LUNES: Agent-based Simulation of P2P Systems

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

joint work with:
Stefano Ferretti

Istanbul, Turkey

International Workshop on Modeling and Simulation of Peer-to-Peer Architectures and Systems (MOSPAS), 2011
Presentation outline

- Yet another **simulator** of P2P systems?
- It is all about **scalability**
- **Discrete Event Simulation (DES)** paradigm
- **Large Unstructured Network Simulator (LUNES)**
- **Parallel And Distributed Simulation (PADS)**
- **Adaptive PADS**: ARTÌS/GAIA/LUNES
- **Performance** evaluation
- **Conclusions** and **future work**
Motivations

- Do we really need yet another simulator of P2P systems?
- The main question is: are you satisfied with current tools?
What's the problem? **Scalability** issues

- **Traditional simulation tools** are **unable to cope** with very large, dynamic, complex and detailed models.

- Many systems are made by a **very large number** of nodes.

- Such nodes can be **heterogeneous** (*with different characteristics*) and very **dynamic** (*in and out of the network*)

- The network topology can be **complex** (*random, scale-free, small-world*)

- The performance evaluation (of such systems) often requires **fine-grained** and **detailed** models of communication protocols.
Simulation paradigm: discrete event simulation

- In the years many simulation paradigms have been proposed
- **Discrete Event Simulation** (DES): very **good expressiveness** and **quite easy** for model developers
- The model evolution is obtained through a chronological sequence of **events**
- Each event is a **change in the system state** and occurs at an instant in time
- The **creation, delivery** and **computation** of events is the main task to be done by the simulator
- In many tools (**PeerSim**, **ns-2**, **OMNeT++**, ...) all is managed by a single CPU (**sequential simulation**)
LUNES: Large Unstructured NEtwork Simulator

- Overall design of LUNES:
  - network topology creation
  - protocol simulation
  - trace analysis
- Different tools for different tasks: all phases are quite complex
LUNES: Large Unstructured Network Simulator

- Overall design of **LUNES**:
  - network topology creation
  - protocol simulation
  - trace analysis
- Different tools for different tasks: all phases are quite complex

The **initial network topology** can be generated using the more appropriate tool (e.g. igraph, custom generators ecc.) and it is exported to the “protocol simulation” module using the graphviz dot language.
LUNES: Large Unstructured Network Simulator

- Overall design of LUNES:
  - network topology creation
  - protocol simulation
  - trace analysis

- Different tools for different tasks: all phases are quite complex

This does not mean that the network topology is static!

The protocol simulation can easily modify the topology at runtime.

The initial network topology can be generated using the more appropriate tool (e.g. igraph, custom generators ecc.) and it is exported to the “protocol simulation” module using the graphviz dot language.
LUNES: Large Unstructured NEtwork Simulator

- Overall design of LUNES:
  - network topology creation
  - protocol simulation
  - trace analysis

- Different tools for different tasks: all phases are quite complex

The core of the simulator: it implements the specific P2P protocols and manage the network topology. It uses the services provided by the simulation middleware.
LUNES: Large Unstructured NEtwork Simulator

- Overall design of LUNES:
  - network topology creation
  - protocol simulation
  - trace analysis

- Different tools for different tasks: all phases are quite complex.

Fine-grained and detailed protocol generate very **verbose trace files**. For statistical correctness many runs have to be completed.

The output generated by medium complexity models is in the order of **gigabytes** (per run).
Parallel and Distributed Simulation (PADS)

- The P2P system is implemented as a **Multi Agent System**: a set of interacting **Simulated Model Entities (SMEs)**
- Each **SME** implements the behavior of an **agent** (e.g. a **peer**)
- The model is partitioned in a set of **Logical Processes (LPs)**
- Each **LP** allocates a set of **SMEs** and is executed on a different **CPU**

- **Parallel simulation**: *the CPU are interconnected by a low latency network (e.g. a bus or shared memory)*

- **Distributed simulation**: *there is a higher latency network that interconnects the different parts of the distributed simulator (e.g. LAN, WAN, Internet)*
**PADS: PROS and CONS**

- **PROS:**
  - The main advantage is that using many CPUs is possible to parallelize computation
  - With the aggregation of memory resources, larger models can be represented

- **CONS:**
  - Synchronization of the distributed execution architecture
  - Data distribution management (delivery of updates)
  - Communication is much more costly than in a single CPU

**PADS can be slower than sequential!**
PADS of P2P systems

- Why is so hard using PADS for P2P systems?

- Because such models are communication bounded (much more than computation)

- ... and in PADS the communication is very costly!

- Execution time saved by parallel computation is often lost in communications (e.g. synchronization, state updates)

- Such applications are not embarrassingly parallel

- In many models, increasing the number of nodes has a linear cost in terms of computation and a super linear increase of communication
Adaptive PADS

- A “suitable” allocation of Simulated Model Entities (SMEs) can greatly reduce the communication cost.

- This is the PADS partitioning problem: with dynamic and heterogeneous systems the static solution does not work!

- Adaptive partitioning: based on the simulation execution.

