A Novel Approach for Distributed Simulation of Wireless Mobile Systems*

Luciano Bononi and Gabriele D'Angelo

Dipartimento di Scienze dell'Informazione,
Università degli Studi di Bologna,
Mura Anteo Zamboni 7, 40126, Bologna, Italy
`{bononi,gdangelo}@cs.unibo.it`

Abstract. This position paper introduces the motivation and preliminary implementation issues of a distributed simulation middleware designed to increase the performance and speed-up in the distributed simulation of wireless systems characterized by mobile hosts. Topology changes due to simulated hosts' mobility map on dynamic causality effects in the "areas of influence" of each mobile device. We analyze the preliminary definition of a new dynamic mechanism for the runtime management and distributed allocation of model-components executed over a cluster of Physical Execution Units (PEUs). A migration mechanism dynamically adapts the topology changes in the wireless network to a reallocation of model components over the PEUs. The aim is the reduction of communication overheads, between the PEUs, required to distribute the event-messages between model components. The distributed simulation framework is based on HLA-compliant runtime infrastructure and preliminary, adaptive load-balancing and migration heuristics.

1 Introduction

A wide research work has been done in recent years in the field of tools and methodologies for modeling and efficient simulation of wireless systems [1,2,11, 12,13,15,18,20,22]. Among the relevant scenarios considered Cellular systems, PCS networks and the Mobile Ad Hoc Networks (MANETs) are gaining an increasing relevance. Wireless networks currently considered appealing for the analysis may include a potentially high number of simulated hosts. The simulation of every host may require a relevant computation time (e.g. due to simulation of protocol-stacks and applications running on top). This is often unpractical or impossible to simulate on a classical Von Neumann (mono-processor) architecture [17,18]. The simulation is unlikely to have success because of huge memory requirements and large amount of time required to complete.

Many practical experiences have demonstrated that a speed-up in the simulation of network systems is achievable using parallel and distributed models and architectures, i.e. a Parallel Discrete Event Simulation (PDES) approach [7,8],

---

* This work is supported by MIUR FIRB funds, under the project: "Performance Evaluation of Complex Systems: Techniques, Methodologies and Tools"

M. Conti et al. (Eds.): PWC 2003, LNCS 2775, pp. 829–834, 2003.

e.g. Glomosim [22] based on PARSEC [1], Maisie [18], parallel and distributed implementations based on Network Simulator (ns-2) [10,15,16,20] based on RTI-KIT [15], on ANSE/WARPED [14], on TeD [13], USSF over Warped/Notime [Rao99], Wippet [11], SWiMNET [2], and many others [10,12,18]. In order to exploit the maximum level of computation parallelism, many research activities dealt with dynamic balancing of logical processes' executions (both cpu-loads and virtual time-advancing speeds) by trading-off communication, synchronization and speed-up, both in optimistic and conservative approaches [4,5,9,19,21].

High Level Architecture (HLA) is a recently approved standard (IEEE 1516) dealing with component-oriented distributed simulation [3,6]. It defines rules and interfaces allowing for heterogeneous components' interoperability in distributed simulation. The definition of model components (formally known as federates) with standard management APIs brings to a high degree of model re-usability. The HLA standard defines APIs for the communication tasks and synchronization between federates. The simulation is supported by a runtime middleware (RTI). The RTI is mainly responsible for providing support for time management, distributed objects' interaction, attributes' ownership and many other optimistic and conservative management policies [3,6].

2 Work Motivation

Many approaches have been investigated in order to reduce the overhead effects of distributed synchronization and communication in both optimistic and conservative distributed simulations. The motivation for this communication-reduction approach is the frequent adoption of networked cluster of PCs, in the place of shared-memory or tightly-coupled multiprocessors, as the execution units of the distributed simulation, primarily for cost reasons. The high network latency in these clusters could play a fundamental role in determining the weight of communication and synchronization between the distributed model components. Solutions have been proposed, relying on the reduction of communication obtained when the update of a event- or state-information (e.g. event and/or anti-message) does not need to be flooded to the whole system, but is simply propagated to all the causally-dependent components. This is the basis of publishing/subscribing mechanisms for sharing state-information and event-notifications between causally dependent components [3,6,15].

