STSM Scientific Report

COST STSM Reference Number : COST -STSM-IC1001-10266
Period: 2012-11-06 to 2012-12-06
COST Action: IC001

Information:

- **Purpose of the STSM : (from the STSM proposal)**

  The purpose of this STSM is to study and develop " A Scalable Protocol, based on Partially Replicated Transactional Systems and Resilient Transport Layer in large multicore chips (beyond 1000 cores), for seamless execution of NonDAG graph parallel applications in the presence of multiple failures"

The following tasks will be investigated and decisions will be proposed for providing feasible solutions:

1) Self-reconfiguration and Self- healing in 2D multicore systems (beyond 1000 cores).
The goal is to achieve maximum throughput, minimum latency and max. execution time reduction ratio (RR) for parallel applications executing on scalable multicore systems.

2) Parallel applications remapping on the multicore chips in the presence of anomalies.
The Resilient Transport Layer will control the adaptive task re-mapping (migration) and reconfiguration of the resources in scalable multicore chips. Different Resilient Transport Layers, for adaptive parallel application mapping on the scalable multicore chips, will be investigated and validated.

2.a) Applications represented as a DAG graph and utilizing the hierarchical organization of the transport layer

2.b) Applications, represented by a NonDAG graph and using the Software Transactional Memory (STM) for accessing the Shared Memory Resources and a Resilient Transport Layer for remapping tasks in the presence of multiple failures.

- **Description of the work carried out during the STSM**

During the STSM, D.R. Avresky had the opportunity to work in close synergy with the researchers of the group of Prof. Bruno Ciciani. This has led to the identification of a novel research line, pursuable by combining the expertise of the two research teams: the study of techniques for optimally mapping the tasks of parallel applications deployed on a highly scalable, fault-tolerant transactional memory platform.

The problem of dynamically "re-mappng" generic application tasks on a multi-core system, by relying on a Resilient Transport Layer has, in fact, already been investigated by D.R. Avresky in his previous work [5, 6, 7, 8] This work introduced a novel Self-Recovering strategy , based on run-time failure aware techniques, and aimed at guaranteeing seamless termination and delivering of the expected results despite multiple node and link failures in a 2D mesh topology.
It has been demonstrated, based on a simulation analysis, that the proposed technique is able to re-map the tasks of faulty nodes in a bounded number of steps. The proposed technique is allowing to bypass multiple nodes, routers and links failures with a predictable number of hops.

This solution, however, assumed that tasks interacted via message-passing, and had no shared state. During this STSM, we investigated the idea of developing an analogous mechanism for a large scale transactional memory system, deployed on machines equipped with 1000s of cores, or on distributed transactional memory platforms (e.g. for cloud environments).

The group of Prof. Bruno Ciciani has been working for years on the design of highly efficient transactional memory systems, both in centralized [1] and in replicated/distributed settings [2]. During the STSM, we focused specifically on a protocol for Distributed Transactional Memory (DTM) systems that has been recently published by members of Prof. Bruno Ciciani's group, namely GMU [3, 4].

GMU is a genuine partial replication protocol for transactional systems, which exploits an innovative, highly scalable, distributed multiversioning scheme.
Unlike existing multiversioning-based solutions, GMU does not rely on a global logical clock, which represents a contention point and can limit system scalability. Also, GMU never aborts read-only transactions and spares them from distributed validation schemes.

GMU guarantees the Extended Update Serializability (EUS) isolation level. This consistency criterion is particularly attractive as it is sufficiently strong to ensure correctness even for very demanding applications (such as TPC-C), but is also weak enough to allow efficient and scalable implementations, such as GMU. Further, unlike several relaxed consistency models proposed in literature, EUS has simple and intuitive semantics, thus being an attractive, scalable consistency model for ordinary programmers.

A first relevant research question that was addressed during this STSM was related to how to adapt GMU for ensuring consistency semantics on multi-core platforms. In fact, GMU was originally developed for a DTM platform, in which nodes communicate via message passing over asynchronous channels. The system model assumed by GMU is certainly attractive for his generality, and it lends itself naturally to deal with non-cache coherent NUMA architectures. However, the problem of how to adapt GMU to operate efficiently in a cache coherent multicore systems is an open and interesting research question. In fact, thanks to its high scalability, GMU appears as a very attractive solution to enhance the fault-tolerance of massively parallel (non-distributed) systems via strongly consistent partial replication techniques.

Beyond this, we explored the idea of combining the fault-tolerant properties of GMU, with the

Resilient Transport Layer approach [5, 6, 7, 8] to provide a deadlock-free mapping of parallel applications, represented as a NonDAG Graph, on multicore systems (in the presence of anomalies). GMU will allow parallel applications to be mapped on multicores when different tasks communicate via a (D)TM, ensuring high availability of data via partial replication and deadlock freedom (thanks to the transaction abstraction). Partial replication will allow re-mapping the lost portion of the tasks' state (partial replicas), due to the failures. In this way, the Resilient Transport Layer will save a significant time, when a given stream leader needs to re-map entire its tasks (streams) to a fault-free core(s). This feature will ensure a seamless execution of parallel applications, represented as a NonDAG graph, on large scalable multicore systems (beyond 1000 cores). Performance analysis of the proposed approach, for seamless mapping parallel applications, based on the Partial Replicas and the Resilient Transport Layer, will be carried out.

