System for Transport Protocol Evaluation with Automatic Calculation of Statistical Accuracy and Distributed and Parallel Execution

(Poster Abstract)

Aleksandar Milenkoski
SCSIT, UACS
Treta makedonska brigada, bb, 1000
Skopje, Macedonia
+38970336746
milenkoski@uacs.edu.mk

Biljana Stojcevska
SCSIT, UACS
Treta makedonska brigada, bb, 1000
Skopje, Macedonia
+38978455132
stojcevska@uacs.edu.mk

Oliver Popov
DSV, SU
Forum 100, Isafjordsgatan 39,
SE- 164 40 Kista, Sweden
+4686747237
popov@dsv.su.se

ABSTRACT
The paper deals with the architecture and the performance of a system for gathering and processing simulation data where use case are communications transport protocols. The system is based on the ns-2 network simulator and the tcpeval tool for the evaluation of TCP and TCP related protocols. The work promotes the concept of a controlled simulation replication as central to achieve statistical accuracy of the simulation results. The later assures that in each case of the specific set of simulation runs, their number does not exceed the minimal one required by the desired and predefined accuracy. Moreover, the use of inter-process and inter-thread communication provided by Open MPI and OpenMP makes the execution of the system possible over a multiprocessor distributed architecture that eventually reduced the time needed to achieve the preferred precision. While the later work is in an early stage, the initial results of the benchmark tests indicate significant gains in time based on the metrics native to tcpeval.

Categories and Subject Descriptors
I.6.3 [**Simulation and Modeling**]: Applications

I.6.6 [**Simulation and Modeling**]: Simulation Output Analysis

I.6.8 [**Simulation and Modeling**]: Types of Simulation – *Distributed, Parallel*

General Terms
Measurement, Performance, Design, Experimentation, Verification.

Keywords
TCP evaluation, distributed execution, statistical accuracy, parallel execution

SIMUTOOLS 2011, March 21-25, Barcelona, Spain

DOI 10.4108/icst.simutools.2011.245532

1. INTRODUCTION
In the last decade, the landscape of network protocols and applications has gone through significant changes. Numerous and noteworthy research efforts have been made in the modification and eventual adaptation of the standard TCP protocol with respect to its performance in various novel network topologies, link properties and traffic characteristics [3]. Considering the ever evolving nature of the Internet, for instance the influx of protocols to run multimedia applications, there is a growing need for their proper evaluation and validation.

In this paper we introduce simulation control and data processing system which offers both topology generation with complex traffic models and statistically valid results. The proposed system combines the use of tcpeval [1] as TCP evaluation suite and ns-2 [13] as network simulator. It offers controlled simulation environment with procedures for automatic calculation of the statistical precision of the results [7]. The user-friendly web interface is an additional asset. Furthermore, to speed up the evaluation time, we reach for the functionalities of Open MPI (Message Passing Interface) [5] and OpenMP which are capable of providing a distributed processor environment. The work builds on the system described in [12] and is part of the continuous effort to extend its functionality and benefits.

In the Section 1 termed as Introduction, the need for further research in the area of communication transport protocols is underlined (especially TCP and TCP related ones). In addition, it enumerates the characteristics of the system for evaluation of the aforementioned protocols and the objectives for the enhancement of the system, which are addressed in detail in Section 2. The performance and the evaluation of the "new" system are subject of Section 3, while the last section summarizes the benefits and outlines the ways for further extensions, as well as venues for improving the computation complexity of the whole platform.

2. THE ENHANCEMENTS
The system presented in [12] uses a predefined number of experiment replication sets to obtain statistically accurate results, where each set consists of 12 simulation scenarios. The rudimentary system includes only total throughput, average throughput, and inter-protocol fairness. The parameterization of the system is performed through a lengthy set of command line

arguments that affects its user-friendliness. Finally, the system is designed to run on single processor architecture which significantly decreases its efficiency.

In order to resolve and overcome some of the enumerated problems and issues we introduce:

Run-time calculation of the number of replications needed to achieve statistical accuracy of the gathered results: The system automatically stops the iterative simulation when the confidence level and the relative precision defined by the user are achieved.

Widened set of processed evaluation results to all metrics measured by tcpeval: As suggested by [4], the set of the processed simulation metrics is enlarged to include all evaluation metrics recorded by tcpeval.

