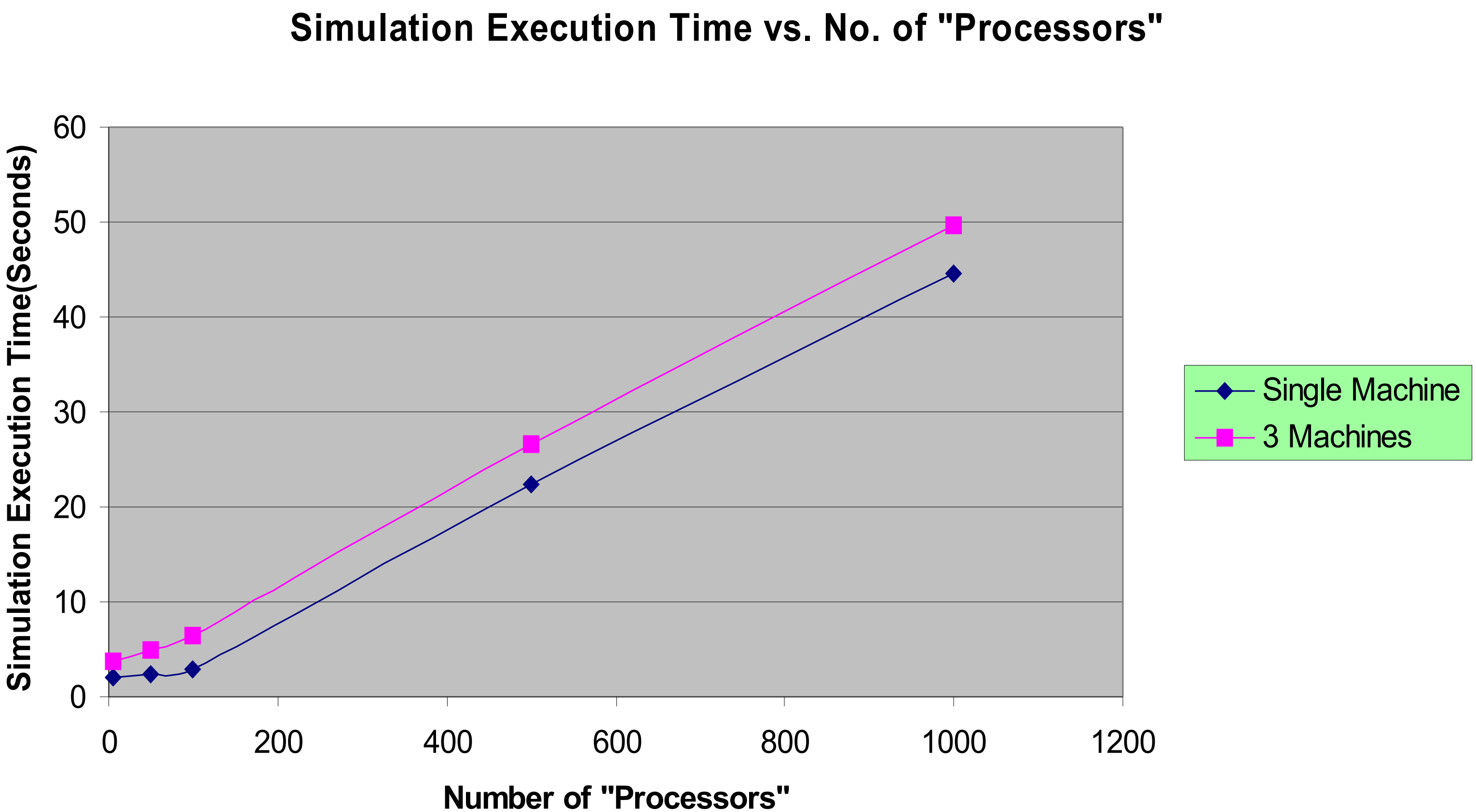
# DEVS/RMI Simulation Controller



# DEVS/RMI-- RMI Overhead Test



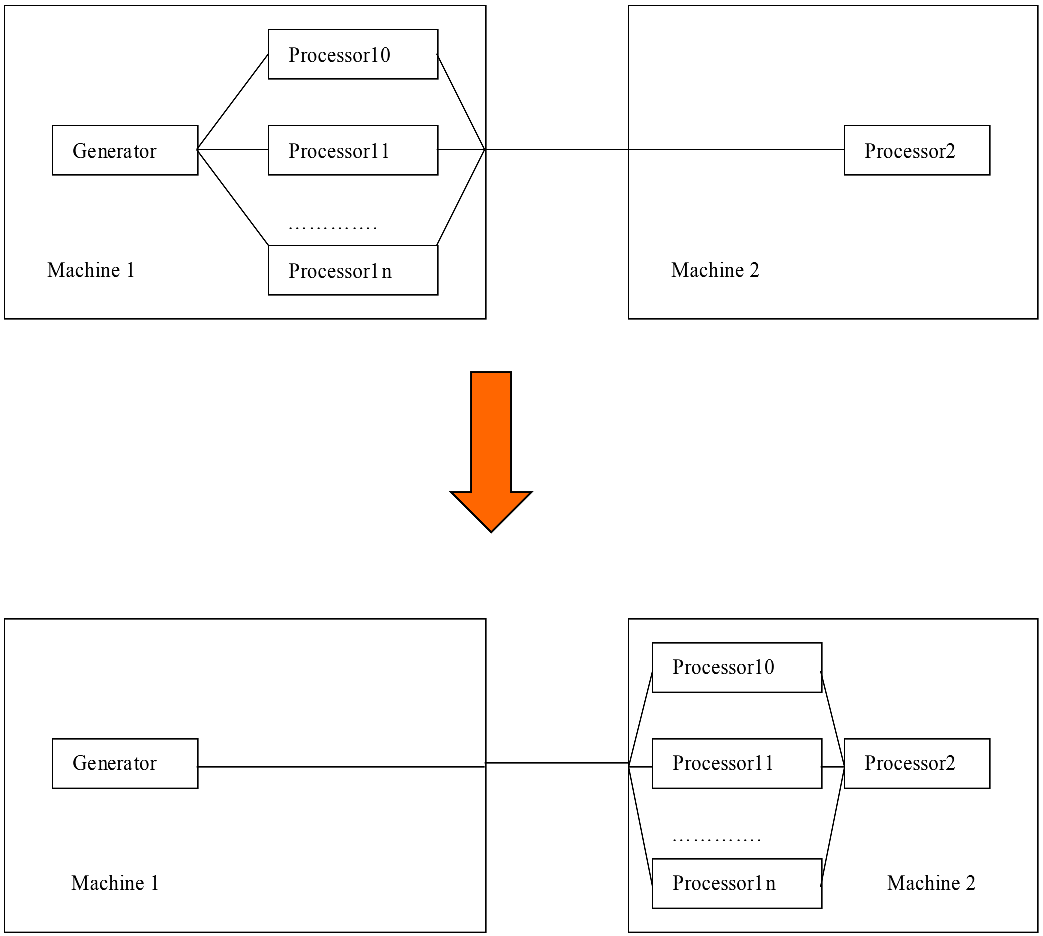
# DEVS/RMI--RMI Overhead



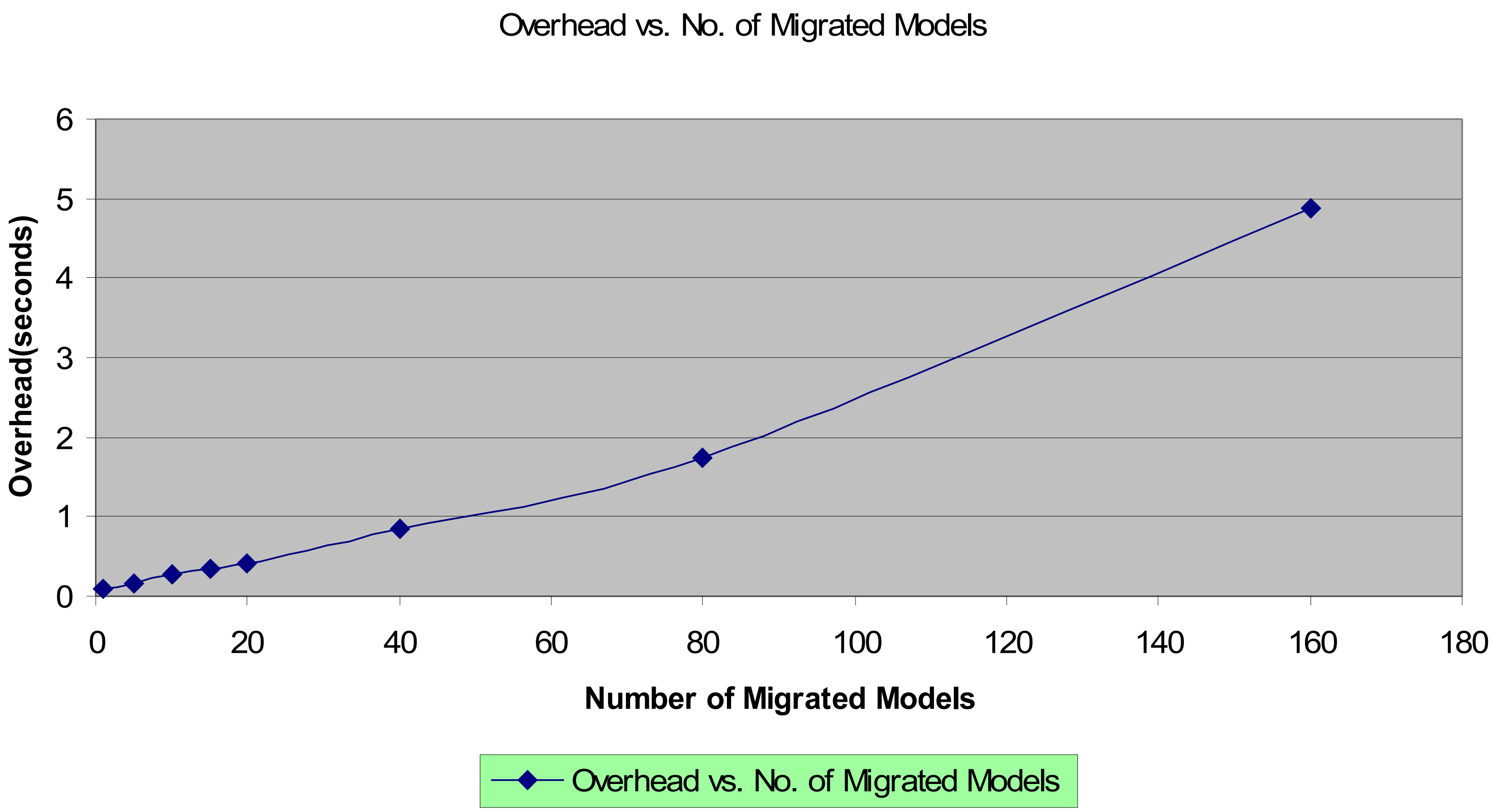
# DEVS/RMI-Reconfigurable Framework(1)

- Supports run-time model migration across machines.
