Simulation of Scale-Free Networks

Gabriele D'Angelo
<gda@cs.unibo.it>
http://www.cs.unibo.it/gdangelo/

joint work with: **Stefano Ferretti**

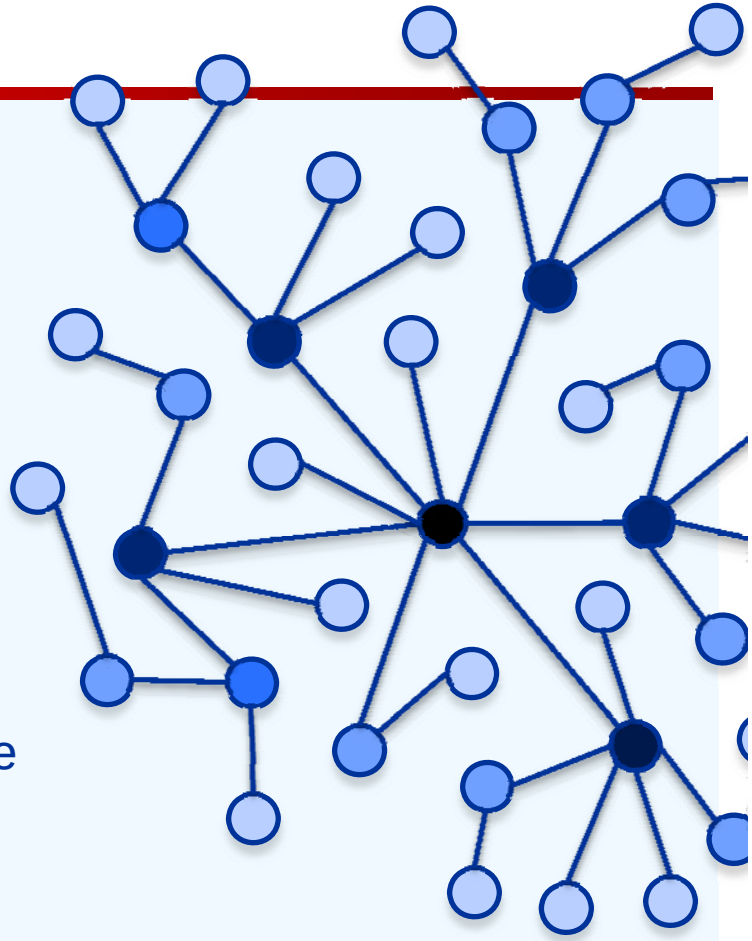
Department of Computer Science
University of Bologna

SIMUTOOLS 2009, Rome (Italy)

Presentation outline

- Scale-Free networks
- Simulation of scale-free networks
- Gossiping protocols
  - fixed probability of dissemination
  - fixed fanout
  - probabilistic broadcast
- Scale-free network simulator: PaScaS / ARTÌS / GAIA
- Experimental evaluation
  - sequential vs. parallel vs. adaptive parallel execution
- Conclusions and future work

Scale-Free networks: definition

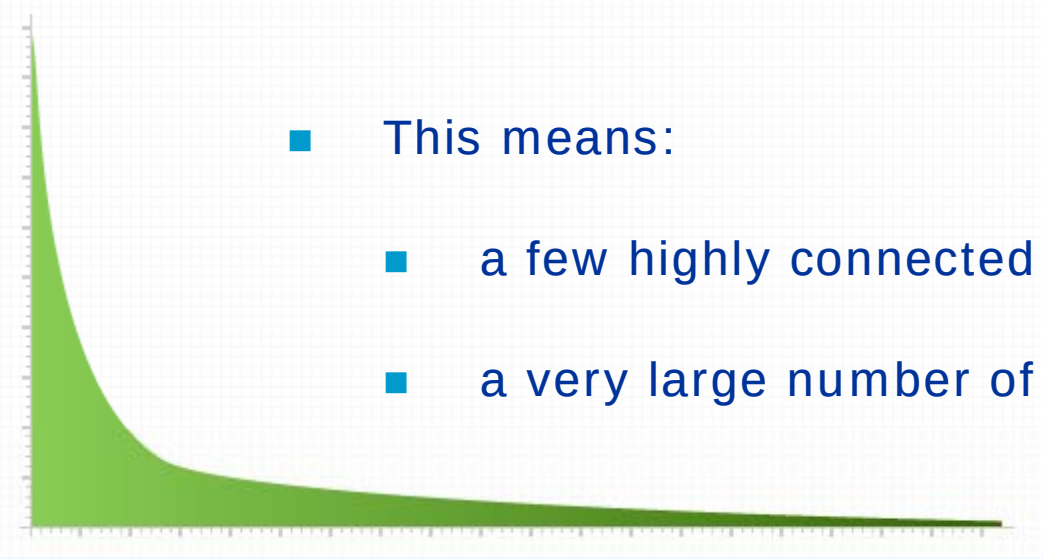


- A **graph** can be used to represent a network and its connectivity
- **Degree of a node** = number of neighbor nodes attached to them
- A **scale-free network** is a network whose degree distribution follows a **power law**
- If $\mathbf{p^k}$ is the probability that a node has a degree equal to **k** then:

  $\mathbf{p^k \sim k^{-\alpha}}$, for some constant value **α**

  (usually: $2 < \alpha < 3$)

Scale-Free networks: meaning and examples



- This means:
  - a few highly connected nodes, called **hubs**
  - a very large number of poorly connected nodes

- Quite good to model several types of real networks:
  - computer networks (e.g. Web, Internet (?))
  - evolving networks in biology
  - transmission of diseases
  - citation graphs
  - social networks, etc.

