09191 Abstracts Collection
Fault Tolerance in High-Performance Computing and Grids
— Dagstuhl Seminar —

Franck Cappello[1], Laxmikant Kale[2], Frank Mueller[3], Keshav Pingali[4] and Alexander Reinefeld[5]

[1] INRIA Futurs - Orsay, F
fci@lri.fr
[2] Univ. of Illinois - Urbana, USA
kale@cs.uiuc.edu
[3] North Carolina State University, USA
mueller@cs.ncsu.edu
[4] Univ. of Texas at Austin, USA
[5] Zuse Institute Berlin
reinefeld@zib.de

Abstract. From June 4–8, 2009, the Dagstuhl Seminar 09191 "Fault Tolerance in High-Performance Computing and Grids" was held in Schloss Dagstuhl – Leibniz Center for Informatics. During the seminar, several participants presented their current research, and ongoing work and open problems were discussed. Abstracts of the presentations given during the seminar as well as abstracts of seminar results and ideas are put together in this paper. The first section describes the seminar topics and goals in general. Links to extended abstracts or full papers are provided, if available. Slides of the talks and abstracts are available online at http://www.dagstuhl.de/Materials/index.en.phtml?09191.

Keywords. High-Performance Computing, Grids, Fault-Tolerance, Applications, Runtime Systems, Operating Systems, Middleware, Peer-to-peer, Overlay Networks

09191 Seminar Report – Fault Tolerance in High-Performance Computing and Grids

The objective of this seminar was to bring together researchers and practitioners from the HPC and Grid communities to discuss medium to long-term approaches to address fault tolerance (FT). The focus of solutions was on the practical, system side and with the intent to reach beyond established solutions.

Overall, the objective of the workshop is to spark research activities in a coordinated manner that can significantly enhance FT capabilities of today's

and tomorrow's HPC systems and Grids. The benefits of this work extend to the community of scientific computing at large, well beyond computer science. Due to the wide range of participants (researchers and industry practitioners from the U.S., Europe, and Asia), forthcoming research work may significantly help enhance FT properties of large-scale systems, and technology transfer is likely to eventually reach general-purpose computing given the increasing trend to multi-core parallelism and server-style computing, such as Google. Specifically, the work should set the seeds for increased collaborations between institutes in Europe and the U.S./Asia. If successful, a follow-up seminar may be organized in the following year.

This meeting was the first of its kind at Dagstuhl and provided a foundation to create a community platform with a cohesive outlook on FT in HPC and Grids. The presentations of participants concentrated on fundamental issues related to FT in HPC applications, runtime systems, operating systems, networking, I/O and scheduler. The program consisted of an introductory session for all participants, 22 presentations well as four "open mic" sessions where time was set aside for spontaneous discussions, brain storming and community-building plans. The seminar brought together a total of 31 researchers and developers working in the areas related to fault tolerance from universities, national research laboratories and computer vendors. The goals were to increase the exchange of ideas, knowledge transfer, foster a multidisciplinary approach to attacking this very important research problem with direct impact on the way in which we design and utilize parallel systems to make applications resilient to faults in hardware or software. The presentations were grouped thematically as follows:

- HPC FT
  - "Transparent Fault Tolerance in Message Passing Systems", Thomas Herault , Université Paris Sud
  - "Sustained Exascale", George Bosilca , University of Tennessee
- Distributed System FT
  - "Handling MapReduce Failures in Dynamic Distributed Environments", Paolo Trunfio , University of Calabria
  - "Implementing Fault Tolerant Services on Structured Overlay Networks - Part 1 of the Messy Details", Alexander Reinefeld , K. Zuse Zentrum Berlin
  - "Implementing Fault Tolerant Services on Structured Overlay Networks - Part 2 of the Messy Details", Florian Schintke , K. Zuse Zentrum Berlin
- Application- vs. Process- vs. OS-level FT
  - "Berkeley Lab Checkpoint/Restart (BLCR): Status and Future Plans", Paul Hargrove , Lawrence Berkeley National Laboratory
  - "Challenges in Fault Tolerance for High-Performance Computing", Frank Mueller , North Carolina State University
  - "Object-based Over-Decomposition can enable powerful Fault Tolerance Schemes", Laxmikant Kale , University of Illinois - Urbana
- Monitoring and Network FT

