

An Adaptive Load Balancing Middleware for Distributed Simulation

Gabriele D'Angelo

joint work with

Luciano Bononi

Michele Bracuto

Lorenzo Donatiello



University of Bologna
Department of Computer Science

WOMP 2006 – Sorrento (NA)

Presentation outline

- Basic assumptions and goals
- Parallel and Distributed Simulation (PADS) main problems
- A migration-based approach to improve the simulation speed
- The proposed adaptive load-balancing mechanism
- Testbed: Ad Hoc network model
- Execution architecture definition
- Simulation results: dynamic reallocation and simulation speed
- Conclusions and future work

Basic assumptions and goal

- The simulation is a useful technique to support the design and the performance evaluation of complex systems
- The systems now considered of interest are composed by many highly dynamic entities, with unpredictable communication patterns
- The simulation of such kind of systems is possible only aggregating together many execution units: **Parallel And Distributed Simulation (PADS)**
- Our goal is to **increase the simulation speed**, that is reduce the **Wall Clock Time (WCT)** required to complete the simulation runs

Parallel and Distributed Simulation (PADS)

■ PROS:

- Clustering together many Physical Execution Units (PEUs) it is possible to reduce the WCT required to complete a simulation
- Aggregating resources (i.e. RAM) it is possible to represent very complex systems

■ CONS:

- A distributed simulation requires a large amount of **communication** and **synchronization** to obtain correct results
- The simulation speed depends on many factors, in example: heterogeneous CPUs and resources, background load

The GAIA migration-based framework

- A distributed simulation can be seen as a set of Logical Processes (LPs). Each LP is run by a possibly different PEU and takes care to manage the evolution of a set of Simulated Model Entities (SMEs)
- What is a good scheme to allocate the SMEs on the LPs/PEUs?
- We have demonstrated that the **static allocation schemes are in most cases sub-optimal**, and that a **migration-based approach of the simulated model entities** (GAIA) can
 - Reduce the amount of communication and synchronization, clustering together the highly interacting simulated entities within the same LP/PEU
 - Reduce the WCT, therefore increase the simulation speed-up

PADS execution architectures

- **Dedicated clusters** composed by homogeneous units are **costly** and often **underutilized**
- For the same reason, **shared** clusters composed by available Commercial-Off-the-Shelf (COTS) hardware are preferable to dedicated systems
- The simulation performance is highly influenced by the CPU load in background
- In a shared cluster it is impossible to predict a good **static allocation scheme** for the simulated entities: the background load is **unpredictable** and the CPU can usually highly **heterogeneous**
- An **adaptive load balancing mechanism** could improve the resources utilization and therefore the simulation speed

The GAIA+ adaptive load balancing middleware

- GAIA+ is an evolution of the migration-based mechanism GAIA
- It is composed by two cooperative parts, both based on the reallocation of simulated entities:
 - **The heuristic migration policy:** to adapt and reduce the communication and synchronization needs
 - **The heuristic load-balancing policy:** the overloaded (*and therefore "slow"*) PEUs can migrate some of the managed model entities to unloaded PEUs
- The simulation is totally distributed and therefore there is no point of centralization. **Slow** and **Fast** are attributes that have to be observed locally, with only a partial knowledge of the whole system and influenced by the networks delays

Performance evaluation: Ad-Hoc wireless network model

The GAIA mechanism outperforms the static distributed simulation approach. This performance evaluation will compare the GAIA and the GAIA+ mechanisms

Ad-Hoc wireless network model definition:

- 9000 **Simulated Mobile Hosts** (SMHs) over a flat topology
- Mobility model:
 - Random Waypoint Model (RWP)
 - uncorrelated SMHs' mobility
- Traffic model:
 - ping messages (CBR) by every SMH to all neighbors within the wireless communication range (250 m)
- Propagation model
 - open space (neighbor-SMHs within detection range)

Ad-Hoc network model characterization

Computation and communication issues:

- The **computation** required for each SMH per time-step is in the order of **$O(\#SMH^2)$** : to determine the neighbor set
- The communication required among SMHs is in the order of **$O(K \cdot \#SMH)$** per time-step, with K defined as a constant value based on SMHs density (assumed as constant)

Testbed execution architecture

Distributed simulation, execution architecture:

- 3 heterogeneous PEU:
 - **2** - Intel **Dual** Xeon Pentium IV 2800 MHz, with **3 GB RAM** and **4 GB RAM**, Debian GNU/Linux OS with kernel version 2.6.x
 - **1** - Intel **Quad** Xeon Pentium IV 1500 MHz, with **1 GB RAM**, Debian GNU/Linux OS with kernel version 2.6.x
- PEUs are interconnected by a Gigabit Ethernet LAN
- 3 LPs = 1 LP for each PEU
- Conservative time-stepped simulation: 2000 time-steps