- The idea is to observe the communication pattern of each SME and to cluster adaptively the highly interacting SME in the same LP (that is on the same CPU).

- This can reduce the costly inter-LP communication.

- Some subtle details are missing from this high level description (e.g. migration of SMEs, load balancing and synchronization).
**ARTÌS/GAIA and LUNES**

- **ARTÌS**: simulation middleware, provides the **basic functionalities** (synchronization, communication, coordination etc.)

- **GAIA**: implementation of **adaptive PADS**. Insulates the middleware from the model. Provides a **Multi Agent System** (MAS) abstraction

- **LUNES**: **model skeleton** with the basic functionalities of P2P systems

---

For details and software download: [http://pads.cs.unibo.it](http://pads.cs.unibo.it)
Performance evaluation of LUNES

- Simulated P2P protocol: **data dissemination (gossip based)**
- **Fixed-probability** and **adaptive gossip**
- Both are very **communication intensive** but the adaptive gossip is slightly more **computation demanding**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>number of <strong>nodes</strong></td>
<td>200-500</td>
</tr>
<tr>
<td>number of <strong>edges</strong></td>
<td>400-1000</td>
</tr>
<tr>
<td>number of <strong>graphs</strong> per evaluation</td>
<td>10</td>
</tr>
<tr>
<td>construction method</td>
<td><strong>Erdos-Renyi generator</strong></td>
</tr>
<tr>
<td>cache size (local to each node)</td>
<td>256 slots</td>
</tr>
<tr>
<td>message Time To Live (<strong>ttl</strong>)</td>
<td>8, 9</td>
</tr>
<tr>
<td>simulated time</td>
<td>1000 time-steps (after building)</td>
</tr>
</tbody>
</table>
Evaluation: **delivered messages**

LUNES, fixed-probability dissemination, number of delivered messages

- $n=200, e=400, \text{ ttl}=8$
- $n=300, e=600, \text{ ttl}=9$
- $n=400, e=800, \text{ ttl}=9$
- $n=500, e=1000, \text{ ttl}=9$
Evaluation: **delivered messages**

This dissemination protocol run (1000 timesteps) on top of a network of 500 nodes (1000 links) generates more than **60.000.000** messages.
Evaluation: **speed-up**, fixed probability dissemination

LUNES scalability, fixed-probability dissemination

- 500 nodes, 1000 edges
- 400 nodes, 800 edges
- 300 nodes, 600 edges
- 200 nodes, 400 edges

Graph showing speedup with varying numbers of LPs and different node and edge counts.
Evaluation: **speed-up**, fixed probability dissemination

LUNES scalability, fixed-probability dissemination

\[ \text{Speedup} = \frac{T_l}{T_p} \]

- \( T_l \) = execution time for **sequential**
- \( T_p \) = execution time for **parallel**

Higher is better!
Evaluation: speed-up, fixed probability dissemination

LUNES scalability, fixed-probability dissemination

Speedup < 1
parallel is slower than sequential
Evaluation: **speed-up**, fixed probability dissemination

LUNES scalability, fixed-probability dissemination

- 500 nodes, 1000 edges
- 400 nodes, 800 edges
- 300 nodes, 600 edges
- 200 nodes, 400 edges

**Graph:**
- **Y-axis:** Speedup
- **X-axis:** Number of LPs
Evaluation: **speed-up**, adaptive dissemination

![Graph showing LUNES scalability, adaptive dissemination](image)

- 500 nodes, 1000 edges
- 400 nodes, 800 edges
- 300 nodes, 600 edges
- 200 nodes, 400 edges
Evaluation: **speed-up**, adaptive dissemination

![LUNES scalability, adaptive dissemination graph](image)

- **GAIA gain**
  - 2 LPs
  - 500 nodes
  - 1000 edges

Number of LPs: 1, 2, 4

- 500 nodes, 1000 edges
- 400 nodes, 800 edges
- 300 nodes, 600 edges
- 200 nodes, 400 edges
Conclusions and future work

- The simulation of **P2P protocols** on top of **large scale** networks is still a hard problem with many **scalability issues**.

- **Parallel And Distributed Simulation (PADS)** is a promising but appropriate techniques for the **reduction of the communication overhead** are necessary.

- **LUNES**: a new simulator for the performance evaluation of protocols on large scale networks.

- **PADS techniques** are **too complex** for many simulators users. The next effort has to be: “**easy to use PADS**”

- More complex (and efficient) forms of **PADS adaptivity**
Further information

Gabriele D'Angelo, Stefano Ferretti

LUNES: Agent-based Simulation of P2P Systems


An extended version of this paper is freely available at the following link:


The ARTÌS middleware, GAIA framework and LUNES can be downloaded from:

- http://pads.cs.unibo.it

Gabriele D'Angelo

- E-mail: <g.dangelo@unibo.it>
- http://www.cs.unibo.it/gdangelo/
LUNES: Agent-based Simulation of P2P Systems

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

joint work with:
Stefano Ferretti

Istanbul, Turkey

International Workshop on Modeling and Simulation of Peer-to-Peer Architectures and Systems (MOSPAS), 2011