Inter-process and inter-thread communication for simultaneous execution of simulation scenarios in multi-core and multi-processor environment: We define 5 levels of simultaneous execution of simulation scenarios:

- Level 1: There are 3 MPI processes, including the MPI master process. Each of the processes sequentially executes 4 simulation scenarios.
- Level 2: In this case 12 MPI processes are created including the MPI master process. Each of the processes executes one simulation scenario.
- Level 3: We create 3 OpenMP threads, where each thread sequentially executes 4 simulation scenarios.
- Level 4: We create 12 OpenMP threads and each thread sequentially executes one simulation scenario.
- Level 5: We create 3 MPI processes which create 4 OpenMP threads. Each thread sequentially executes 1 simulation scenario. Hence, this level represents hybrid MPI – OpenMP approach to execution of simulation tasks.

The current implementation of the system requires that all nodes use shared file system.

User-friendly web-interface: The objectives are to have a simple, intuitive web interface that provides:

- Straight-forward definition of the system's parameter values
- Evaluation of the user input
- Execution of the simulation system
- Links to the obtained results

3. Performance Evaluation of the System's Operation in Distributed Nodes Environment

In order to monitor the results of the simultaneous execution of ns-2 simulation scenarios in multi-processor architectures, a series of "what-if" analysis in terms of execution speed-up and total execution time is employed. Only pure MPI implementations on Level 1 and Level 2 are under consideration. Consequently, Level 1 is observed in 2 and 3 node architecture, while Level 2 in 2, 4, 6, 8, 10 and 12 node architecture.

We collected traces by using MPITrace [2] from multiple executions of the evaluation system on OpenSUSE 11.0 operating system, with Linux kernel version 2.6.16-21. The hardware platform for the simulation consisted of Intel Pentium D microprocessor with clock speed of 3 GHz, 512MB cache and 2GB main memory. To measure the workload assigned to the processor and its performance, the kernel was patched with the Perfctr 2.6.26 performance monitoring tool [10]. The trace-driven simulator in our case was Dimemas [6] that was able to reconstruct the behavior of the system on diverse distributed machine architectures. Paraver [11] provided the corresponding statistical metrics and visualization of the output data.

The executed simulation experiments modeled single and multiple-bottleneck networks with bottleneck bandwidth of 10 Mbps. The duration of each simulation run was set to 100 sec. The simulated traffic includes one forward and one reverse long-lived FTP flows, two voice flows, 10 video flows and sporadic short-lived HTTP traffic. The TCP variants used were SACK [8] and TCP Vegas [3] with an infusion of packet error rate that ranged from 0 to 10 percent.

Three different categories of processors qualified as low, medium and high performance were modeled with performance benchmarks and appropriate categorization drawn from [9]. The performance ratios for the three categories of processors are presented in Table 1.

Table 1. Processor performance ratios

Processor Category	Range of Processor Performance Ratio
Low performance	[1.0 – 1.26]
Medium performance	[1.27 – 2.38]
High performance	[2.39 – 15.1]

The processors with different performances and communication latencies (in the interval of 0 to 100 msec are normally) distributed on single processor nodes which comprise the whole distributed environment. The generated MPI tasks were mapped according to the default MPI by-node task mapping algorithm in a round-robin fashion.

The execution speedup is defined as a ratio between the sum of the task durations on the distributed processors and the total execution time of the distributed application.

In a low performance, but highly power-balanced, processor architecture which consisted of 3 nodes, the maximum speedup is 2.8 when 3 simulation tasks are executed (Level 1). On the contrary, due to the wide range of high performance processor ratios (Table 1) that are assigned to small number of nodes, indicative were fairly large variations in the performance with respect to the overall processor architecture. Hence, the records of the simulation experiments indicate fairly long duration of the collective MPI communications which were needed to finalize one replication set. The phenomenon was exhibited through considerably long idle time in many of the generated MPI tasks. In this case, the speedup was only 1.05 which clearly reduced any advantages that simultaneous execution might bring.

As expected, the total execution time of all three MPI tasks is the shortest one when high performance processors are employed. The minimal completion time of a replication set is 29.310 seconds on 3 high performances nodes. On the other hand, the

greatest completion time of a single replication set is 410.953 seconds on 2 low performance nodes.

The performance analysis of the system in the case of 12 MPI tasks (Level 2), showed maximum speed-up of 9.99 when executed on 12 low performance and well power-balanced processor architecture. Conversely, the minimal speedup is recorded with 4 high performance processors. It should be noted that due to the better balance in the distribution of processor execution power, the 12 node high performance architecture achieves almost twice is high speed-up of 2.67 than its 4 node equivalent.