- Model states are kept persistent.
- Model structure can evolve during a distributed simulation execution.
- High-level support of run-time model re-partition.

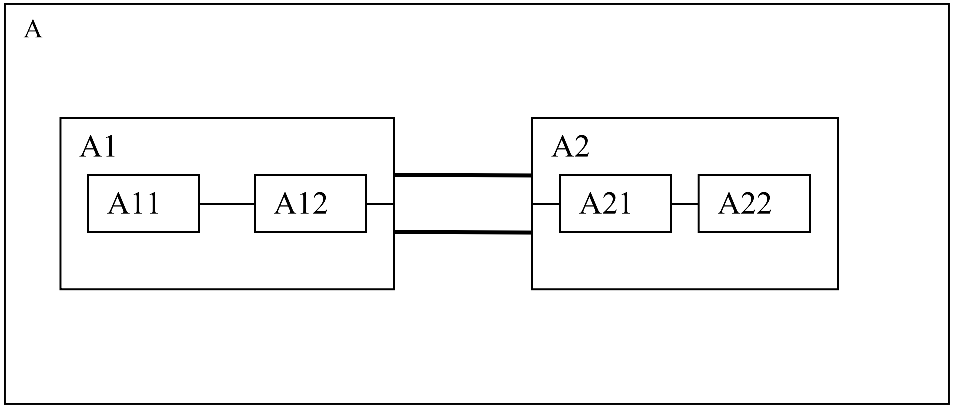
# DEVS/RMI-Reconfigurable Framework(3)