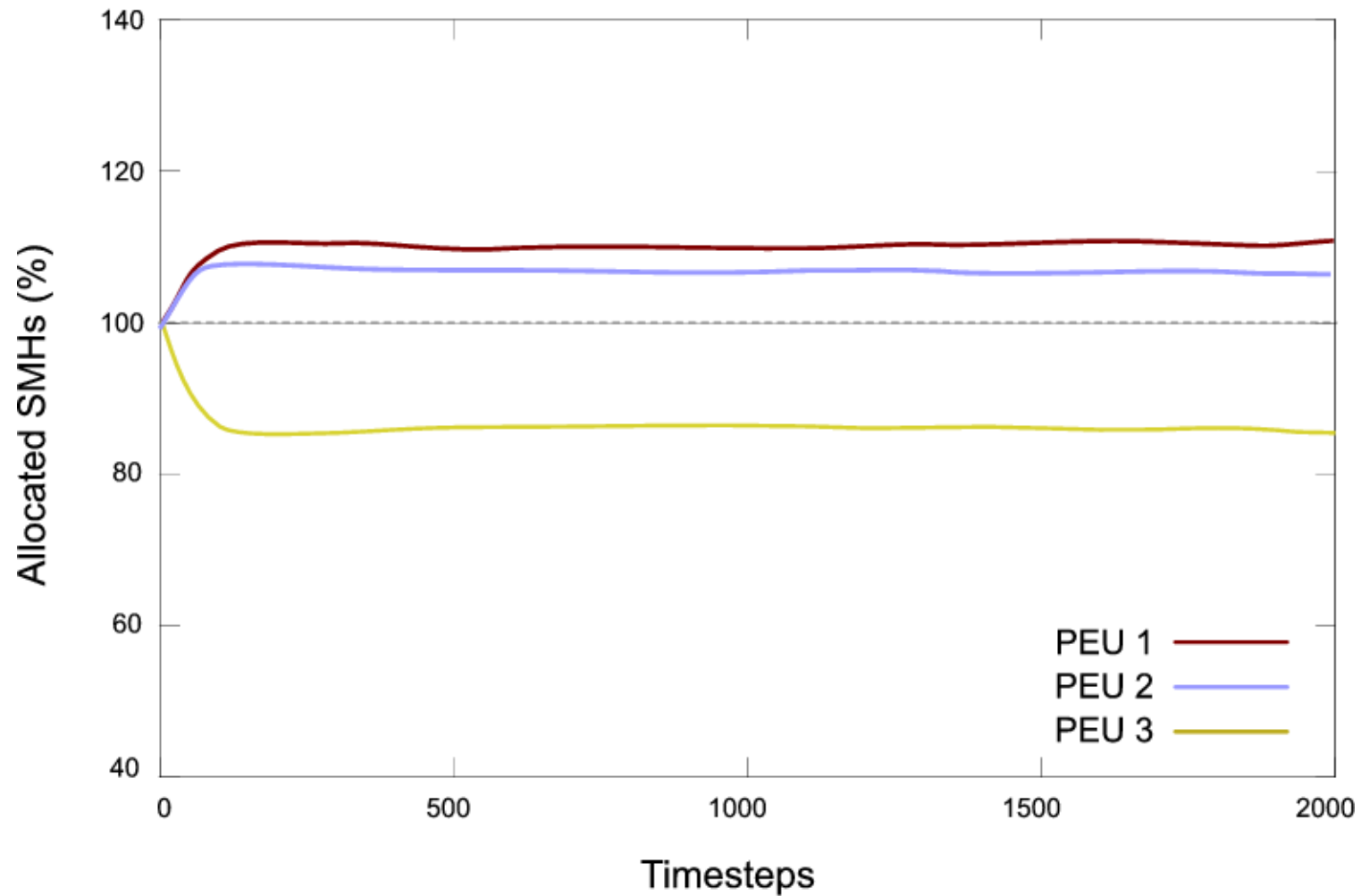
Performance evaluation: three different scenarios

The performance evaluation of the GAIA+ mechanism has involved three different scenarios:

