Project Meeting - Scalable Dependability

CAPES PVE Project
Coordination: Fernando Dotti (PUCRS)
Visiting Professor: Fernando Pedone (USI-Suíça)
Present institutions: USI - Suíça, PUCRS, UFSC, UNB, UFPR, UFU

Faculdade de Informática, PUCRS

May 9, 2016

Program

- Venue: PUCRS, Facin, Building 32, Room 516
- May 16, 14:30, PhD viva (Odorico M. Mendizabal)
- May 17, 9:00, see agenda - Workshop on Scalable Dependability

Project: Scalable Dependability

Project Description

Many current online services have stringent availability and performance requirements. High availability entails tolerating component failures and is typically accomplished with replication. For many online services, high performance essentially means the ability to serve an arbitrarily large load, something that can be achieved if the service can scale throughput when provided with additional resources (e.g., processors). In light of the requirements of modern online services, combining high availability and high performance without sacrificing consistency is a challenging endeavor of utmost importance. This project puts forward a research agenda for a collaborative effort initially between the University of Lugano (USI) in Switzerland, the Pontifícia Universidade Católica do Rio Grande do Sul (PUCRS), and the Universidade Federal de Santa Catarina (UFSC), to advance the state of the art on scalable and dependable (i.e., highly available) distributed systems. During the project, UFPR, UNB and UFU joined efforts.

Replication has been extensively studied by the research community. Early work in the database community dates back to the late 1970s and early 1980s, addressing both theoretical and practical concerns (e.g., one-copy serializability, primary copy replication). In the distributed systems community, foundational work on object replication (i.e., non-transactional behavior) dates back to the late 1970s (e.g., state machine replication). Although the topic is well-established, general-purpose replication techniques are mostly used for high availability, not performance - in fact, a replicated service is usually outperformed by its single-server implementation. Replication for performance typically sacrifices consistency: the behavior of the replicated service does not correspond to the intuitive behavior of a single-server service. Although some services can settle for weak consistency, it renders the design of services more complex, if possible at all.

Project Description - cont.

Scalability, the ability to increase performance by upgrading existing resources (scale up) or aggregating additional resources to the existing ones (scale out) is fundamental to meeting the performance requirements of modern services. Unfortunately, existing replication techniques cannot benefit from either approach. Modern servers increase processing power by aggregating multiple processors (e.g., multicore architectures), therefore scaling up a replicated service requires a replication technique that supports parallelism (multithreading) but existing techniques have at least one sequential part in their execution. Some replication techniques provide performance improvements by aggregating additional replicas (scale out), but the improvements are limited. The reason is that each new replica contains the complete ser- vice state and must execute every request, thus adding replicas will not result in improved performance, although it will lead to higher availability.

This project proposal is related to our research agenda towards techniques for scalable general-purpose replication. It is motivated by the observation that guarding service designers against the complexity of replication and scalable performance allows designers to focus on the service's inherent complexity. We aim at state-machine replication, a fundamental approach to replication, used in many current systems (e.g., Google's Chubby, Spanner, Scatter). Scaling state-machine replication is a formidable challenge since the technique relies on the assumption that execution at each replica is deterministic, which usually requires sequential execution. The project promises to deliver two main contributions: (a) It will advance the state-of-the-art in general-purpose replication protocols and hence match the requirements of modern online services with- out overburdening service designers with additional complexity. Moreover, we expect that overcoming the performance limitations of state-machine replication will likely lead to developments that will be useful in other distributed settings that require scalable performance. (b) The project will strengthen the collaborations between researchers from the involved institutions and pave the way for a durable collaboration between the main applicant (USI) and researchers from PUCRS, UFSC, UFPR, UNB and UFU.

Program

PhD Thesis viva - may 16, 14:30 - Room 516 - Fac. de Informática

- Title: Fast Recovery in Parallel State Machine Replication
- Author: Odorico Machado Mendizabal - Advisors: Fernando L. Dotti (PUCRS) e Fernando Pedone (USI)
- Abstract: A well-established technique used to design fault-tolerant systems is state machine replication. In part, this is explained by the simplicity of the approach and its strong consistency guarantees. The traditional state machine replication model builds on the sequential execution of requests to ensure consistency among the replicas. Sequentiality of execution, however, threatens the scalability of replicas. Recently, some proposals have suggested parallelizing the execution of replicas to achieve higher performance. Despite the success of parallel state machine replication in accomplishing high performance, the implication of such models on the recovery is mostly left unaddressed. Even for the traditional state machine replication approach, relatively few studies have considered the issues involved in recovering faulty replicas. The motivation of this thesis is clarifying the challenges and performance implications involved in checkpointing and recovery for parallel state machine replication. The thesis also aims to advance the state-of-the-art by proposing novel algorithms for checkpointing and recovery in the context of parallel state machine replication. Performing checkpoints efficiently in such parallel models is more challenging than in classic state machine replication because the checkpoint operation must account for the execution of concurrent commands. In this thesis, we review checkpointing techniques for parallel approaches to state machine replication and compare their impact on performance through simulation. Furthermore, we propose two checkpoint techniques for one of these parallel models. Recovering a replica requires (a) retrieving and installing an up-to-date replica checkpoint, and (b) restoring and re-executing the log of commands not reflected in the checkpoint. Parallel state machine replication render recovery particularly challenging since throughput under normal execution (i.e., in the absence of failures) is very high. Consequently, the log of commands that need to be applied until the replica is available is typically large, which delays the recovery. We present two novel techniques to optimize recovery in parallel state machine replication. The first technique allows new commands to execute concurrently with the execution of logged commands, before replicas are completely updated. The second technique introduces ondemand state recovery, which allows segments of a checkpoint to be recovered concurrently. We experimentally assess the performance of our recovery techniques using a full-fledged parallel state machine replication prototype and compare the performance of these techniques to traditional recovery mechanisms under different scenarios.

