COGITARE: A Cloud Infrastructure for Grid and Overlay Network Simulation Research

Luís Veiga, João Nuno Silva, João Garcia
INESC-ID / Technical University of Lisbon, Portugal
{luis.veiga, joao.n.silva, joao.c.garcia}@inesc-id.pt

ABSTRACT

We describe COGITARE, a novel approach to grid and overlay network research leveraging distributed infrastructures and multi-core machines for increased simulation complexity and speed. We present its motivation, background, current shortcomings and the core architectural concepts of the novel research proposed. This is an on-going effort to further research on topologies for peer-to-peer overlays, cycle-sharing and the Grid by providing a scalable, efficient and reliable simulation substrate for the very grid and overlay topologies developed by the research community.

Thus, grid and overlay simulation is improved due to: 1) increasing scalability of simulation tools with a novel parallel, distributed and decentralized architecture; 2) unleashing the power of idle CPU cycles over the Internet harnessing them as a desktop grid (on a peer-to-peer overlay); and 3) ease grid and overlay research with a framework for topology definition, dissemination, evaluation and reuse.

The infrastructure, simulation engine, topology modeling language, management services comprise a Grid4GridResearch platform that can be accessed via a portal.

1. INTRODUCTION

Grid infrastructures and peer-to-peer overlay networks are areas of very active research within the distributed systems, middleware and network communities. Given the large number of elements in such topologies (grids and overlays), an important fraction of the research in this field is performed not on real systems but instead resorting to simulation tools (SimGrid [2], PeerSim, OverSim [1]), hence avoiding machine cost and administration issues.

However, current grid and overlay simulation is encumbered by architecture, scale and performance limitations that cannot be solved simply by stacking more powerful computers. The performance of simulation tools is hindered because network and topology simulation code is mostly serial and manipulates a large global state assumed to be consistent. Simulations are run centrally, serially not drawing many of the advantages of increasingly prevalent multi-core machines or computing clusters. Thus, although the increased capability may be used to execute more grid and overlay simulations simultaneously, it is not possible to run each individual simulation faster. Moreover, one cannot leverage more CPUs to run more complex simulations (larger number of elements) within a given time frame.

There are three important challenges that correspond to present shortcomings causing drag in these areas, regarding not only research but also even teaching: i) serial nature of simulation tools, commonly involving global state and event queue manipulations, prevent the scale-up that could be provided by the increasing democratization of access to multi-core machines (e.g., dual-core laptops and quad-core desktops); ii) impossibility to execute simulations in distributed manner thereby preventing the scale-out that could be reached by engaging platforms for voluntary cycle-sharing, utility-computing and grid infrastructures (in testbeds as PlanetLab full execution imposes an overhead that greatly limits simulation size), and iii) dominance of programmatic descriptions of researched and tested topologies, locked in specific programming languages and simulator API idioms.

The aforementioned shortcomings cause a number of problems that hinder the expansion of current overlay and grid research, both in depth, breadth and community impact. Examples of such include: limitations to size and complexity of the simulations; limited scope of results; inefficient employment of resources involved in experiments; lack of architecture and simulator-agnostic descriptions of protocols, middleware, and schedulers; very restricted ability to perform independent repeated research [3]; no uniform repository for reusability of such research; global absence of accessible teaching platforms.

We propose to further grid and overlay simulation by: 1) increasing scalability of simulation tools with a novel parallel, distributed and decentralized architecture; 2) unleashing the power of idle CPU cycles over the Internet harnessing them as a desktop grid (on a peer-to-peer overlay); and 3) ease grid and overlay research with a framework for topology definition, dissemination, evaluation and reuse.

In essence, this provides a Grid4GridResearch.

2. ARCHITECTURE

The primary substrate for COGITARE is an extendable peer-to-peer overlay platform (the *mesh*) comprising possibly asymmetrical participant nodes, aggregating individual desktop peers as well as efficiently leveraging available time on server clusters, grid sites and EC2 utility-computing time.

Simulations of peer-to-peer overlay and grid topologies are subject to parallel and distributed execution. COGITARE deploys *partial simulations* of topology regions over the mesh nodes. Thus, what is now mostly a single main task in current simulation tools can be decoupled in order to allow concurrent progress of (partial) simulation of different regions of the simulated topologies.