# DEVS/RMI-Reconfigurable Framework(4)

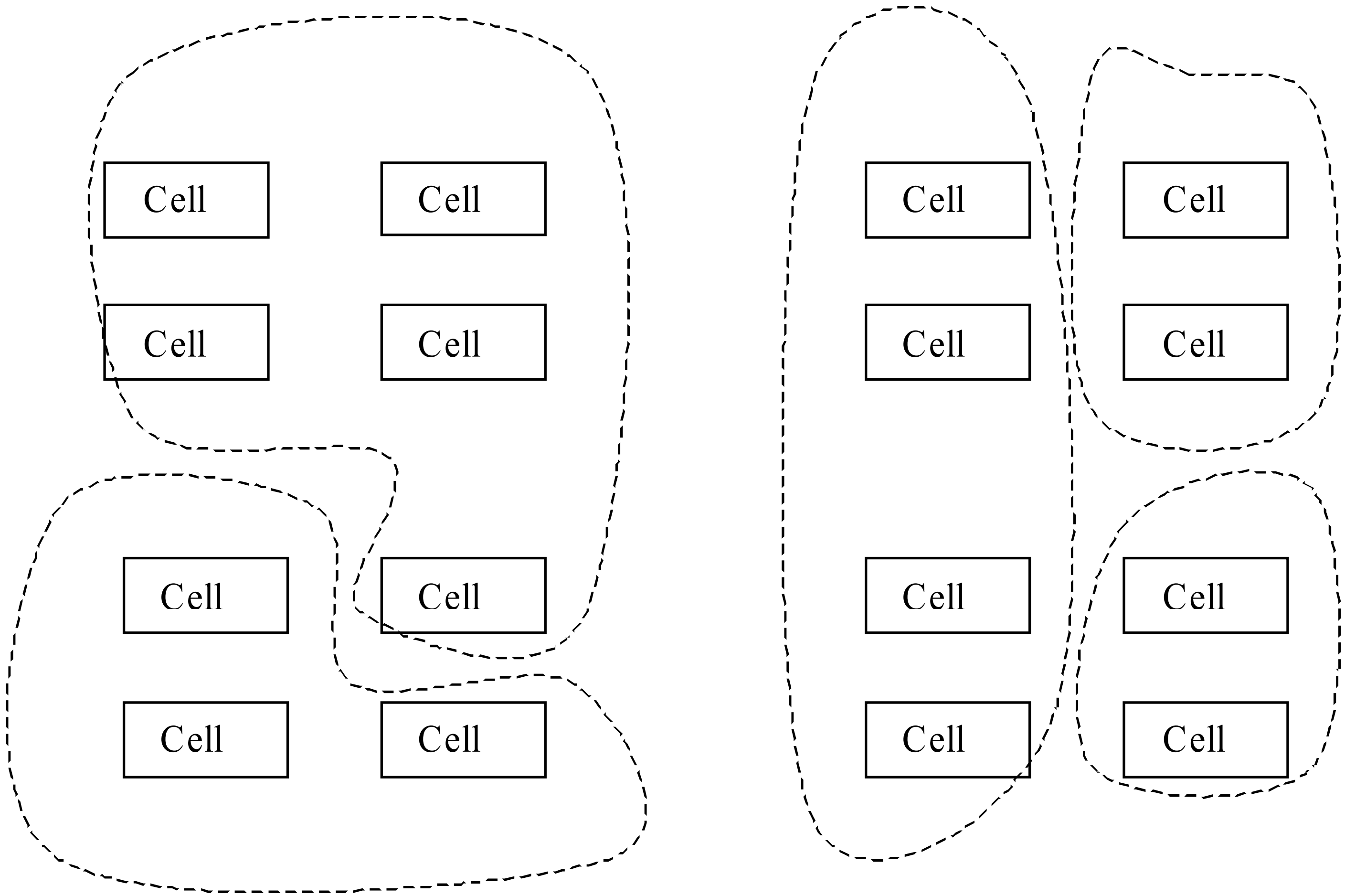


# Model Partition in DEVS/RMI(1)

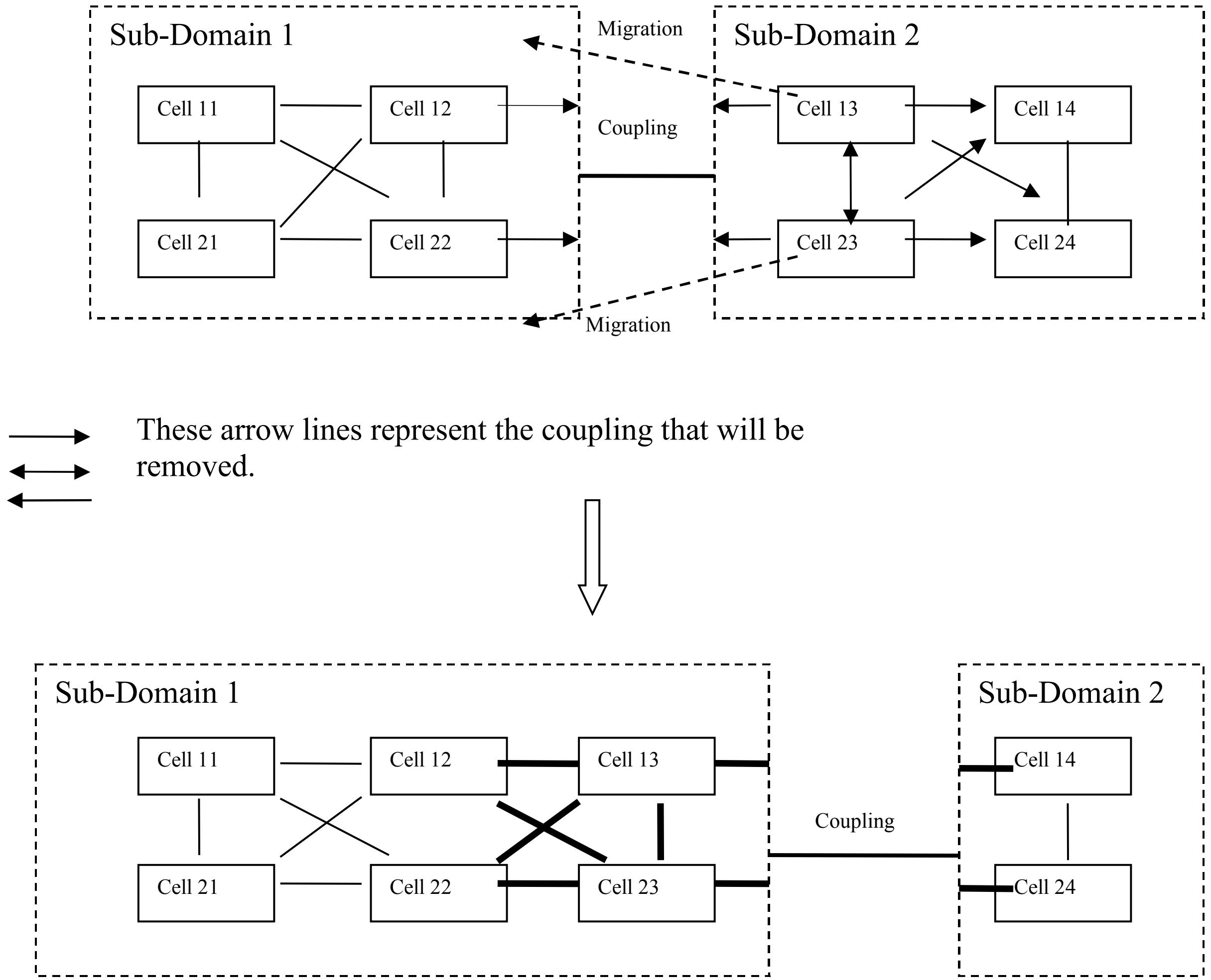



```
ViewableAtomic A1 = new generator("A1","node2");
add(A1);
ViewableAtomic A11 = new generator("A11","node3");
add(A11);
```

