Proceedings of the 2012 Winter Simulation Conference
C. Laroque, J. Himmelspach, R. Pasupathy, O. Rose, and A.M. Uhrmacher, eds

THE EFFECTS OF SPEEDUP AND NETWORK DELAYS ON DISTRIBUTED SIMULATIONS

Alessandra Pieroni and Giuseppe Iazeolla

University of Rome TorVergata
Via del Politecnico, 1, 00133
Roma, Italy

ABSTRACT

A simulation model can be seen as consisting of a set of sub-models or federates. In local simulation (LS), a single federate exists that simulates the entire system and is run by a single host. In distributed simulation (DS), various federates that simulate distinct parts of the system are run by separate hosts that operate in parallel and are connected by a computer network (LAN, MAN or WAN, or a composition thereof).
Predicting at design-time the convenience of implementing the DS version of the LS can be of interest. Indeed, the development of a DS system is a complex and expensive task, since of the cost of achieving the necessary know-how of the distributed simulation standard, the cost of the extra-lines of code to develop for each Federate, the number of design alternatives to face (in terms of simulator partitioning, host capabilities, etc.).
This paper introduces a performance model to support the evaluation of DS convenience before implementation.

1 PROBLEM STATEMENT

Assume a local simulator (LS) of a given system Σ is available and that we wish to turn it into a distributed simulator (DS). In the DS case, the LS is partitioned into segments called federates, each federate being run by a separate host. Fig.1 shows the two federate case, with N_S denoting the network for the exchange of synchronization messages and N_D the one for data messages.

Before implementing the DS (i.e., at design-time) we wonder: will the DS execution time be shorter than LS one? In some cases the DS may run slower than the equivalent LS.

To answer this question we are to consider that the execution time of a distributed simulation system depends on 3 interacting factors: 1) the speedup "S" (or run-time gain) resulting from the partitioning of the local simulator into federates. 2) The communication overhead of synchronization-messages. 3) The communication overhead of data-messages.

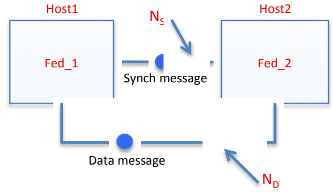

Fig.1: DS system with two Federates

Let us denote by S(N) the maximum speedup that can be achieved using N processors, and by Q the fraction of computation that is inherently serial. According to Amdahl's law even with an arbitrarily

978-1-4673-4781-5/12/$31.00 ©2012 IEEE

large number of processors ($N \rightarrow \infty$), S(N) can be no larger than the inverse of the inherently serial portion Q of LS.

The two communication delays play a counteracting effect on the speedup. The speedup, indeed, depends on the amount of federate code that can be effectively computed in parallel, while the two communication overheads depend on the amount of communication between computations and on the efficiency of the N_S and N_D networks. The combination of all such factors makes it very hard to predict the advantage of transforming a local version of the simulator (LS) into a distributed version (DS).

In order to predict the advantage of transforming a LS into its equivalent DS, a decision procedure is proposed to evaluate the DS convenience before implementation. The procedure is guided by a Performance Model (PM) of the DS, and investigates the effects of the three interacting factors: 1) the speedup S, 2) the communication overhead of synchronization-messages, 3) the communication overhead of data-messages. The PM assumes the DS is based on the HLA protocol standard and middleware.

To perform an example prediction of the DS(Σ) execution time (T_{DS}), we developed the OMNet++ simulation version of the PM model.

The OMNet++ simulation was carried-out in two scenarios A and B: Scenario A being one in which the fraction Q of inherently serial computation was high and Scenario B in which Q was low. The simulation results are shown in Tab.1. The first column reports the execution time T_{LS} of the real local simulator (that was implemented in Java). The second column gives the execution time T_{DS} of the distributed simulator predicted by the OMNet++. The third column gives the time of the real DS (that was implemented in Java+HLA). Such a column thus provides a validation of the OMNet++ results, and shows how the predicted results adequately match the real ones.

Tab.1 also shows how in the Scenario B the distributed simulator outperforms the local one.

Indeed, in such a Scenario the DS execution time (T_{DS}) is much lower than the LS time (T_{LS}).

Finally by using the expression $S = T_{LS}/T_{DS}$, the results in Tab.1 were used to obtain the speedup results shown in Tab.2, witch shows that a quite good speedup (S = 2.64) is obtained in the B Scenario. In other words, in this case the run-time gain obtained by the parallel execution on two hosts compensates for the data and synch communication overheads.

In the scenario A, instead, the parallelism does not yield a sufficient run-time gain to compensate for the communication overheads, and the resulting speedup (S = 0.08) is practically irrelevant.

Tab. 1 Execution-time results

	T_{LS}	**PM results (OMNet++ predictions)**	**PM validation (real DS measurements)**
A (high Q)	0.7s	T_{DS} = 8.3s	T_{DS} = 8.2s
B (low Q)	33 min	T_{DS} = 12.5 min	T_{DS} = 12.0 min

Tab. 2 Speedup results

A: High Q	S = 0.08
B: low Q	S = 2.64

The Tab.1 and 2 results are used by the DS decision procedure to decide at design-time whether to remain on the LS version of the simulator or carry out the implementation of its DS version.