To overcome the lack of expressiveness in current simulation tools, regarding the definition of simulations, we propose a domain-specific language that allies greater expressiveness with the ability of reuse for teaching, study, and repeated research. Such a topology modeling language (TML) allows the specification of simulation requirements, as well as flat, layered, multidimensional, hierarchical and novel recursive topologies to simulate arbitrarily large systems.

Over time, the execution of simulations on the mesh, together with the expectable fluctuations in the mesh membership, trigger higher-level services such as migration in order to restore load-balance. Moreover, performance is increased by aggregating simulated regions with more intensive intercommunication within neighboring mesh nodes, even within the same node if sufficient capability is available there.

Finally, we are extending available open-source groupware and content management systems to create a portal, for groups and communities to interface with COGITARE and interactively deploy, simulate and share topologies, simulations, results.

3. OVERLAY INFRASTRUCTURE

The COGITARE mesh is a peer-to-peer overlay able to

harness idle computing cycles from asymmetrical participant nodes. Thus, it incorporates specific mechanisms and interfaces to engage peers, being them individual desktop, server clusters, grid sites and utility-computing time slices.

Each type of peer has different capabilities and a dedicated module is able to access them (invoking corresponding middleware) and exploit them (representing them as higher capacity nodes) by feeding higher simulation loads and additional simulation/result data to store. The mesh is extendable. It serves only to provide fundamental support for: simulation deployment, result caching, data storage, and resource discovery and management for cycle-sharing and distributed scheduling. The mesh obeys to a hybrid and hierarchical structure. In this sense, the mesh behaves as recursive overlay topology.

Concerning content and routing, the mesh is a structured overlay. Each top-level data item stored in the mesh is in itself either: topology description, description of resource discovery and management policies, simulation setting, or results of simulation execution.

Regarding base-level services such as resource discovery (mainly w.r.t, CPU, memory, bandwidth) and scheduling of simulations, the mesh employs automatically designated superpeers to act as hubs of information about resources of groups comprising their neighboring nodes. Superpeers aggregate into a structured overlay only to exchange resources when there are not enough within each group.

4. PARALLEL/DISTRIBUTED SIMULATION

Parallelized and distributed simulation of topologies allows speedups and increases scalability by enabling the mesh to host simulations of possibly unbounded size and interaction complexity. In fact, most events localized in the vicinity of a simulated node cannot interfere with other simulated nodes many hops afar. Therefore, consistency can be enforced by employing more flexible approaches.

Each node of the mesh executes a component of the distributed simulation engine. Communication among nodes running partial simulations of one topology simulation resorts to a standard API that either manages shared memory or exchanges messages via the mesh. Partial simulations of a given simulation may be running in the same or different nodes.

Each concurrent simulator thread is in charge of executing a number of simulation steps to simulate a specific region of the topology, i.e., it performs a partial simulation. Events localized in the vicinity of a simulated node cannot interfere with other simulated nodes at a distance, preventing them from being influenced by it (i.e., out of the event/message horizon for a given number of hops) until some time has passed. This leverages notions of locality-awareness employed in massive multiplayer games: while guaranteeing global consistency, a region-based approach speeds up most of game interaction. The amount of time, rounds and divergence forcing synchronization among regions to ensure safety when interactions cross region boundaries, is enforced resorting to novel optimistic consistency algorithms, adapted from existing divergence-bounding and vector-field consistency[4].

5. TOPOLOGY MODELING LANGUAGE

TML is designed to overcome the lack of expressiveness, w.r.t. definition of toplogies, in current simulation tools. TML syntax includes overall definition of simulations and topology structure (e.g., flat, layered, multidimensional, hierarchical and possible novel recursive topologies). This allows easy definition of arbitrarily large systems in comparison with today's cumbersome programmatic approach. Regarding data structures inside simulated nodes, TML syntax encompasses rules to specify neighbor and routing tables. Concerning behavior, TML allows expressing protocols (e.g., entry, exit, routing, and recovery). TML also allows the definition of pre-existing data content placed at simulated nodes prior to simulation start.

At a higher level, TML improves simulations by separating topology structure and routing from higher-level policies such as resource discovery, reservation, scheduling, accounting, recycling, redundancy, versioning. Once defined, the description of a policy should be reusable in other simulations. This way, definitions of topology-related structure, data, behavior and policy can be mixed-and-matched to increasing reusability, productivity and observability of studied properties.