Simulation models for wireless systems have to deal with at least two innovative concepts with respect to wired networks' models: *i)* the user-mobility and *ii)* the open-broadcast nature of the wireless medium.

Communication is commonly defined between neighbor hosts and determines a causal effect in the simulated system. The causal effect of communication may be extended to all the neighbor-hosts (even if their communication-links are partitioned over different logical channels) due to the broadcast nature of the wireless medium. A multi-hop communication between non-adjacent hosts can be considered as a communication based on local interaction between a chain of neighbor hosts. Thus, a high degree of causality in the simulation of the wireless hosts' communication is driven by the local-topology interaction (i.e. messages) between neighbor hosts. This interaction can be considered the most relevant causality effect to be modeled in many wireless system simulations (e.g. when evaluating MAC protocols, routing protocols, or physical interference effects).

Topology changes (due to simulated hosts' mobility) map on causality effects in the "areas of influence" of each mobile device, resulting in dynamically shaped causality-domains. Accurate simulation results would require accurate details to be modeled, and many fine-grained, low-level causal effects (i.e. events) to be kept into account in the simulation process. We define a dynamic system as a system where the interactions (i.e. the causal effects of events) are dynamically subject to fast changes driven by the system (and model) evolution over time. Given this definition, a wireless network (e.g. a MANET) can be considered a highly dynamic system. A static definition of publishing/subscribing lists, groups and causal domains could not be more convenient, in presence of a dynamic system, than implementing a complete state-sharing. A dynamic approach for the distribution of events and state-information (e.g. dynamic lists and groups updates) would lead to network communication overheads. In some scenarios, the cost of list-updates or fine-grained events' communication between a dynamically variable set of components could be traded-off with the migration cost needed to cluster the whole interacting components on a single Physical Execution Unit (PEU). This would be more attractive if the object migration could be implemented incrementally as a simple data-structure (i.e. state) transfer, and if the object interaction would be maintained for a significant time (time-locality). A certain degree of time-locality of local communication can be considered an acceptable assumption in many mobile wireless systems, depending on the motion model characteristics.

Our proposal is to define and investigate a simple mechanism allowing for Simulated Mobile Hosts' (SMHs) migrations in HLA-based distributed simulations. We realized a prototype migration framework, adopting it together with heuristic load-balancing and migration policies, whose aim is to dynamically partition and cluster the interacting SMHs among federates executed over a set of PEUs. SMHs in our approach have a common definition and can be migrated simply by serializing and transferring their state-information. This would realize an example of a prototype framework and an adaptive, tunable mechanism able to react to dynamic systems' behavior (like the mobile and wireless systems') under the communication-reduction viewpoint. In this position paper we sketch the prototype implementation of this mechanism and future work.

3 The Distributed Simulation Framework

We implement a parallel discrete event simulation of model components (federates) by using a set of physical execution units (PEUs) connected by a physical network (e.g. a networked cluster of PCs). The PDES simulator built to obtain an experimental evidence of our proposal is based on a distributed architecture made by a set of federates glued together by HLA middleware. Our approach at this level is mainly focused on the communication reduction between the PEUs where federates are executed. We implement a federate as a single logical process, managing and updating the state information (data structures) of a set of (at least one) Simulated Mobile Hosts (SMHs). We follow a time-stepped, conservative approach for ease of implementation, and given the high number of unpredictable, simulated entities with a high ratio of interactions performed. This means a conservative, fixed-lookahead time-management based on a HLA RTI implementation.