Simulation of scale-free networks

- Real networks are usually composed of a huge number of nodes
- Under the **simulation point of view**:
  - **large amount of memory** used to represent the massively populated models
  - **huge amount of communication** when simulating real-world protocols on top of such models
  - nodes in real-world networks are very **heterogeneous** (i.e. hubs leaf nodes, different hardware and software characteristics)
- Lack of powerful tools to simulate such complex networks

Gossiping protocols: fixed probability

- Gossiping protocols have been implemented on top of the simulated scale-free networks, to build **realistic case-studies**

- **Gossip #1: Fixed Probability**

  For each message, the node **randomly** selects those edges through which the message must be propagated

  (v = threshold value)

Algorithm

```
function INITIALIZATION()
v ← CHOOSE_PROBABILITY()

function GOSSIP(msg)
for all n_j in Π_j do
    if RANDOM() < v then
        SEND(msg, n_j)
    end if
end for
```

Gossiping protocols: fixed fanout

- **Gossip #2: Fixed Fanout**

Each message is sent to a fixed number of nodes (*fanout*), the receivers are selected at **random** among the neighbors

Algorithm

```
function INITIALIZATION()
fanout ← RETRIEVE_FANOUT()

function GOSSIP(msg)
if fanout ≥ |Π_j| then
    toSend ← Π_j
else
    SELECT_NODES()
end if
for all n_j in toSend do
    SEND(msg, n_j)
end for
```

Gossiping protocols: probabilistic broadcast

- **Gossip #3: Probabilistic Broadcast**

If the message is **locally generated** then it is always **broadcasted** to all neighbors, otherwise it is **randomly** decided if it has to be **broadcasted** or **ignored**

Algorithm

```
function INITIALIZATION()
p_b ← PROBABILITY_BROADCAST()

function GOSSIP(msg)
if (RANDOM() < p_b or
    FIRST_TRANSMISSION()) then
        for all n_j in Π_j do
                SEND(msg, n_j)
        end for
end if
```

PaScaS: the scale-free network simulator

- **Parallel and distributed Scale-free network Simulator (PaScaS)**
  - Implements the building of the scale-free networks (based on the **preferential attachment algorithm**), the **gossiping protocols** and some **data analysis utilities**
  - Based on the **Advanced RTI System** (**ARTÌS**), a middleware used to implement sequential/parallel/distributed simulations that follows an event-based approach
  - It can exploit the adaptive simulation features provided by the **Generic Adaptive Interaction Architecture** (**GAIA**)
  - PaScaS will be freely available as part of **ARTÌS 2.0**, planned for release in the next weeks: http://pads.cs.unibo.it