# Model Partition in DEVS/RMI(2)



# Model Partition in DEVS/RMI(3) --Dynamic model repartition



# Model Partition in DEVS/RMI(4)

- Increase locality whenever possible using model domain decomposition.
- Overhead incurred by dynamic repartition should be carefully evaluated.
- Load balance technique needs to be applied whenever necessary.

# DEVS/RMI on large-scale model

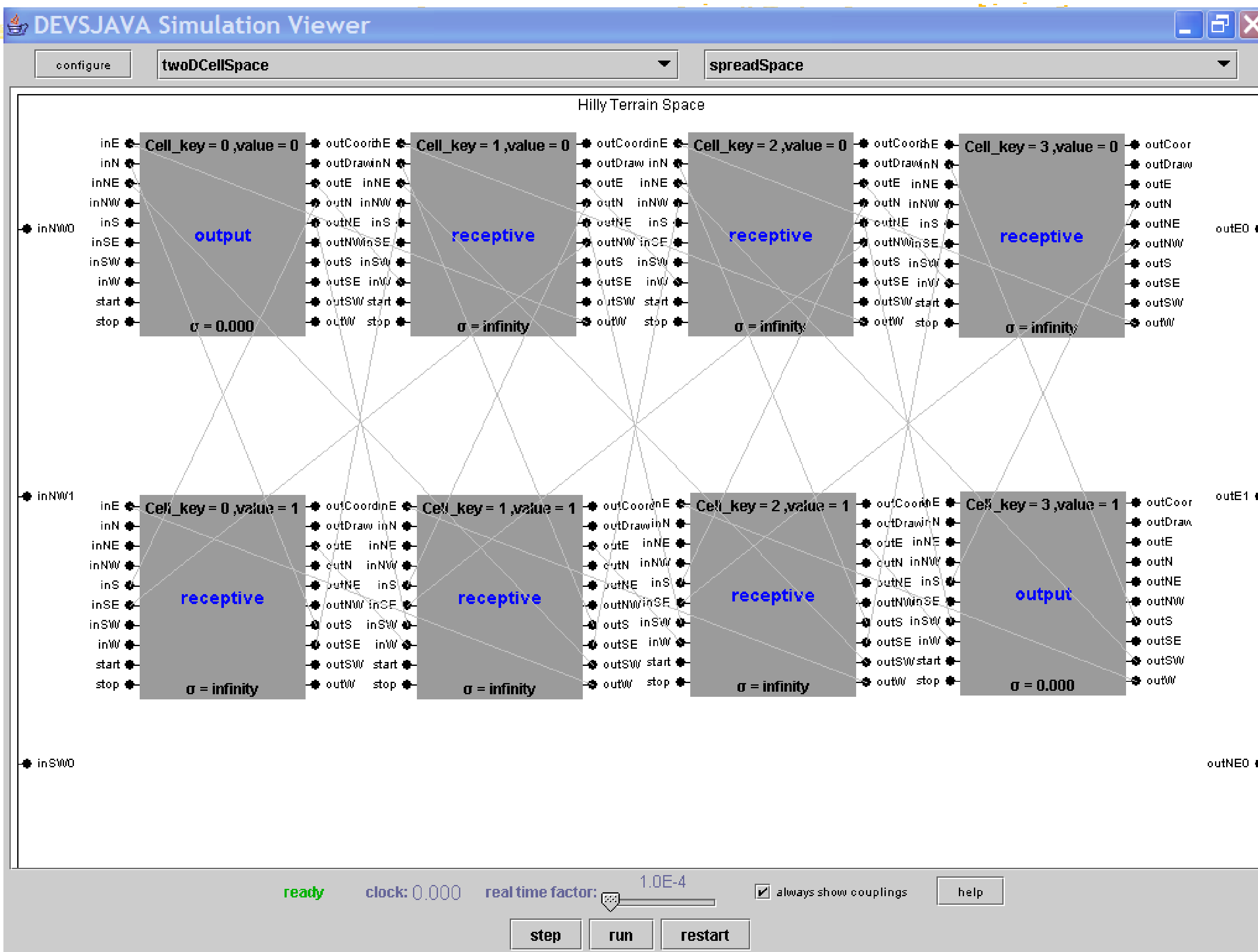
**Hilly Terrain Model**

- Measure the shortest travel time for a traveller in a 2D space with hills.
- The "direction" of traveller is determined by the gradient of hill at certain point.
- Increasing the resolution results in using larger cell space.