On top of the RTI we built a middleware extension called Generic Adaptive Interaction Architecture (GAIA). GAIA provides the interaction to the simulation core, the location and distribution data management, the random number generator, tracefile-logging and other simulation facilities. The HLA definition and components to handle a distributed simulation from the federate viewpoint is left untouched. The target of GAIA is to provide migration and service APIs to the simulation developer. We implement SMH models as code (they have a common code definition in the federates) with data structures to define and maintain the SMH state information. We choose to migrate the data structure, i.e. the state information of SMHs between federates. This requires to design and implement a supporting middleware for the data structure management and distribution of SMH entities between HLA federates. Our models can be simply executed as HLA federates in the static approach. If it is required to exploit migration, then the models should be defined to deal with migration APIs provided by GAIA on the top of HLA middleware. The equivalent of data distribution management and object ownership in the HLA meaning has been re-implemented in the GAIA framework. This allow us to deal with controlled overheads and ad hoc implementation for our purposes. In the current release, GAIA includes only prototype data-location and distribution-management services whose overheads have been preliminarily considered as an implementation issue. Currently, the GAIA framework includes also simple heuristic functions defined to evaluate the migration of SMHs between PEUs. The migration is performed to the PEU which is the target of the majority of interactions required. The steady state behavior of the proposed heuristic in isolation would lead to a long-term concentration of the SMHs over a restricted set of the available execution units, because the adaptive effect is focused on the "external" communication overhead. For this reason, we introduced the migration heuristic on the top of a simple load-balancing policy implemented by the GAIA middleware supporting the set of federates.

4 Preliminary Results

In our preliminary tests, we assume a high number of simulated mobile hosts (SMHs), each one following a mixed variation of the Natural Random and Random Waypoint (RWP) motion models. This motion model gives unpredictable and uncorrelated mobility pattern of SMHs under the motion viewpoint. In this way, any heuristic definition governing the mechanism cannot rely on any assumption about the motion correlation and predictability of SMHs. Space is modeled as a torus-shaped 2-D grid-topology, populated by a constant number of SMHs. In this way the clustering of SMHs is not trivially determined by high concentration in small areas. The modeled communication between SMHs is a constant flow of ping messages (i.e. constant bit rate), transmitted by every SMH to its nearest device. We ran our simulation experiments over a variable set of PEUs equipped by Dual-Pentium III 600Mhz, 256MB RAM, connected by a FastEthernet (100Mb/s) LAN. Preliminary results shown a speed-up up to 23% given by adaptive distributed simulation with respect to a static distributed simulation, for 900 SMHs simulated over 3 PEUs.

5 Conclusions and Future Work

We propose an adaptive framework, named Generic Adaptive Interaction Architecture (GAIA), for the dynamic allocation of model entities (ME) in a HLA-based framework for distributed simulations. GAIA is based on runtime migration and load-balancing policies, to reduce in adaptive way the amount of external communication between the PEUs. We tested our mechanism for simple, sub-optimal migration and load-balancing heuristics in the testbed simulation of a prototype mobile wireless system, characterized by Simulated Mobile Hosts (SMHs). The runtime mechanism adapts the MEs' allocation over the PEUs to the dynamic interactions of SMHs. Preliminary results demonstrated the effectiveness of the proposed mechanism and performance enhancements, with controlled overhead for the worst-case scenario defined. We expect increasing performance to be obtained by optimizing the framework code, by tuning the mechanism's heuristics on the hypothesis and assumptions related to real system models.

Our future work will extend our analysis from a qualitative to a quantitative one. Additional efforts will be done in the code-optimization for the migration mechanism implemented. We will study new Data Distribution Management (DDM) implementations (both centralized and distributed) to enhance the message reduction and filtering. We will also design new HLA-based interaction management and filtering, development of detailed heuristics based on analysis of multiple metrics and parameters, many different hardware and network architectures. The ad hoc network scenario will be extended to deal with protocols and complex SMHs' behaviors. Specifically, the migration mechanism will be evaluated with respect to many dynamic factors to be modeled, in addition to the host mobility (e.g. dynamic communication-session establishment). The migration-based approach of this work could be extended to a wide set of simulations where the sequential approach gives low performance (e.g. multi-agent systems, genetic and molecular systems, P2P models).