Experimental evaluation: model parameters

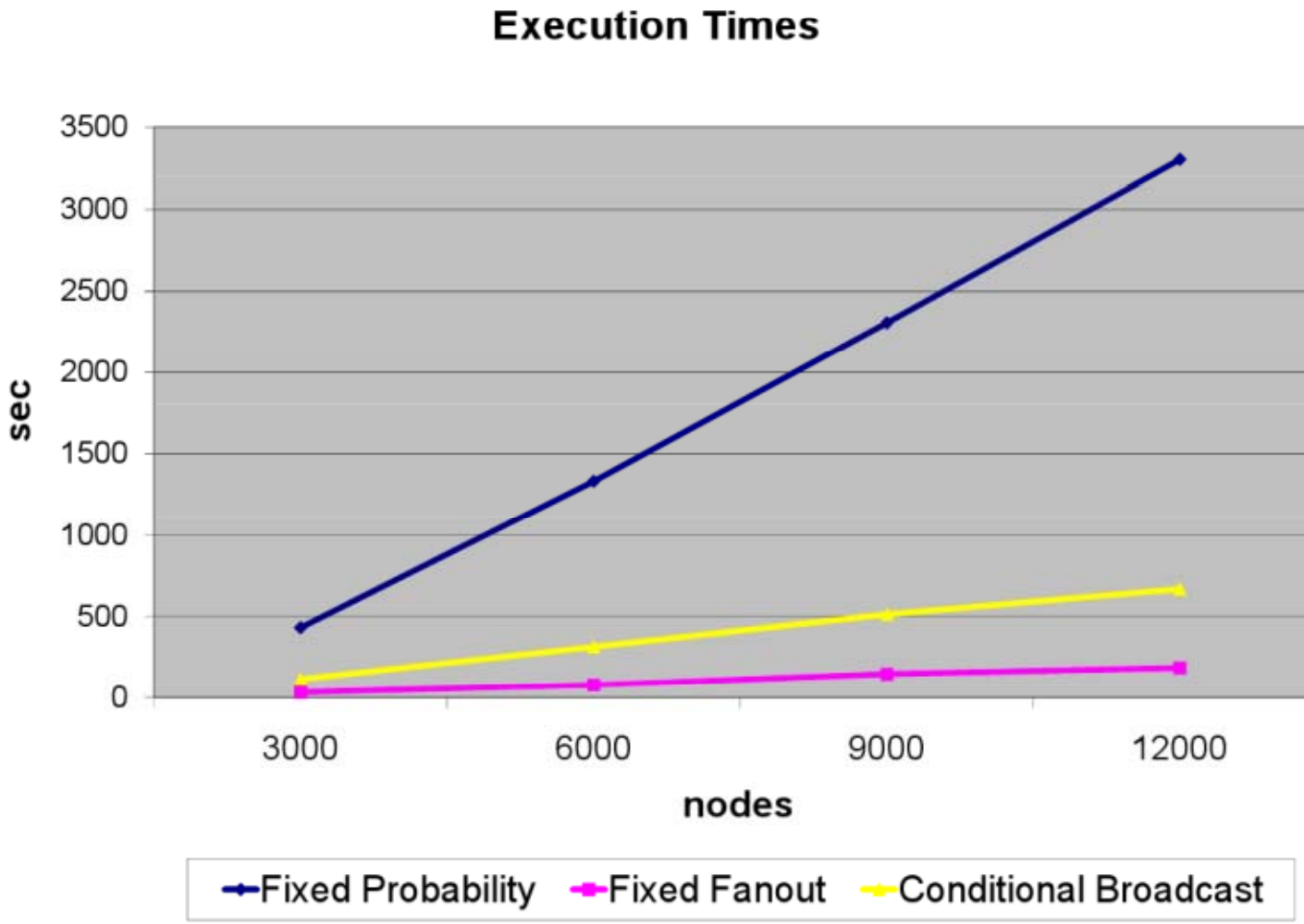
Model parameters and simulation scenario

Parameter	Value
number of nodes	3000, 6000, 9000, 12000
message generation	*exponential distribution* *mean* = 50 *time-steps*
cache size (local to each node)	10 *slots*
message Time To Live (TTL)	6 *(fixed prob. and fanout)* 4 *(conditional broadcast)*
probability of dissemination (v)	0.5 *(i.e. 50%)*
fanout value	5
probability of broadcast (p_b)	0.5 *(i.e. 50%)*
simulated time	1000 *time-steps (after building)*

Experimental evaluation: execution architecture

- **Hardware**:
  - **Dual processor** unit, each processor equipped with a **single-core** Intel Xeon “Gallatin” CPU 2.80 GHz Hyper-Threading, 2 GB RAM
- **Software**:
  - Debian GNU/Linux, Kernel 2.6.16.19
  - PaScaS 0.9, ARTÌS 1.9.3
- **Note**: in case of monolithic (sequential) simulation, PaScaS uses only a single processor (also in presence of multiprocessors or multi-cores)