The shortest execution time of one replication set (Level 2) is 11.619 seconds on 12 high performance processors, while the longest one is 288.652 seconds on 2 low performance processors. For comparison, the longest simulation time in a single processor environment is 627 seconds.

The discrepancy between the number of tasks (12) and the corresponding small number of processors assigned to the tasks which results in long execution times is to be expected due to the computational inefficiencies stemming from oversubscribed nodes. Maintaining a proper balance between the potential and the capability of the available hardware platform and the number of generated tasks to be completed appears to be essential to the good performance of the entire system.

4. CONCLUSION

The enhanced evaluation system introduces four new general features: run-time calculation of the statistical precision of the evaluation results, larger set of processed evaluation metrics, Open MPI and OpenMP implementation for simultaneous simulation experiment execution, and user friendly web interface.

The automatic calculation of the statistical accuracy, as defined by the user, generates results within the smallest possible time.

The eventual speedup in the execution of the replication sets posits the need for implementing the concept of distributed and parallel computing. Initial experiments and simulation runs showed speedup in the execution in the range of 1.05 up to 9.99. The improvement in the performance of the simulation platform also depends on the specific hardware architecture and its fine tuning between the number of tasks and the number of processors. The preliminary results of the ongoing research with respect to the most efficient and effective design of the evaluation system point to the organization based on tightly clustered and well-balanced group of processing nodes.

This work can be extended in several ways. More extensive system benchmarks could be done, especially on the impact of disk latency on the file I/O operations of the system. Furthermore, to avoid long execution times, reducing the complexity of the data processing algorithms implemented in the system remains one of the priorities, as well as reducing the overhead of I/O operations.

5. REFERENCES

[1] An NS2 TCP Evaluation Tool. Retrieved October 19, 2010, from NEC Labs China: http://labs.nec.com.cn/tcpeval.htm.

[2] Trace generation: just some examples. Retrieved October 19, 2010, from Barcelona Supercomputing Center: http://www.bsc.es/plantillaA.php?cat_id=492.

[3] Brakmo, L. S., O'Malley, S. W., and Peterson, L. L. 1994. TCP Vegas: new techniques for congestion detection and avoidance. *SIGCOMM Comput. Commun. Rev.* 24, 4 (Oct. 1994), 24-35. DOI= http://doi.acm.org/10.1145/190809.190317.

[4] Floyd S. 2008. Metrics for the Evaluation of Congestion Control Mechanisms. RFC 5166.

[5] Gabriel E., *et al.* 2004. Open MPI: Goals, Concept, and Design of a Next Generation MPI Implementation. In *Proceedings of the 11th European PVM/MPI Users' Group Meeting* (Budapest, Hungary, September 2004).

[6] Girona, S. and Labarta, J. 2000. Sensitivity of Performance Prediction of Message Passing Programs. *J. Supercomput.* 17, 3 (Nov. 2000), 291-298. DOI= http://dx.doi.org/10.1023/A:1026567408307.

[7] Hassan M. and Jain R. 2004. *High Performance TCP/IP Networking: Concepts, Issues, and Solutions*. Pearson Prentice Hall, United States.

[8] Mathis M., Mahdavi J., Floyd S., Romanow A. 1996. TCP Selective Acknowledgment Options. RFC 2018.

[9] PassMark Software – CPU Benchmark Charts. Retrieved October 19, 2010, from PassMark Software - PC Benchmark and Test Software: http://www.cpubenchmark.net/.

[10] Petterson, M. Linux x86 Performance-Monitoring Counters Driver. Retrieved October 19, 2010, from Computing Science Department, Uppsala University, Sweden: http://user.it.uu.se/~mikpe/linux/perfctr/.

[11] Pillet V., Labarta J., Cortes T., and Girona S. 1995 "Paraver: A Tool to Visualize and Analyze Parallel Code." Technical report. WoTUG-18.

[12] Stojcevska B., Popov O., Milenkoski A. 2010. Iterative System For Simulation of E2E Transport Protocols In Heterogeneous Networks. In *Proceedings of the 7th EUROSIM Congress on Modeling and Simulation* (Prague, Czech Republic, 6-10 September, 2010).

[13] The Network Simulator - ns-2. Retrieved October 19, 2010, from Information Sciences Institute: http://www.isi.edu/nsnam/ns/.