Program

Workshop Scalable Dependability - May 17, Room 516, Agenda

9:00 - 9:05	Opening
9:05 - 9:40	Dynamic Scalable State Machine Replication. Long Hoang Le (USI), Carlos Eduardo Bezerra (USI), and Fernando Pedone (USI)
9:40 - 10:10	Supporting Parallel Execution on BFT-SMaRt. (Suportando Execuções Paralelas no BFT-SMaRt). Eduardo Alchieri (UNB)
10:10 - 10:40	A Consensus-based Fault Tolerant Event Logger for High-Performance Computing. (Um Event Logger Tolerante a Falhas Baseado em Consenso para Computação de Alto Desempenho.) Edson Tavares de Camargo (UFPR), Elias P. Duarte Jr. (UFPR), Fernando Pedone (USI)
10:40 - 11:10	A failure detection service with auto adjustable QoS for multiple simultaneous applications. (Um Serviço de Detecção de Falhas com QoS Auto-Ajustável para Múltiplas Aplicações Simultâneas.) Rogério C. Turchetti (UFSM/UFPR), Elias P. Duarte Jr. (UFPR), Luciana Arantes (LIP6) e Pierre Sens (LIP6)
11:10 - 11:40	Discussion
11:40 - 14:00	Lunch
14:00 - 14:30	Leaderless Algorithms for Distributed Agreement. (Algoritmos Leaderless de Acordo Distribuído.) Lásaro Camargos (UFU)
14:30 - 15:00	(Implementing and evaluating the Collision-Fast Atomic Broadcast Protocol.) Implementação e avaliação do protocolo Collision-fast Atomic Broadcast. Rodrigo Queiroz Saramago - rod@comp.ufu.br, Lásaro Jonas Camargos - lasaro@ufu.br
15:00 - 16:00	Discussion - Correctness of Distributed Algorithms
16:00 - 17:00	Discussion of next steps in the project

Program

Workshop Scalable Dependability
Visiting Prof. Fernando Pedone

- Title: Dynamic Scalable State Machine Replication
- Authors: Long Hoang Le (USI), Carlos Eduardo Bezerra (USI), and Fernando Pedone (USI)
- Abstract: State machine replication (SMR) is a well-known technique that guarantees strong consistency (i.e., linearizability) to online services. In SMR, client commands are executed in the same order on all server replicas, and after executing each command, every replica reaches the same state. However, SMR lacks scalability: every replica executes all commands, so adding servers does not increase the maximum throughput. Scalable SMR (S-SMR) addresses this problem by partitioning the service state, allowing commands to execute only in some replicas, providing scalability while still ensuring linearizability. One problem is that S-SMR quickly saturates when executing multi-partition commands, as partitions must communicate. Dynamic S-SMR (DS-SMR) solves this issue by repartitioning the state dynamically, based on the workload. Variables that are usually accessed together are moved to the same partition, which significantly improves scalability. We evaluate the performance of DS-SMR with a scalable social network application.

Program

Workshop Scalable Dependability

- Título : Suportando Execuções Paralelas no BFT-SMaRt
- Autores : Eduardo Alchieri (UNB)
- Resumo : A replicação Máquina de Estados é uma das abordagens mais usadas na implementação de sistemas tolerantes a falhas, tanto por parada quanto bizantinas. Esta abordagem consiste em replicar os servidores e coordenar as interações entre os clientes e as réplicas dos servidores, com o intuito de que as várias réplicas apresentem a mesma evolução em seus estados. Para isso, as requisições dos clientes devem ser ordenadas e executadas seguindo esta ordem em todas as réplicas. Este requisito fez com que a maioria dos trabalhos utilizassem uma única thread de execução em cada réplica. Com o objetivo de melhorar o desempenho do sistema, novas abordagens foram introduzidas para suportar várias threads de execução por réplica. Dando seguimento a estes trabalhos, este artigo descreve como um protocolo que possibilita o emprego de várias threads de execução nas réplicas foi adaptado e implementado no Bft-SMaRt, além de analisar uma série de experimentos realizados.