Experimental evaluation: sequential execution

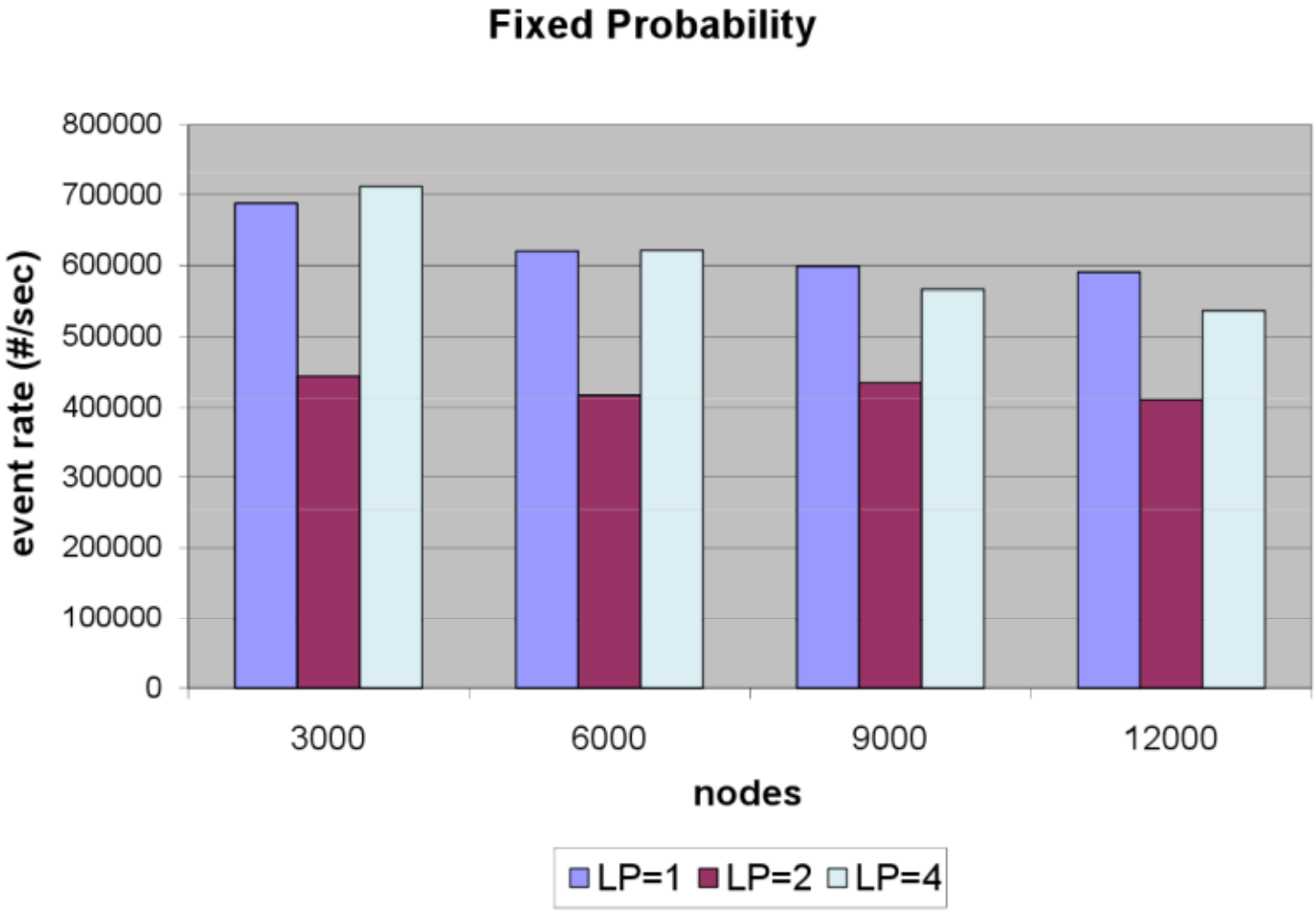


A single process is responsible to manage the whole simulation. The fixed probability has computational requirements higher than other gossip protocols

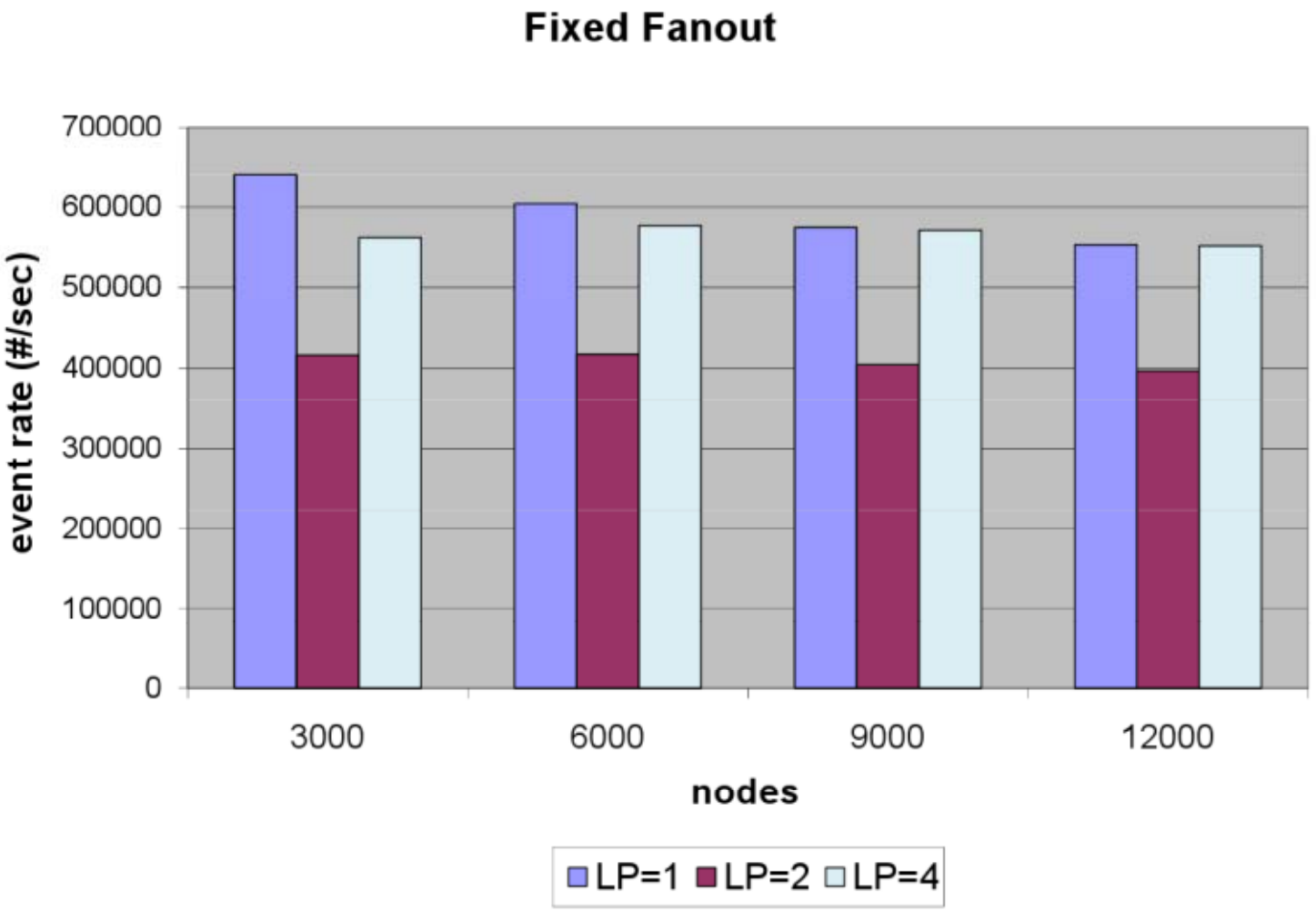
Experimental evaluation: parallel execution