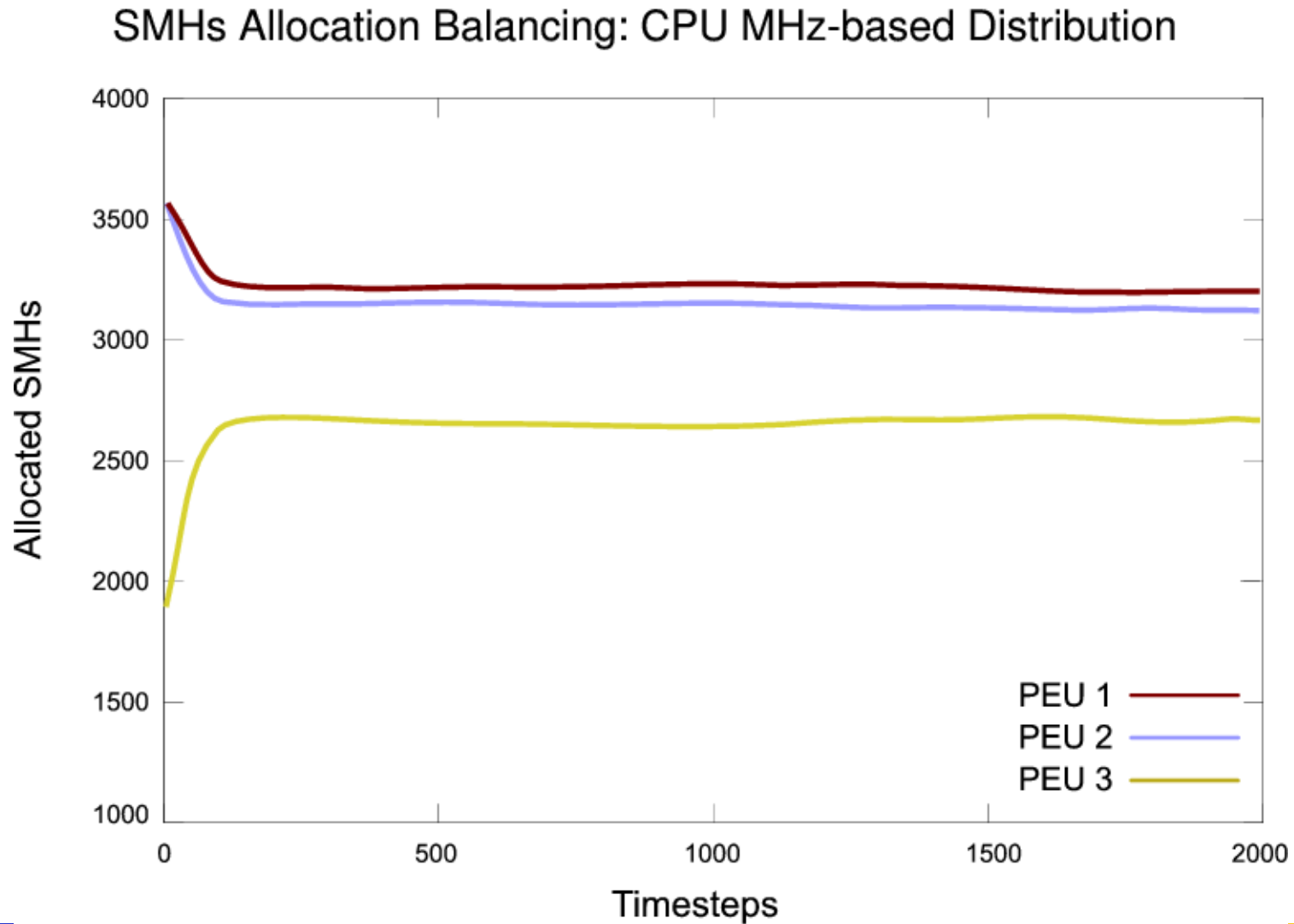
- a) The CPU are considered as homogeneous: **initially** each PEU allocates **the same number** of SMHs
- b) The **initial** allocation is based on the **nominal performance of the CPUs** (as expressed in MHz)
- c) A **synthetic background load** (in form of a sinusoidal wave) is injected in part of the simulation cluster

Performance evaluation: Equal Distribution (scenario a)