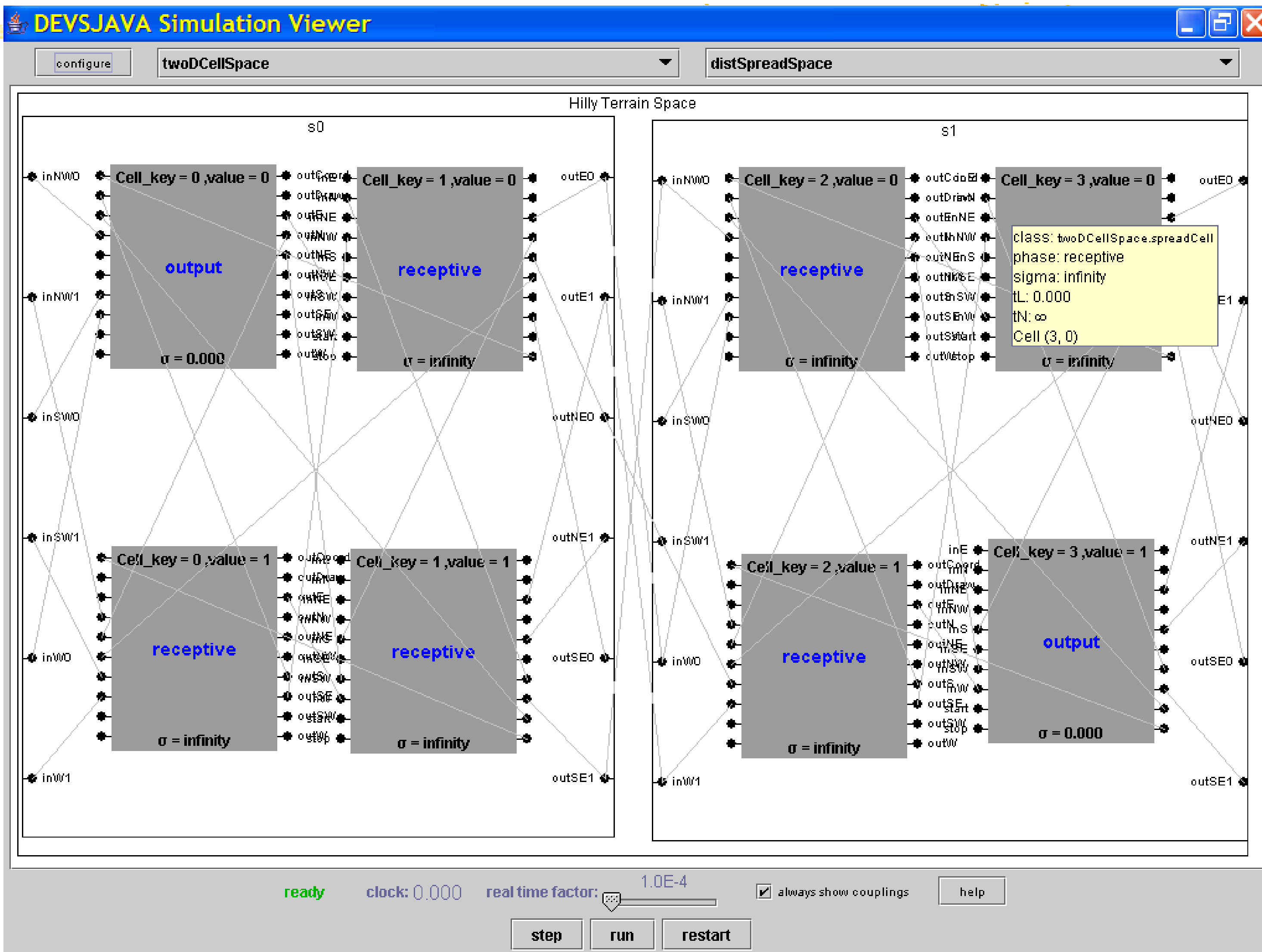
# Why use DEVS/RMI

- Express the continuous spatial model using DEVS Quantization Technique.
- 2D DEVS Cellular Space is used.
- Problems:
  - cell space should be large enough to get necessary resolution, which results in "out of memory" in a single machine.
  - Simulation execution time increases significantly when cell space increases.
- Solution: Distributed Simulation using DEVS/RMI.