- The simulation is obtained through the coordinated execution of a set of components (**Logical Processes, LPs**)
- Each LP manages the evolution of a **part of the model** and it is usually run by a different CPU
- Each node in the scale-free net is modeled as a **Simulated Entity (SE)**, therefore each LP manages a set of SEs
- SEs are **randomly allocated** in the LPs
- It is worth noting that the **performance** of a Parallel And Distributed Simulation (PADS) is a **trade-off** between: **a)** load parallelization; **b)** communication overhead in the parallel/distributed architecture (also including synchronization)

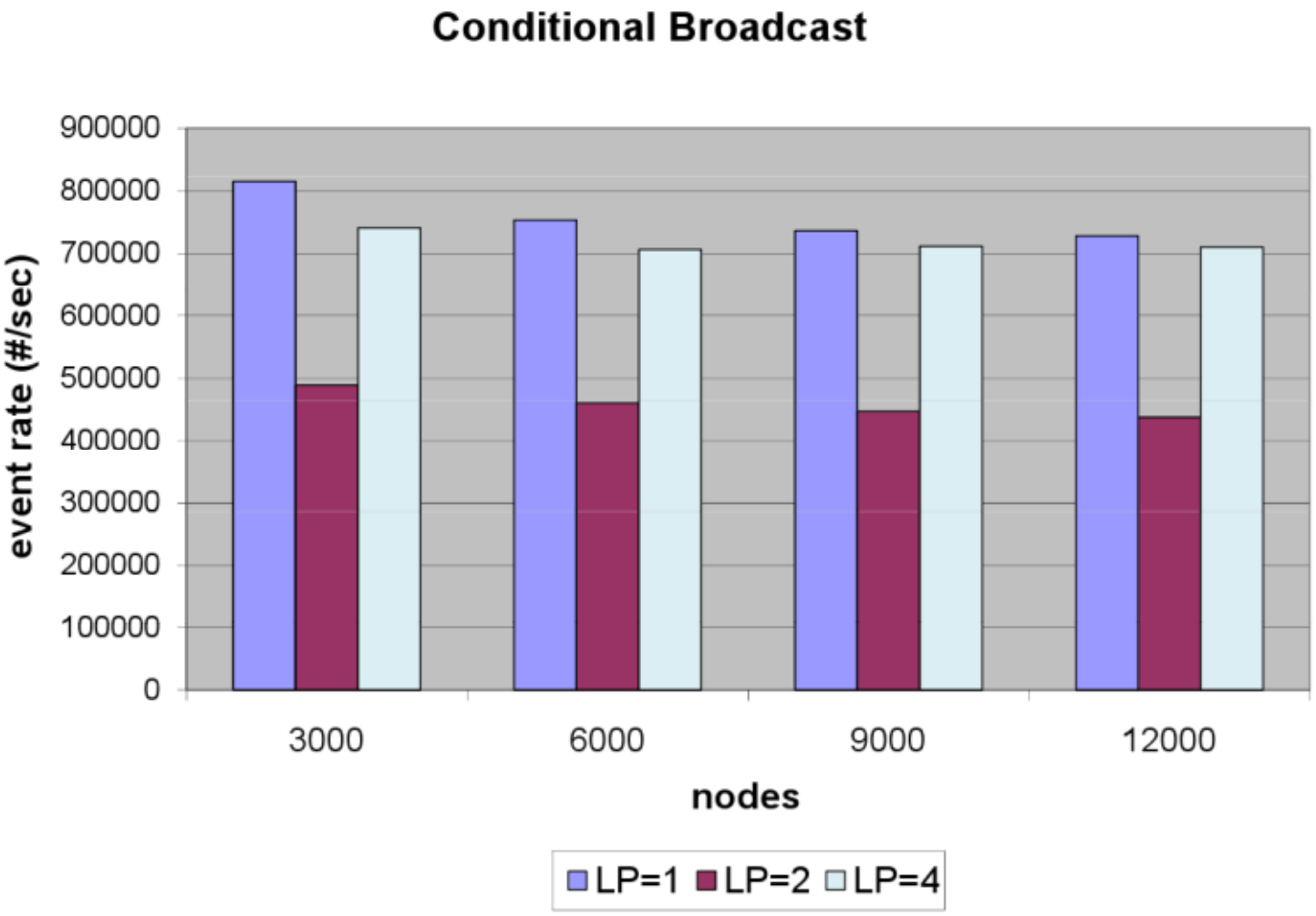
Experimental evaluation: parallel execution (fixed probability)