6. HIGH-LEVEL SERVICES

Being the mesh mostly comprised of voluntary desktop machines, node exit and failure will be frequent, not an exceptional event. This may affect fairness, balancing, resource efficiency and reliability. To address this, a number of higher-level service modules are deployed on top of the basic mesh, embedded in clients bearing mesh protocol and simulator engine.

The base mechanism to perform fair load-balancing is migration of partial simulations. This is also useful to attempt at co-locating sibling partial simulations in the same mesh node or neighboring nodes. This reduces network communication among simulation engines and increases speed. Overall balance and fairness are driven by global resource recycling and accounting to enforce similar quality-of-service across users.

For reliability and preventing wasted work, fundamental support for fault-tolerance is provided. Each partial simulation is scheduled several times to different nodes in the mesh, its results gathered and compared in order to confirm their correctness. To ensure that very long running simulations make any progress, partial simulations are checkpointed periodically (possibly migrated) and their state stored in the mesh as regular simulation result data.

A user interface to access COGITARE services is provided by the GRid4GridResearch portal. It supports content sharing (topologies, simulations, policies, results) among users, groups, and simulation-related communities. Web interactive and automatic (scripted) operation are supported by XML-RPC and REST-based interfaces. This produces the interesting outcome of allowing embedding of COGITARE in web pages and mashups, and automatic deployment and retrieval of results.

7. CONCLUSION

This document proposes a novel approach to grid and peer-to-peer overlay simulation, by performing it in parallel and distributed fashion on top of a mesh, that itself, manages a voluntary desktop grid on a peer-to-peer overlay. Topologies and simulations are defined with higher expressiveness using a domain-specific topology modeling language. Higher-level services deployed over the mesh include base support for fault-tolerance, load-balancing and resource management integrated with a portal for community access.

8. REFERENCES

[1] I. Baumgart, B. Heep, and S. Krause. Oversim: A flexible overlay network simulation framework. In *IEEE Global Internet Symposium*, 2007.

[2] R. Buyya and M. M. Murshed. Gridsim: a toolkit for the modeling and simulation of distributed resource management and scheduling for grid computing. *Concurrency and Computation: Practice and Experience*, 14(13-15), 2002.

[3] S. Naicken, B. Livingston, A. Basu, S. Rodhetbhai, I. Wakeman, and D. Chalmers. The state of peer-to-peer simulators and simulations. *Computer Communication Review*, 37(2), 2007.

[4] N. Santos, L. Veiga, and P. Ferreira. Vector-field consistency for ad-hoc gaming. In *ACM Middleware*, 2007.

COGITARE: A Cloud Infrastructure for Grid and Overlay Network Simulation Research

Luís Veiga
luis.veiga@inesc-id.pt

João Nuno Silva
joao.n.silva@inesc-id.pt

João Garcia
joao.c.garcia@inesc-id.pt

INESC-ID / Technical University of Lisbon, Portugal

Distributed Systems Group

Grid and P2P simulations

Challenges
- serial nature of simulation code
- impossibility to executing simulations in distributed manner
- dominance of programmatic approaches to topology description

Current problems
- limitations on size and complexity of simulations
- inefficient resource utilization
- lack of simulator-agnostic description languages
- no repository for research results
- absence of teaching platform

1

Solution: Grid4GridResearch

Increase scalability of simulation tools
- use novel parallel, distributed and decentralized architecture

Unleash the power of idle CPU cycles
- create a desktop grid on a P2P overlay for topology simulation

Further Grid and overlay research
- provide framework for topology definition, reuse, dissemination and evaluation for repeated research