SMHs Allocation Balancing: Equal Distribution

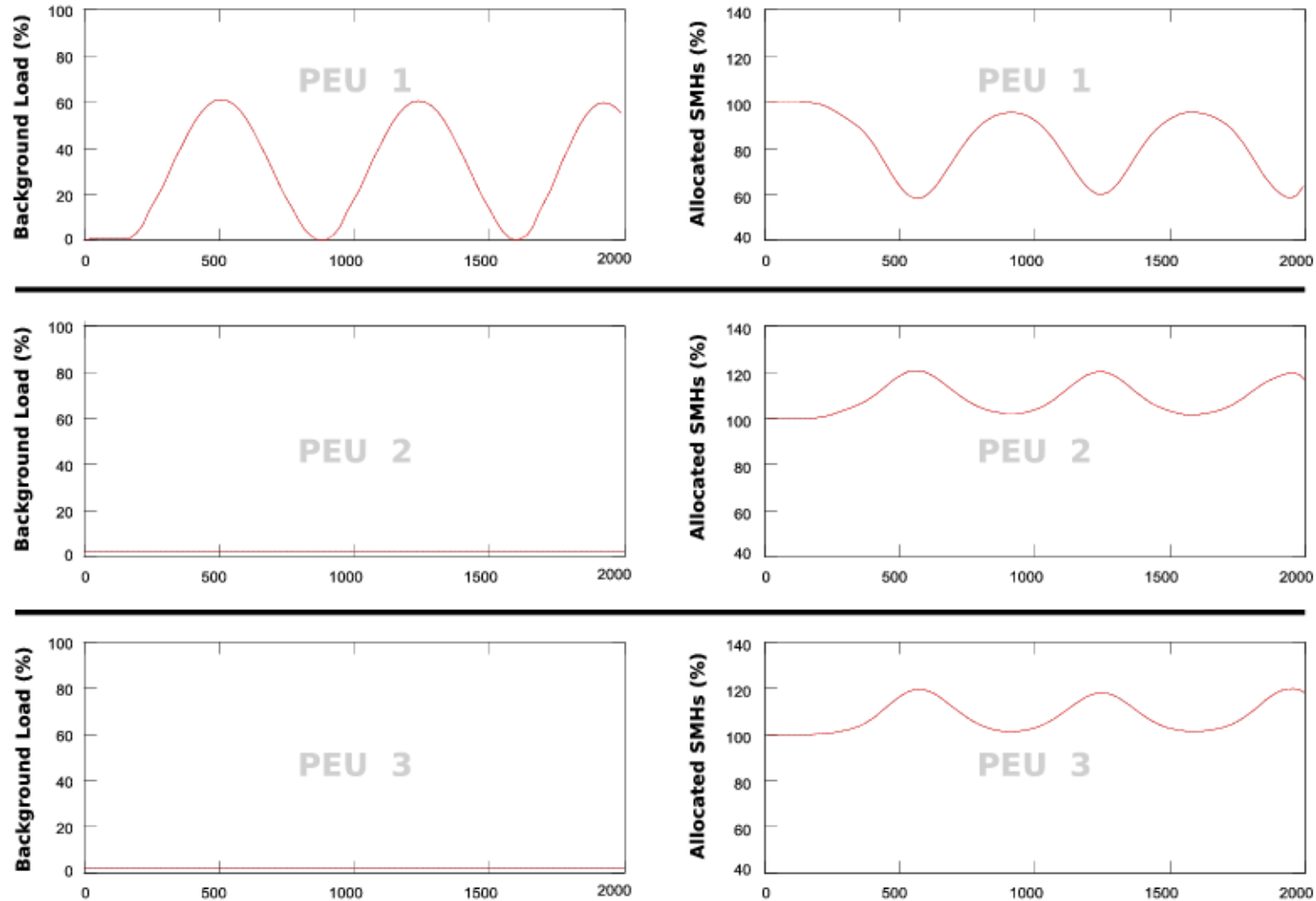


Performance evaluation: MHz-based Distribution (scenario b)



Performance evaluation: Variable Background Load (scenario c)

SMHs Allocation with Variable Background Load



Performance evaluation: Wall-Clock-Time (WCT)

scenario	GAIA	GAIA+	diff (%)
a	3600	3442	-4.38 %
b	3989	3568	-10.41 %
c	5232	4128	-21.10 %

WCT (seconds) to complete a simulation run of 1000 timesteps

Conclusions and future work

- The **communication** and the **load balancing** are two main problems of PADS, they should be **managed together**
- An approach based on the migration of simulated entities can: **reduce the communication overhead** of a distributed simulation, induce a **good level of load balancing** and therefore **increase the simulation speed**
- In many cases the very costly High-Performance-Computing (HPC) clusters can be replaced by **shared clusters of Commercial-Off-the-Shelf (COTS)** computers
- The Computational Grid architecture could benefit of similar approaches, enhancing the use of the Grid for simulation tasks

An Adaptive Load Balancing Middleware for Distributed Simulation

Gabriele D'Angelo

gdangelo@cs.unibo.it

<http://www.cs.unibo.it/~gdangelo>



University of Bologna
Department of Computer Science

WOMP 2006, Sorrento (NA)