Experimental evaluation: parallel execution (fixed fanout)



Experimental evaluation: parallel execution (cond. broadcast)



Experimental evaluation: monolithic vs. parallel execution

- **Performance gap** (%) between **LP=1** (monolithic) and **LP=4**

Nodes	Gossip #1	Gossip #2	Gossip #3
3000	**3.46**	-12.22	-9.1
6000	**0.19**	-4.49	-6.23
9000	-5.35	-0.63	-3.36
12000	-9.07	-0.25	-2.42

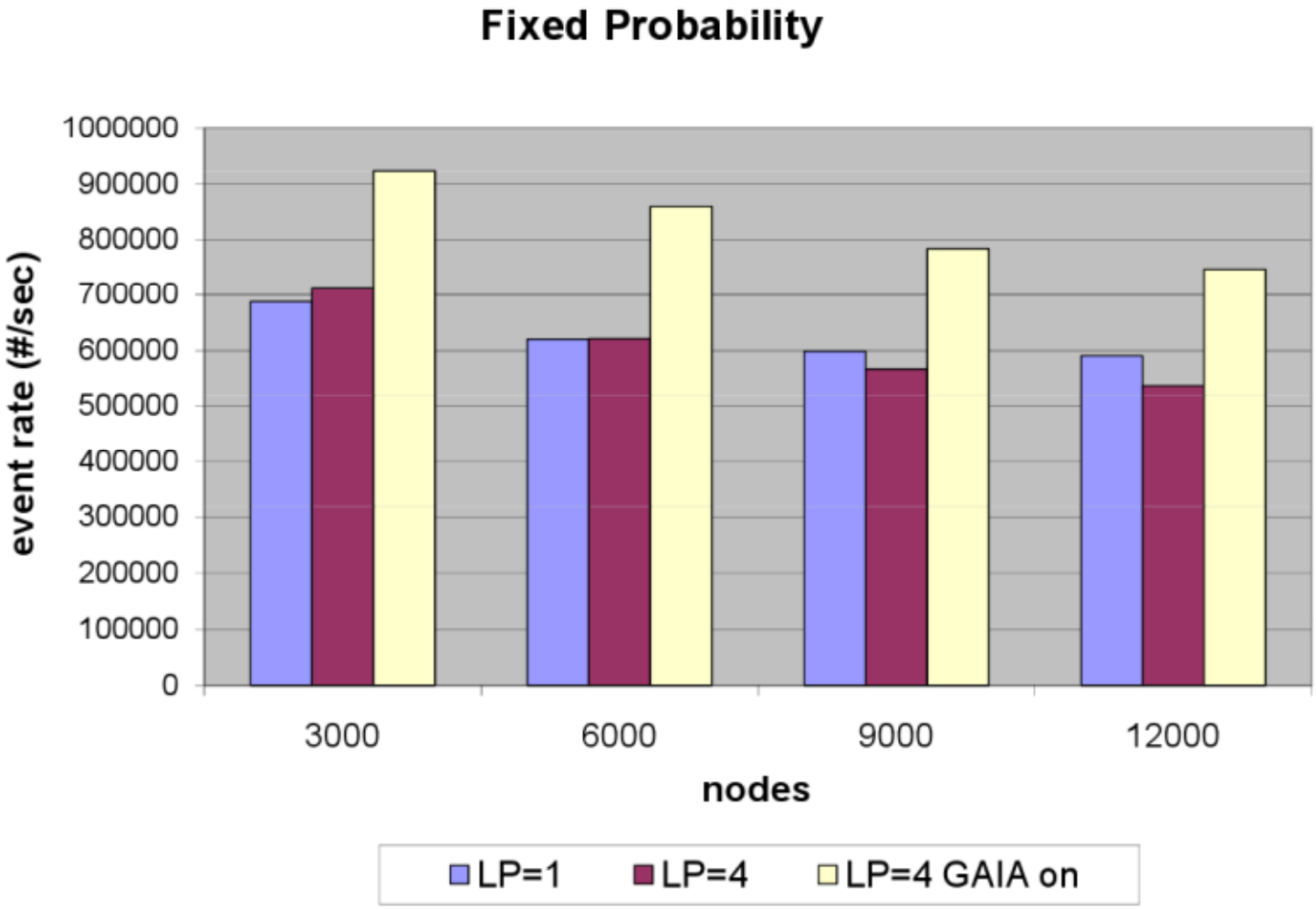
green *= parallel faster than monolithic*

- The parallel execution gives very **unsatisfactory results**
- The communication overhead introduced by the parallel execution does not balance the gain given by the load parallelization