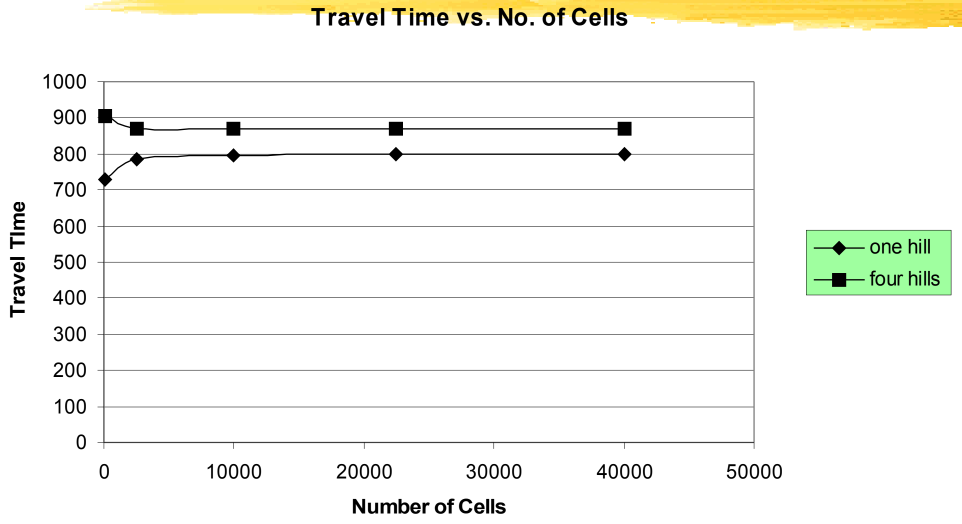
# Hilly Terrain Model in DEVS

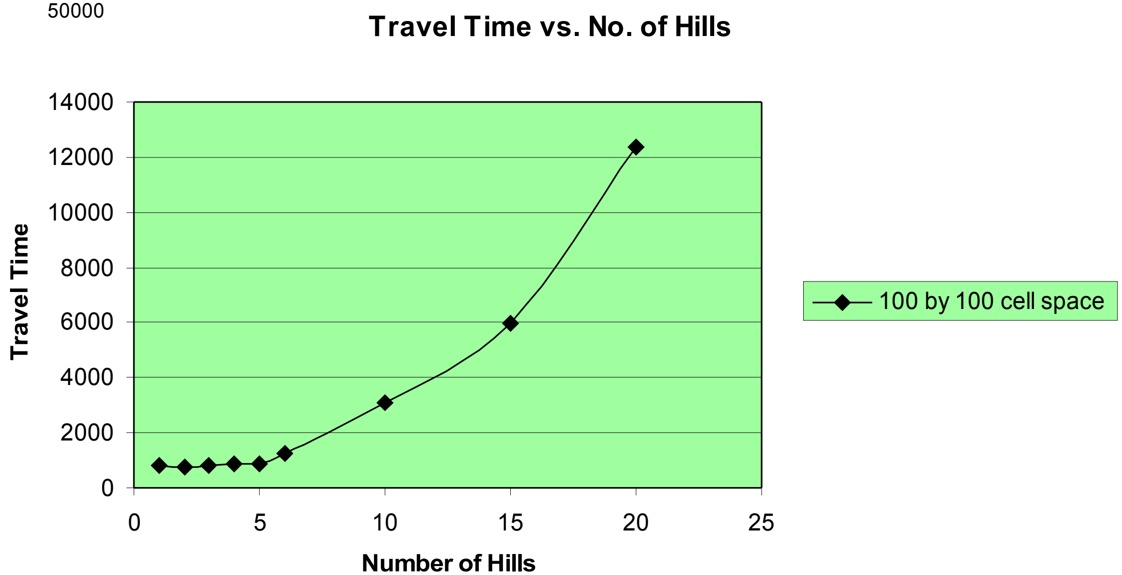


# Computation Domain Decompose for Hilly Terrain Model

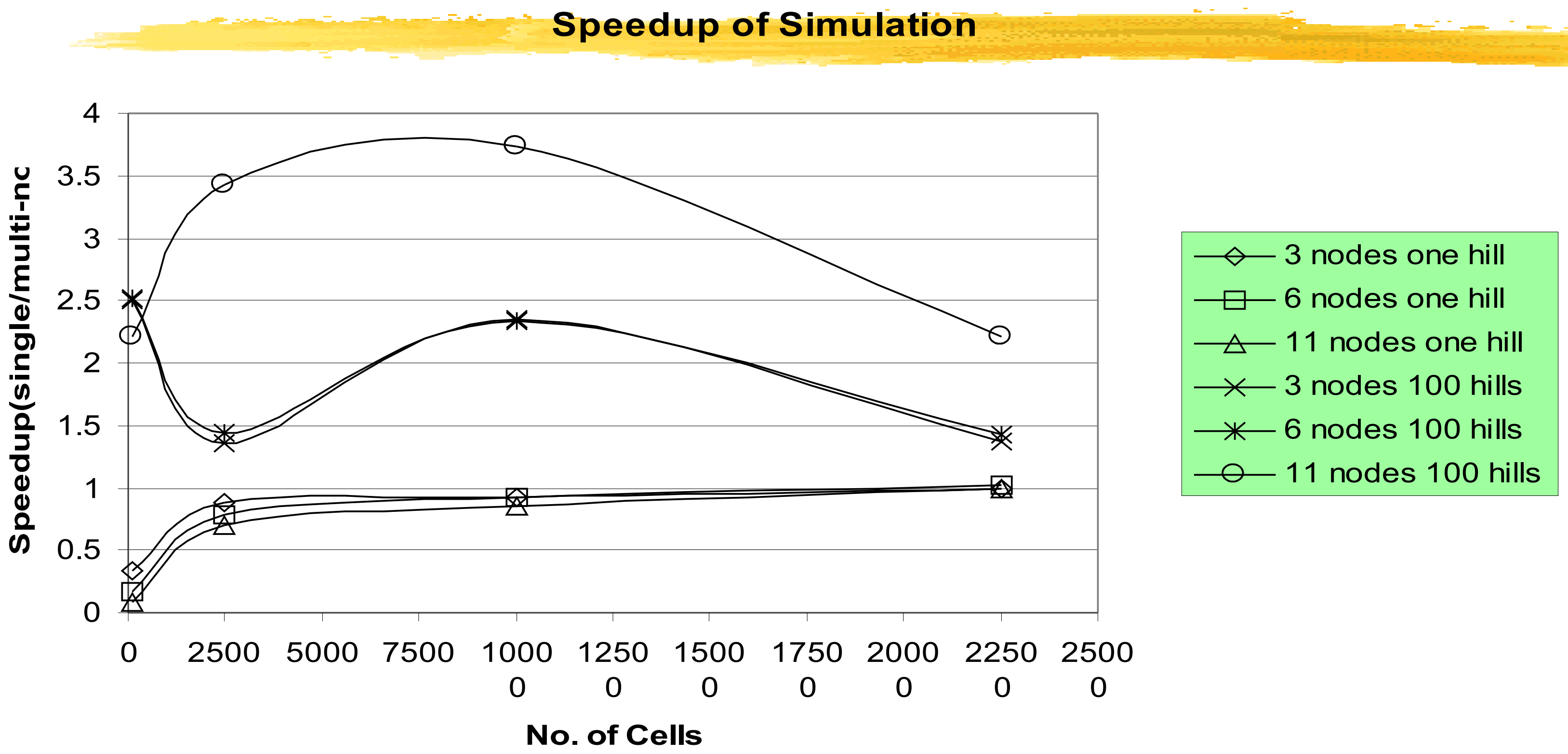


# Measure the Travel Time



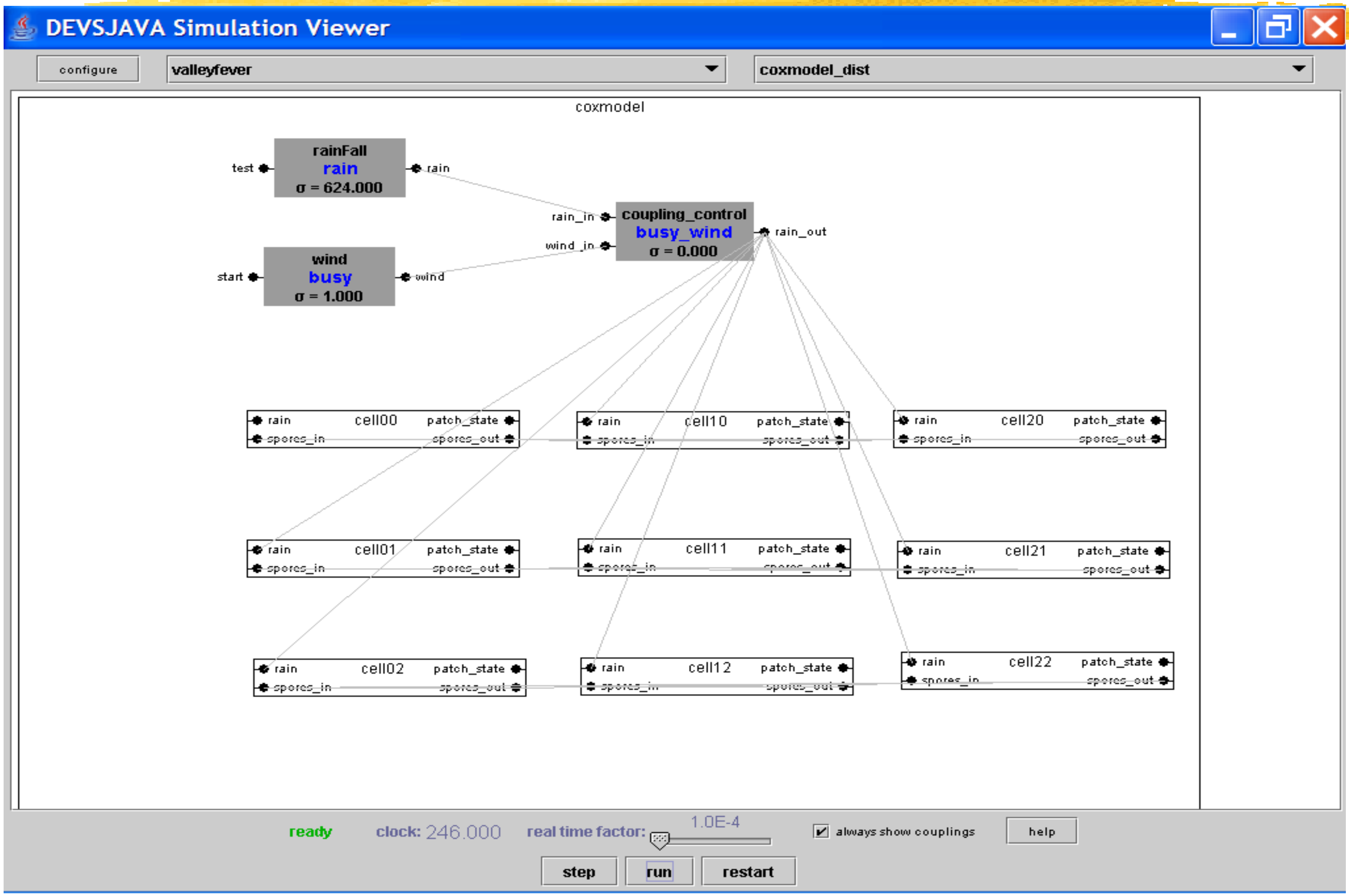
# Speedup of Distributed Simulation



# Studying Valley Fever Model

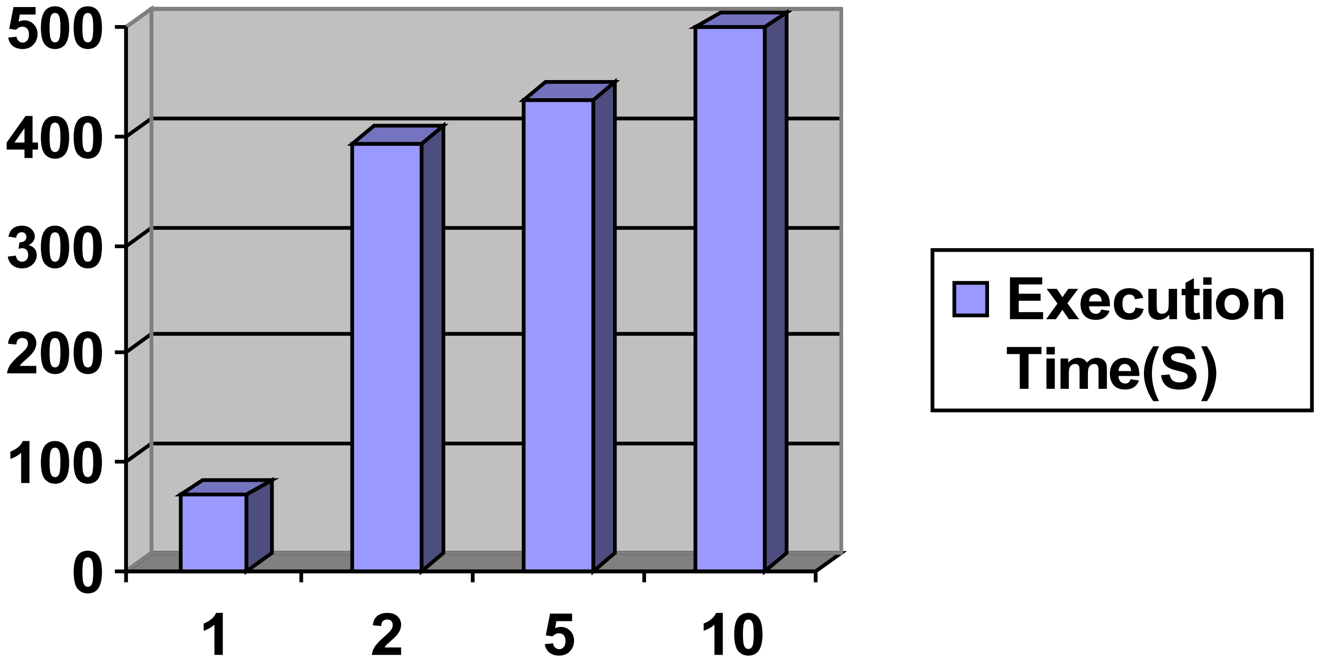
- Distributed simulation of Valley fever model, a highly dynamic 2D cell space, using DEVS/RMI
- Static model partition and "activity" based dynamic repartition are used
- Simulation execution performance is measured in terms of different computing workload
- Effects of "activity" based repartition is studied.