2

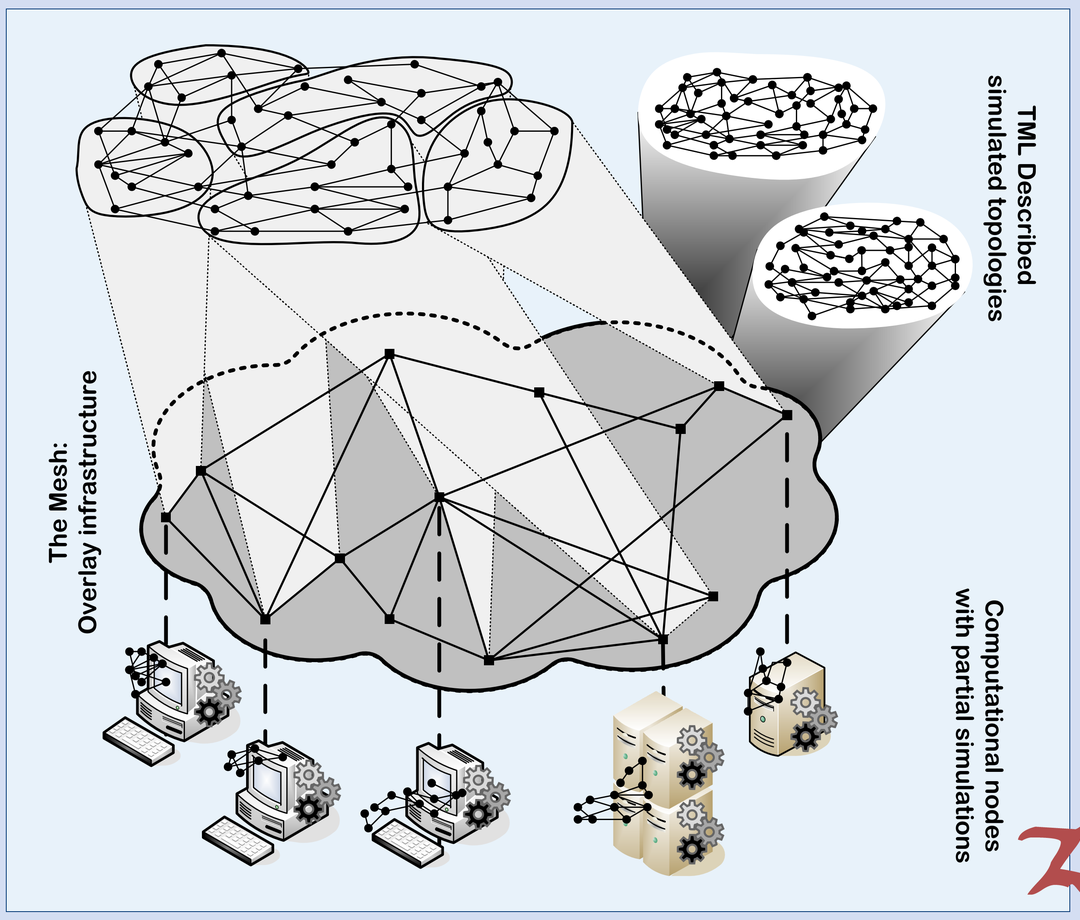
Grid4GridResearch: Architecture

The Mesh: Overlay infrastructure
- discover and aggregate cycle and storage donors

Parallel / Distributed simulations
- simulations broken into partial simulations assigned to nodes

Topology Modeling Language (TML)
- define simulated topologies, structure, content, behavior, policies

High-level services
- resource management, balancing, fault tolerance, user interaction



3

The Mesh: Overlay infrastructure

Extendable peer-to-peer architecture
- harness idle cycles of desktops, clusters, utility-computing
- gathers asymmetric participants with different capabilities

Hybrid structured/unstructured overlay
- structured: replicated topology descriptions, policies, results
- unstructured: partial simulations scheduled on any node

Hierarchical overlay
- super-peers aggregate resource information of neighbors

4

Parallel / Distributed simulations

Goals
- increase on simulation speed / increase on simulation size

Concurrent partial simulations
- partial simulations assigned to cycle donors
- exploit locality on exchanged messages

Weaker consistency possible
- local events do not interfere with remote partial simulations
- use of divergence-bounding and vector-field consistency

5

Topology Modeling Language (TML)

Current definition of simulated network topologies
- lack of expressiveness of topology description approaches
- lack of interoperability between simulators
- addressed with domain-specific language for topology definition

TML: a high-level description language
- definition of network nodes and behavior
- neighbor and routing tables
- protocol definition (entry, exit, resource discovery, routing, ...)
- no dependence on languages (e.g., Java) or simulator API

6

High-level services

Load-balancing
- efficient deployment of partial simulations
- favor co-location of sibling partial simulations
- migration of partial simulations

Fault-tolerance
- replication of partial simulations
- partial simulations state verification
- checkpointing and recovery of partial simulations

Grid Portal
- deployment of simulations and access point for results
- starting point for e-learning / e-science infrastructures

7

Conclusion

Improvement on Grid and overlay simulation
- increase on scalability of simulation tools
- deploy novel parallel, distributed and decentralized architecture
- leverage the power of idle remote CPU cycles
- offer new topology definition language
- facilitate grid and overlay research

8