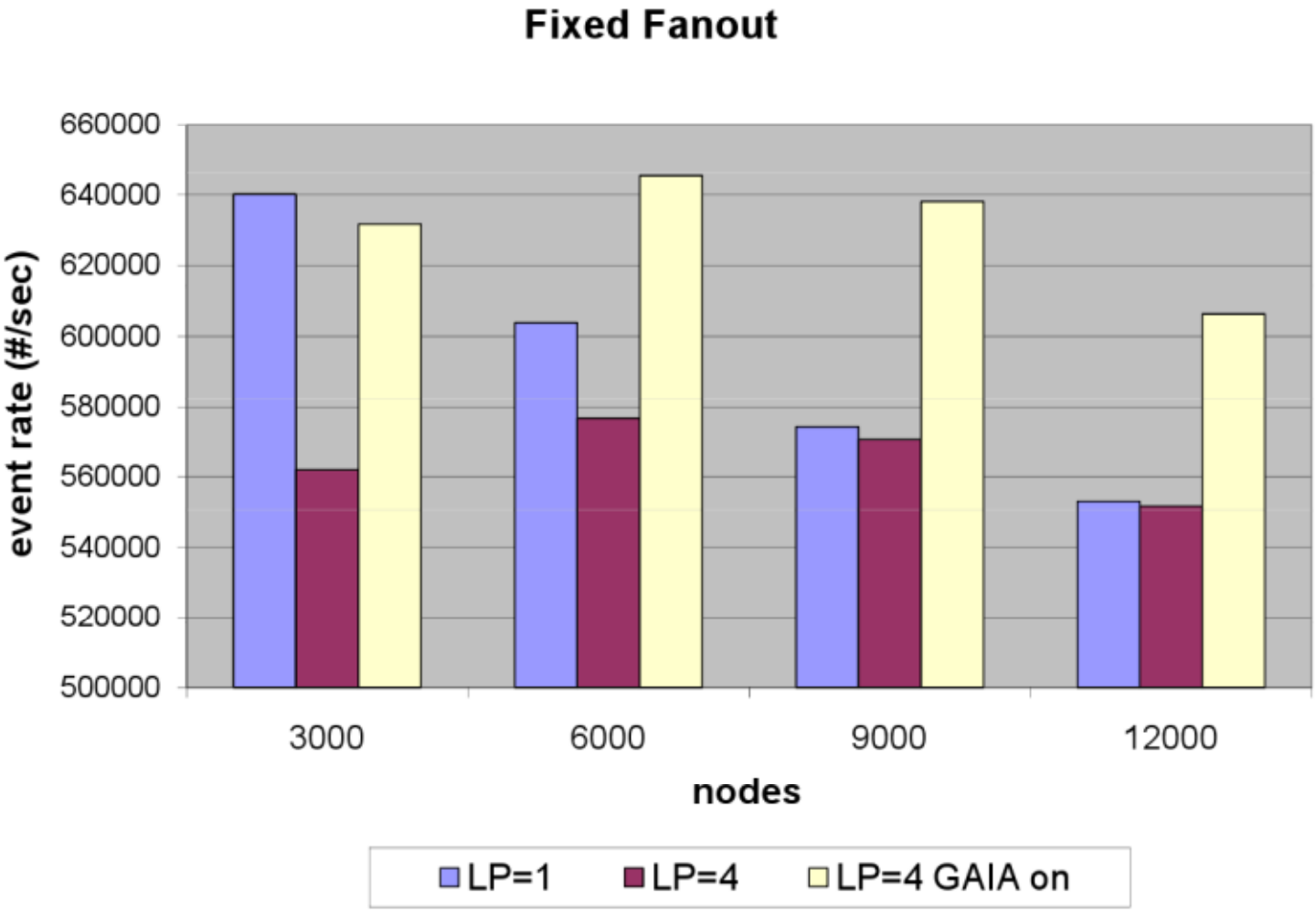
Experimental evaluation: adaptive parallel execution

- The goal of this approach is to reduce the communication overhead
- **Observation**: in PADS the LP-to-LP (that is CPU-to-CPU) communication represents and overhead
- **GAIA** framework introduces an **adaptive mechanism** that:
  - step by step analyzes the **communication pattern** of each Simulated Entity (SE)
  - using a migration-based approach, **clusters** the highly interacting SEs in the same LP
  - introduces a cost (i.e. migration) but in many cases reduces the communication overhead
- The mechanism dynamically reacts to the model behavior
- It is based on **heuristics** that controls the migrations and aims to good **load balancing** in the execution architecture