# Valley Fever Java Model in DEVS



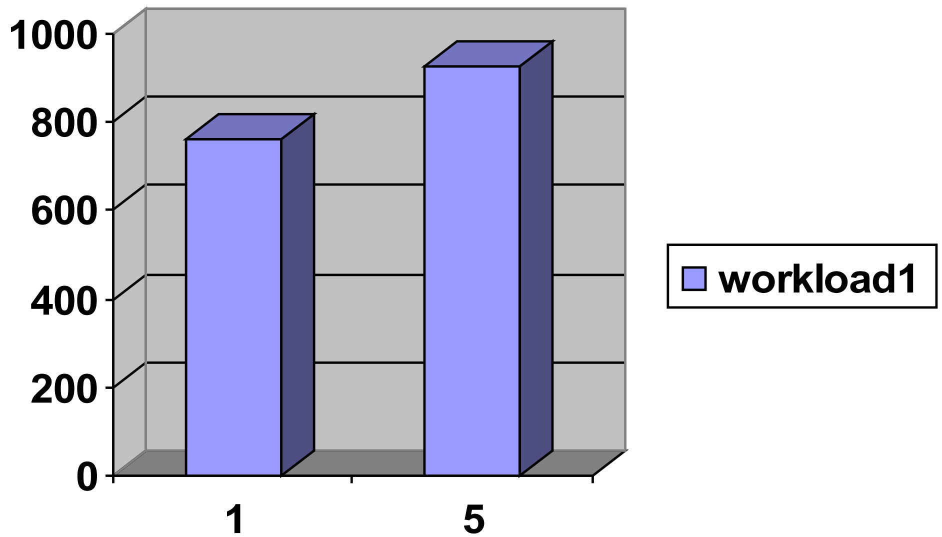

Valley fever model in Simview

# Original model distributed simulation performance

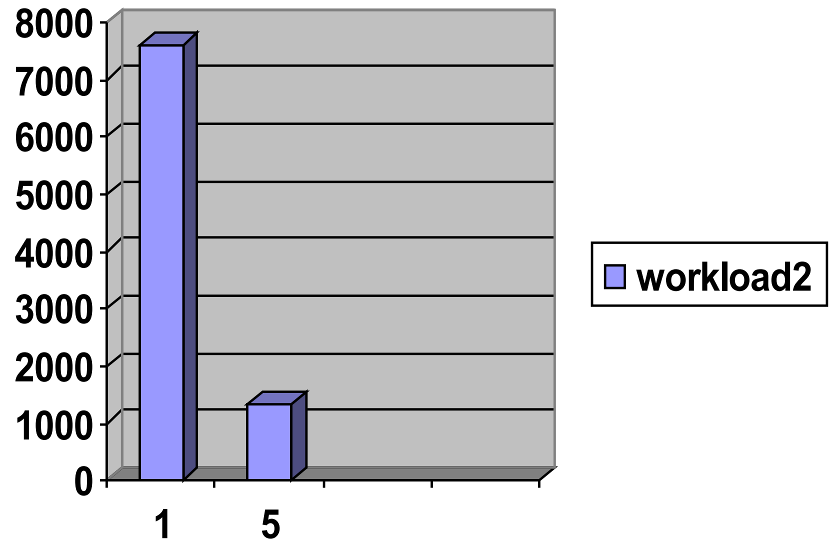



Original model simulation execution performance in DEVS/RMI

# Injecting workload




Injecting workload to partitioned cells (a sum of 1 to 100)




Injecting workload to partitioned cells (a sum of 1 to 150)

# Dynamic repartition using "activity"

- "activity" metric is measured by counting the internal transitions of each individual cell.
- "activity" metric is used to repartition the model dynamically to achieve better load balance.
- High "activity" cells are assigned more computing power.

# Valley Fever Model-using activity

| Using 5 computing nodes including 1 head node. | Static Blind Partition not considering model activities | Dynamic reconfiguration using "activity" | Performance increase by percentage |
|---|---|---|---|
| 4 by 4 cells with 400 simulation steps | 28.124s | 27.566s | 1.98% |
| 4 by 4 cells with 2000 simulation steps | 113.977s | 114.968s | -0.87% |
| 8 by 8 cells with 400 simulation steps | 256.49s | 248.644s | 3.06% |
| 8 by 8 cells with 2000 simulation steps | 1238.479s | 1216.97s | 1.73% |

| Using 9 computing nodes including 1 head node. | Static Blind Partition not considering model activities | Dynamic reconfiguration using "activity" | Performance increase by percentage |
|---|---|---|---|
| 4 by 4 cells with 2000 simulation steps | 134.74s | 110.49s | 18% |
| 8 by 8 cells with 2000 simulation steps | 1348.17s | 1199.87s | 11% |

# Performance Issues

- Distributed Simulation performance of DEVS/RMI closely relates to the computation and RMI communication workload partitions.
- Load balance is a key factor.
- Locality should be increased whenever possible.

# Advantages of DEVS/RMI

- DEVS/RMI provides an flexible and easy-to-use reconfigurable distributed simulation framework.
- Refactoring a distributed simulation becomes easier.
- Support run-time model structure evolution in a distributed environment.
- Achieves significant speedup when dealing with large-scale model.

# DEVS in the near future

- SOA based architecture.
- Running on P2P network
- Towards to distributed execution.
- Towards running on grid.
- Keep its role as a formalized modeling framework.

# References:


- 1.Saehoon Cheon, Chungman Seo, Sunwoo Park, Bernard P. Zeigler, “Design and Implementation of Distributed DEVS Simulation in a Peer to Peer Network System”, 2004 Military, Government, and Aerospace Simulation.
- 2. Chungman Seo, Sunwoo Park, Byounguk Kim, Saehoon Cheon, Bernard P. Zeigler, “Implementation of Distributed High-performance DEVS Simulation Framework in the Grid Computing Environment”, 2004 High Peformance Computing Symposium.
- 3. Mittal, S., Risco-Martin, J.L., Zeigler, B.P., "DEVS-Based Simulation Web Services for Net-centric T&E", Summer Computer Simulation Conference SCSC'07, July, 2007.
- 4. **Ming Zhang, B.P. Zeigler, P. Hammonds, "DEVS/RMI-An Auto-Adaptive and Reconfigurable Distributed Simulation Environment for Engineering Studies", International Test & Evaluation Association (ITEA) Journal of Test and Evaluation, March/April 2006, Volume 27, Number 1, Page 49-60.**


* http://www.acims.arizona.edu/EDUCATION/ECE575Fall03/Note/

Thank You!

Questions?

Email: mizhang@site.uottawa.ca

Google: **ACIMS**