Program

Workshop Scalable Dependability

- Título: Um Event Logger Tolerante a Falhas Baseado em Consenso para Computação de Alto Desempenho
- Autores: Edson Tavares de Camargo (UFPR), Elias P. Duarte Jr. (UFPR), Fernando Pedone (USI)
- Resumo: Sistemas de computação de alto desempenho tradicionalmente empregam técnicas baseadas em rollback-recovery para permitir que aplicações paralelas executem corretamente apesar das falhas. O registro de mensagens (message-logging) é uma atrativa estratégia baseada em rollback-recovery que evita as desvantagens do checkpoint coordenado em sistemas com baixo tempo médio entre as falhas (Mean Time Between Failures - MTBF). A maioria dos protocolos de registro de mensagens se apoia em uma entidade centralizada, chamada de Event Logger, para armazenar informações (isto é determinantes) que permitam a recuperação de um processo da aplicação. Essa abordagem centralizada, além representar um evidente ponto único de falha, também prejudica a eficiência dos protocolos de registro de mensagens. Neste trabalho apresentamos um Event Logger distribuído e tolerante a falhas baseado em consenso que apresenta desempenho superior à abordagem centralizada. Um Event Logger que armazena determinantes de aplicações MPI através do algoritmo Paxos foi implementado. O Event Logger proposto herda as propriedades do Paxos: a segurança é garantida mesmo se o sistema é assíncrono e o progresso é garantido apesar de falhas de processos. Resultados experimentais apresentam o desempenho do Event Logger distribuído aplicado à aplicação AMG (Algebric MultiGrid) e ao NAS Parallel Benchmark usando tanto uma configuração clássica quanto paralela do Paxos.

Program

Workshop Scalable Dependability

- Título: Um Serviço de Detecção de Falhas com QoS Auto-Ajustável para Múltiplas Aplicações Simultâneas
- Autores: Rogério C. Turchetti (UFSM/UFPR), Elias P. Duarte Jr. (UFPR), Luciana Arantes (LIP6) e Pierre Sens (LIP6)
- Resumo: Detectores de falhas monitoram os estados de processos de uma aplicação distribuída, efetuando hipóteses temporais sobre atrasos no sistema e disponibilizando informações sobre os estados destes processos. Neste trabalho é proposto o serviço de detecção de falhas denominado QoS-CFDS (Quality of Service-Configurable Failure Detection Service). O serviço QoS-CFDS é auto-ajustável, isto é, adapta seus parâmetros de monitoramento conforme as variações percebidas no ambiente e de acordo com as necessidades de cada aplicação. São propostas estratégias para ajustar a QoS do detector de acordo com os requisitos fornecidos por múltiplas aplicações simultâneas. Um protótipo do serviço proposto foi implementado com o protocolo SNMP e resultados experimentais são descritos para o desempenho do detector, incluindo os benefícios dos ajustes realizados para atender requisitos das aplicações simultaneamente.

Program

Workshop Scalable Dependability

- Título: Algoritmos Leaderless de Acordo Distribuído
- Autores: Lásaro Camargos (UFU)
- Resumo: Em algoritmos de consenso/difusão atômica como Paxos, Raft e muitos outros, o coordenador, eleito por um oráculo ω, é responsável por alguns passos do algoritmo, podendo se tornar um gargalo. Alguns recentes protocolos minimizam, distribuem, ou mesmo eliminam o papel do coordenador. Nesta apresentação darei uma visão geral destes algoritmos *leaderless* de difusão atômica, apontando alguns possíveis passos futuros nesta linha.

Program

Workshop Scalable Dependability

- Título: Implementação e avaliação do protocolo Collision-fast Atomic Broadcast
- Autores: Rodrigo Queiroz Saramago - rod@comp.ufu.br, Lásaro Jonas Camargos - lasaro@ufu.br
- Resumo: Apesar de muito comum, descrições de algoritmos distribuídos em pseudo-código são propensas a erros, uma vez que não se pode testá-las. Existem especificações formais para construção de sistemas distribuídos, como TLA+, porem utilizam de linguagens que não se assemelham as linguagens de programação comumente utilizadas na indústria, sendo raramente utilizadas na prática. Além disso, tais especificações tendem a omitir detalhes de implementação, que podem influenciar diretamente no tempo de desenvolvimento e desempenho do sistema, como por exemplo, valores de "timeouts", estruturas de dados mais eficientes a serem utilizadas e mecanismos de recuperação de falhas, dentre outros. Não obstante, implementar tais algoritmos é notoriamente difícil, e muitos esforços são empregados na tentativa de transcrever especificações formais em um código confiável e com bom desempenho. Este trabalho descreve um desses esforços, na tentativa de implementar, sendo o mais fiel possível à especificação, o algoritmo de replicação de máquinas de estados Collision-fast Atomic Broadcast.