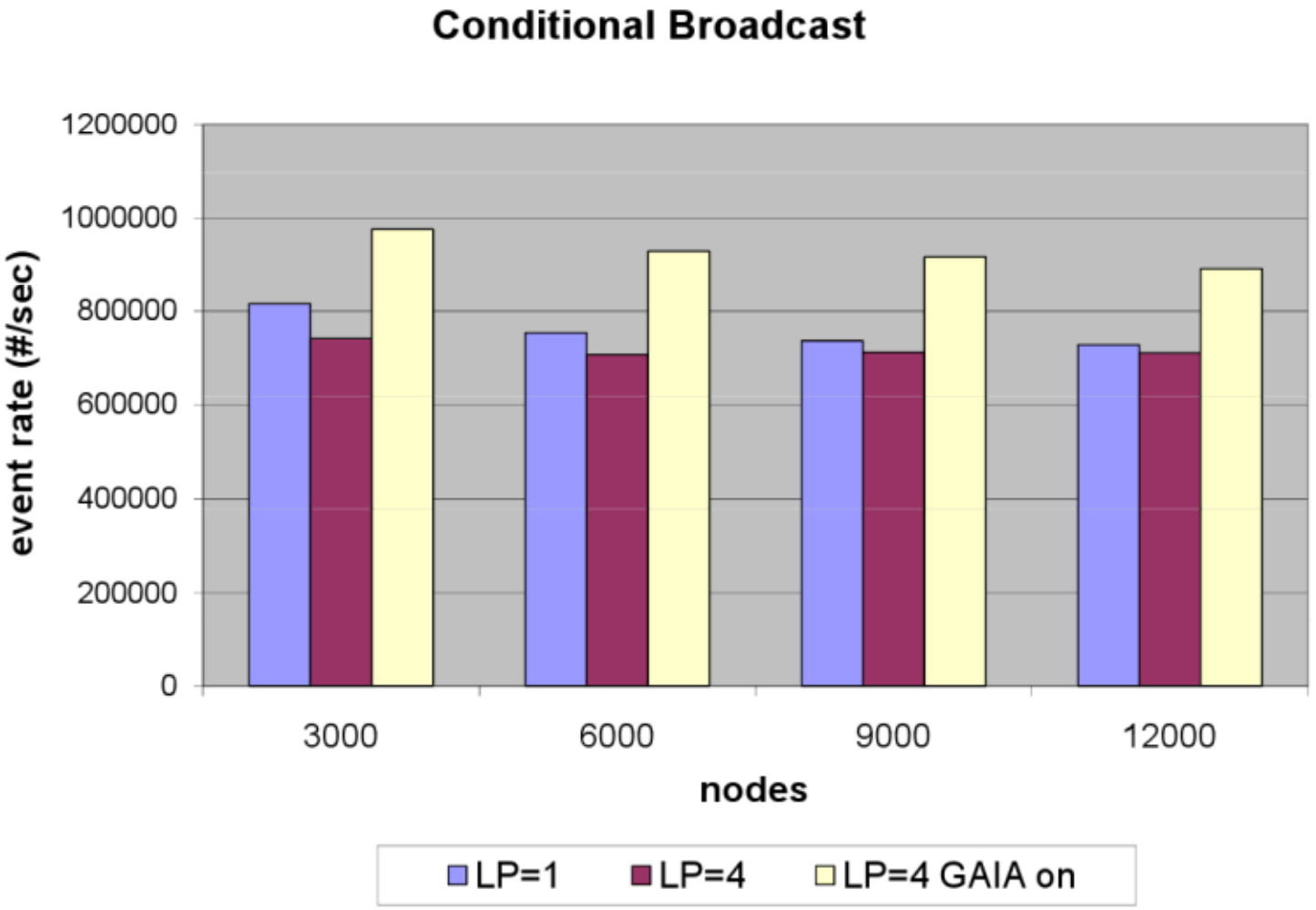
Experimental evaluation: adaptive parallel execution



Experimental evaluation: adaptive parallel execution



Experimental evaluation: adaptive parallel execution



Parallel execution vs. adaptive parallel execution

- **Performance gap** (%) between **LP=1** and **LP=4 GAIA ON**

Nodes	Gossip #1	Gossip #2	Gossip #3
3000	**34.18**	-1.33	**19.75**
6000	**38.63**	**6.87**	**23.37**
9000	**30.97**	**11.07**	**24.59**
12000	**26.47**	**9.65**	**22.56**

green *= parallel faster than monolithic*

- The results obtained by **adaptive parallel execution** are quite good
- The best performances are obtained for gossip #1 and gossip #3, because both protocols are communication-bounded

Conclusions and future work

- **PaScaS** is a novel freely available scale-free network simulator
- **Parallel approach** gives **unsatisfactory results** in the simulation of such complex networks
- An approach based on **dynamic** and **adaptive clustering** of the simulated entities can give valuable results

- **Future work**: more **building algorithms** for scale-free nets, more **detailed heuristics** for the clustering, support for **multi-core processors**, detailed evaluation of the performances of the gossip protocols in very **heterogeneous networks**

For more information

- **PaScaS / ARTÌS software:** **http://pads.cs.unibo.it**

- **Adaptive parallel/distributed simulation**:

  G. D'Angelo, M. Bracuto. ***Distributed Simulation of Large Scale and Detailed Models***. To appear, International Journal of Simulation and Process Modelling (IJSPM), Special issue on "Parallel and Distributed Simulation“, InderScience, 2009.

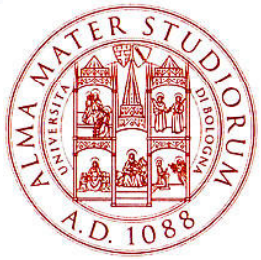
  (email to: gda@cs.unibo.it for the draft)

Simulation of Scale-Free Networks

Gabriele D'Angelo
<gda@cs.unibo.it>
http://www.cs.unibo.it/gdangelo/

joint work with: **Stefano Ferretti**

Department of Computer Science
University of Bologna



SIMUTOOLS 2009, Rome (Italy)