. The major steps of the proposed technique, in this research work , are the following:

- Fault-Tolerant and Adaptive Routing in Scalable Multicore (beyond 1000 cores) systems
- System Model and Nearby Strategy for Adaptive-Fault Tolerant Routing
- Adaptive Task Mapping in Scalable Multicore Systems
- Resilient Transport Layer for Mapping Applications, represented by a General Graph, onto Scalable Multicores (beyond 1000 cores)
- Parallel Applications, represented by General Graphs
- A Scalable Protocol, based on Partially Replicated Transactional Systems and Resilient Transport Layer, in large multicore chips (beyond 1000 cores), for seamless execution of NonDAG graph parallel applications in the presence of multiple failures.

We identified a set of tasks that will be fulfilled for the realization of the proposed scalable protocol for mapping NonDAG graph of parallel applications onto large multicore chips:

1) Algorithms' enhancement of the Partially Replicated Transactional System for providing an interface to the Resilient Transport Layer .
2) Upgrading and extending the set of algorithms of the Fault-Tolerant and Recovery Strategy for implementing the interface to the Partially Replicated Transactional Systems, which will be obtained through:
2.1 - Hierarchical Organization and Commitment
2.2 - RealTime Failure Detection
2.3 - Fault-Tolerant Nearby Strategy
2.4 - NonDAG graph mapping in the presence of multiple failures
2.5 - Validation of the NonDAG graph's mapping onto multicore system:
- computation and communication time of NonDAG parallel application;
- reduction ratio of the execution time (Speed Up) of NonDAG parallel application depending on the size of the transferred data, which is based on the results (number of hops and latency) obtained by the Nearby Strategy of the Resilient Transport Layer.

Major Features of the proposed Scalable Protocol for mapping NonDAG Parallel Applications on large multicore systems:

a) Transparency
- Adaptive remapping and seamless execution of NonDAG Parallel Applications , in the presence of multiple failures, in multicore systems (beyond 1000);

b) **Consistency** due to GMU (distributed multiversion concurrency control algorithm) and Extended Update Serializability (EUS) consistency criteria;

c) **Deadlock free solution is guaranteed at two levels:**
- Level 1 - Resilient Transport Layer for message passing in 2D mesh multicore systems, based on the virtual channels;
- Level 2 - Partially Replicated Transactional System, which is utilizing GMU and EUS;

Future collaboration with host institution (if applicable)

During the STSM, had been discussed with the host institution and mutually agreed to prepare a proposal for FP7/FP8 Calls in this challenging research area, specially in Cloud Computing .

Foreseen publications/ articals resulting or to result from STSM

We are working for as joint paper, in which we will propose a scalable protocol, based on Partially Replicated Transactional Systems and Resilient Transport Layer in large multicore chips (beyond 1000 cores), for seamless execution of NonDAG graph parallel applications in the presence of multiple failures.

Conformation by the host institution of the successful execution of the STSM

The collaboration has been useful and it allowed to lay the foundations for the design of a new scalable protocol for parallel applications in the case of concurrency on shared data.

Other comments
The mission had been postponed from the initial data, due to the personal reasons and finding better timing for the host institution.

1. P. Di Sanzo, B. Ciciani, R. Palmieri, F.Quaglia and P. Romano,
   Analytical Modeling of Commit-Time-Locking Algorithms for Software Transactional Memories,
   36th International Computer Measurement Group Conference (CMG), Orlando, Florida, USA, CMG, December 2010

2. R. Palmeri, P. Romano and F. Quaglia,

AGGRO: Boosting STM Replication via Aggressively Optimistic Transaction Processing, Proc. 9th IEEE International Symposium on Network Computing and Applications (NCA), Cambridge, Massachussets, USA, IEEE Computer Society Press, July 2010.

3. S. Peluso, P. Romano and F. Quaglia,
   Genuine replication, opacity and wait-free read transactions: can a STM get them all?,
   4th Workshop on the Theory of Transactional Memory (WTTM), Madeira, Portugal, July 2012

4. S. Peluso, P. Ruivo, P. Romano, F. Quaglia and L. Rodrigues,
   When Scalability Meets Consistency: Genuine Multiversion Update-Serializable Partial Data Replication,
   Proc. 32nd IEEE International Conference on Distributed Computing Systems (ICDCS), Macau, China, IEEE Computer Society Press, June 2012.

5. F. Chaix, D. Avresky, N. Zergainoh, and M. Nicolaidis, “Fault-tolerant deadlock-free adaptive routing for any set of link and node failures in Multi-Cores systems,” in IEEE International Symposium on Network Computing and Applications,july 2010.

6. F. Chaix, D. Avresky, N. Zergainoh, and M. Nicolaidis, “A fault-tolerant deadlock-free adaptive routing for On Chip interconnects,” in Design, Automation and Test in Europe Conference and Exhibition, march 2011

7. G. Bizot, D. Avresky, F. Chaix, N. Zergainoh, and M. Nicolaidis, “Self-recovering parallel applications in multi-core systems,” in Network Computing and Applications (NCA), 2011 10th IEEE International Symposium on, aug. 2011, pp. 51 -58.

8. Gilles Bizot, Dimiter Avresky, Fabien Chaix, Nacer-Eddine Zergainoh and Michael Nicolaidis , ”Adaptive Mapping of Parallelized Application (fork-join DAG) on Multicore System in the presence of Multiple Failures.” in IEEE DPDNS Workshop in conjunction with IEEE IPDPS Int. Symposium - May, 2012.