  - "Supporting Fault-Tolerance in Modern High-End Computing Systems with InfiniBand", Dhabaleswar K. Panda , Ohio State University
  - "A Proactive Resiliency Approach for Large-Scale HPC Systems", Geoffroy Vallee , Oak Ridge National Lab.
- I/O FT
  - "Checkpoint I/O requirement for peta-scale applications", Xiaosong Ma , North Carolina State University
  - "Scalable Massively Parallel I/O to Task-Local Files", Wolfgang Frings , Jülich Supercomputing Centre
  - "Revisiting Fault Tolerance for HPC Applications", Franck Cappello, INRIA Futurs / Univ. of Illinois - Urbana, USA
- Grid FT
  - "Who cares about the future (jobs)?", Jörg Schneider , TU Berlin
  - "Bringing together fault tolerance and data consistency to enable grid data sharing.", Gabriel Antoniu , IRISA - Rennes
  - "Fault-tolerant Data Sharing in Grid Environments", Michael Schöttner , Universität Düsseldorf
- Soft Errors
  - "Modular Redundancy for Soft-Error Resilience in Large-Scale HPC Systems", Christian Engelmann , Oak Ridge National Lab.
  - "The plan for the new Frankfurt Green-IT HPC Computer and its requirements on fault tolerance infrastructure", Volker Lindenstruth , Universität Heidelberg
  - "Ensuring Collective Availability in Large Resource Pools via Classification", Artur Andrzejak , K. Zuse Zentrum Berlin
- FT Standardization
  - "Towards Support for Fault Tolerance in the MPI Standard", Richard L. Graham , Oak Ridge National Lab.
  - "Towards HPC Resilience standardization", Chokchai Leangsuksun , Louisiana Tech University
  - "Fault-tolerance - RAS - Resiliency: "you guys start writing code and I'll go see what they want..."", Stephen L. Scott , Oak Ridge National Lab.

The following open mic discussions complemented the program:

- "Learning from FT results across domains: Grids, HPC and beyond", Alexander Reinefeld
- "What is the right FT approach?", Laximant Kale
- "International collaborations [not only on system-level FT] for PetaScale and ExaScale systems", Franck Capello
- "FT Standardization", Frank Mueller

Through the lively engagement of all participants, the seminar was very successful and conducted in a professional, friendly and collegial atmosphere supported by the kind and helpful staff at Schloss Dagstuhl. Lively discussions continued every day well beyond meeting times. The group meeting in its four-day

format combined with the cosy confinement of Dagstuhl provided an umbrella for thoughtful discussion that conferences or workshops cannot provide. This helped create a community feeling that could become a building block for a concerted effort to coordinate future research activities, cooperate in outreach effort and maximize everyone's productivity and impact in fulling together each one's unique expertise for a combined effort to successfully solve the grand challenges of FT in HPC and Grids. During the meeting, follow-up action items with community-building character were identified, as detailed in the online discussion notes of the open mic sessions. They include (a) creation of a mailing list to coordinate activities and disseminate information on FT in high-performance computing and related areas, (b) provision of a Wiki designated to the collection of information on active projects, existing solutions and the coordination of future research activities, and (c) organization of follow-on meetings for the community. Within one month of the seminar, action items (a) and (b) have been realized and a follow-up meeting (c) is in the planning stages.

The seminar also included a half-day excursion with a guided tour of "Burg Eltz" and a wine-tasting and dinner at the lovely down of Beilstein on the Mosel river, which was well received by the participants. This document provides an account of Dagstuhl seminar 9191: Fault Tolerance in High-Performance Computing and Grids.

Keywords: Dagstuhl Seminar Proceedings, DROPS

Joint work of: Mueller, Frank

Open Mic: International collaboration

- Discuss the need for International collaboration or coordination
- Initiate a collaborative effort to collect and publish information about the community activities and needs (all the community is not here)
- For Peta Exascale Machines but also Grid

Joint work of: Franck Cappello

Ensuring Collective Availability in Large Resource Pools via Classification

Artur Andrzejak (K. Zuse Zentrum Berlin, DE)

Resource pools such as BOINC voluntarily computing project or Google data centers reach sizes of several hundred thousands of PCs. While availability of individual hosts cannot be guaranteed in such scenarios (and is notoriously bad in case of BOINC), the inherent resource redundancy offers new ways for achieving availability.