References

1. Bagrodia, R., Meyer, R., Takai, M., Chen, Y., Zeng, X., Martin, J., Song, H.Y.: PARSEC: a parallel simulation environment for complex systems, IEEE Computer, 31(10):77–85, October 1998

2. Boukerche, A., Fabbri, A.: Partitioning Parallel Simulation of Wireless Networks, Proc. of the 2000 Winter Simulation Conference (WSC), 2000
3. Dahmann, J., Fujimoto, R. M., Weatherly, R. M.:, High Level Architecture for Simulation: an update, Winter Simulation Conference, December 1998.
4. Das, S. R.: Adaptive protocols for Parallel Discrete Event Simulation, Proc. of Winter Simulation Conference, 1996.
5. Deelman, E., Szymanski, B.K.: Dynamic load balancing in parallel discrete event simulation for spatially explicit problems, Proc. of the twelfth workshop on Parallel and distributed simulation PADS'98, July 1998
6. Defence Modeling and Simulation Office (DMSO): High Level Architecture RTI Interface Specification, Version 1.3, 1998
7. Ferscha, A.: Parallel and Distributed Simulation of Discrete Event Systems, In Handbook of Parallel and Distributed Computing, McGraw-Hill, 1995
8. Fujimoto, R.M: Parallel and Distributed Simulation Systems, Wiley & Sons, 2000
9. Gan, B.P., Low, Y.H., Jain, S., Turner, S.J., Cai, W., Hsu, W.J., Huang, S.Y.: Load balancing for conservative simulation on shared memory multiprocessor systems, Proc. of 14-th Workshop PADS'00, May 2000, Bologna, Italy
10. Jones, K.G., Das, S.R.: Parallel Execution of a sequential network simulator, Proc. of the 2000 Winter Simulation Conference, 2000
11. Kelly, O. E., Lai, J., Mandayam, N. B., Ogielski, A. T., Panchal, J., Yates, R. D.: Scalable parallel simulations of wireless networks with WiPPET: modeling of radio propagation, mobility and protocols, MONET, v.5, n.3, Sep. 2000.
12. Liu, W.W., Chiang, C.C., Wu, H.K., Jha, V., Gerla, M., Bagrodia, R.: Parallel simulation environment for mobile wireless networks, Proc. of Winter Simulation Conference 1996.
13. Perumalla, K., Fujimoto, R., Ogielsky, A.: TeD – A language for modeling telecommunications networks, Performance Evaluation Review 25(4), 1998
14. Rao, D.M., Wilsey, P.A.: Parallel Co-simulation of Conventional and Active Networks, Proc. of MASCOTS'00, August 2000
15. Riley, G.F., Fujimoto, R.M., Ammar, M.H.: A generic framework for parallelization of network simulations, Proc. of MASCOTS'99, College Park, MD, October 1999
16. Riley, G.F., Ammar, M.F., Fujimoto, R., Perumalla, K., Xu, D.: Distributed Network Simulations using the Dynamic Simulation Backplane, Proc. of MASCOTS'01.
17. Riley, G.F., Ammar, M.H.: Simulating Large Networks How Big is Big Enough? Proc. of Intern'l Conf. on Grand Challenges for Modeling and Simulation, Jan 2002.
18. Short, J., Bagrodia, R., Kleinrock, L.: Mobile wireless network system simulation, Wireless Networks 1, August 1995, (pp. 451–467).
19. Som, T.K., Sargent, R.G.: Model structure and load balancing in optimistic parallel discrete event simulation, Proc. of 14-th workshop PADS'00, Bologna, Italy
20. UCB/LNBL/VINT The NS2 network simulator, http://www.isi.edu/nsnam/ns/
21. Vee, V-Y, Hsu, W-J: Locality-preserving load-balancing mechanisms for synchronous simulations on shared-memory multiprocessors, Proc. of 14-th workshop PADS'00, May 2000, Bologna, Italy
22. Zeng, X., Bagrodia, R., Gerla, M.: GloMoSim: A library for parallel simulation of large-scale wireless networks, Proc. of 12-th workshop PADS'98, May 1998, Alberta, Canada.