In this talk we evaluate the model of *collective availability*, i.e. assurance that at least a certain fraction of a specified set of hosts will be available in the short-term future. To this aim we use a set of scalable techniques for estimating the predictability of host's uptime and efficient forecasting of this uptime. By segmenting the resource pool according to availability, predictability and the forecasted uptime we are able to identify resource subsets with different classes of such collective availability. In conjunction with replication and migration such subsets are (statistically) robust against failures of individual boxes. We evaluate our approach on the availability data collected from over 25000 institutional hosts participating in the SETI@home project, and contrast it with the evaluation on over 75000 private hosts from this environment.

Keywords: Availability, machine learning, statistical models, large resource pools

Joint work of: Andrzejak, Artur; Kondo, Derrick; Anderson, David P.

Full Paper:
http://www.zib.de/andrzejak/my-papers/AndrzejakKondoAnderson-DSOM2008.pdf

See also: Artur Andrzejak, Derrick Kondo, David P. Anderson: Ensuring Collective Availability in Volatile Resource Pools via Forecasting, 19th IFIP/IEEE Distributed Systems: Operations and Management (DSOM 2008) (part of Manweek 2008), Samos Island, Greece, September 22-26, 2008

Fault Tolerance and Data Consistency for Grid Data-Sharing Services: From JuxMem to BlobSeer

Gabriel Antoniu (INRIA - Rennes Bretagne Atlantique, FR)

This talk addresses the challenge of transparent data sharing within computing grids built as cluster federations. On such platforms, the availability of storage resources may change in a dynamic way, often due to hardware failures. We focus on the problem of handling the consistency of replicated data in the presence of failures. We describe a software architecture which decouples consistency management from fault tolerance management. We illustrate this architecture with a case study showing how to design a consistency protocol using fault-tolerant building blocks. A prototype implementation of this protocol has been realized within JuxMem, a software experimental platform for grid data sharing. In a second step, to address today's new challenges of large-scale data managements on grids and clouds, we have recently started to explore a new approach, targetting the management of huge data accessed at fine grain, under heavy consistency. We experiment this approach with the Blobseer platform. To address fault tolerance in this approach, we investigate the benefits of global behavior modelling.

Keywords: Transparent data access, data sharing, fault tolerance, data consistency, grid data management, global behavior modelling

Sustained Exascale

George Bosilca (University of Tennessee, US)

Even making generous assumptions about the reliability of a single processor, it is clear that as the processor count in high end clusters grows into the tens of thousands, the mean time to failure (MTTF) will drop from hundreds of days to a few hours, or less. The type of 100,000- processor machines that now exist are expected to experience a processor failure almost daily, perhaps hourly. Although today's architectures are robust enough to incur process failures without suffering complete system failure, at this scale and failure rate, the only technique available to application developers for providing fault tolerance, within the current parallel programming model of checkpoint/restart, has performance and conceptual limitations that make it inadequate to the future needs of the communities that will use these systems.

The MPI standard, as defined by the MPI forum in 1994-1997, has undoubtedly become the de-facto standard for parallel applications. While few other parallel programming paradigms are currently available, their share of the parallel applications world is limited to some narrow application domains and the overall number of applications using these paradigms is limited. Unfortunately, neither of the first or second version of the MPI standard were addressing the fault tolerance issue. We can find two explanations for this statement: at this time, HPC architecture were using fewer CPUs, thus failures were very uncommon events and were not a practical issue; and fault tolerant mechanisms, especially automatic ones, are usually introducing some overhead on message passing performance. Though MPI standard does not provide any help to recover from failures, it defines several error codes returned by MPI functions to warn the application that may be used to take some corrective action. Several projects have aimed at easing the development of fault tolerant MPI applications.

Although work in fault tolerance has been dominated in recent years by systems-oriented approaches that are transparent to the application and relatively easy to use, we believe that, in the next generation of high-end computing environments, successful approaches to fault tolerance must leverage intimate knowledge of the parallel environment, the application and its dominant algorithm in order to achieve the efficiencies required for faster (or even adequate) recovery times and memory requirements. This talk will present the latest advances in uncoordinated checkpoint/restart as well as user level recovery algorithms based on the research and prototypes from the University of Tennessee.

Keywords: Fault tolerance, MPI, ABFT

Revisiting Fault Tolerance for HPC Applications (parallel executions)

Franck Cappello (INRIA - Orsay, FR)

Currently, Fault Tolerance for HPC applications considers equivalence between parallel executions and distributed algorithm executions. As a consequence, classic approaches for fault tolerance in distributed systems have been considered such as replication and complex protocols computing consistent cuts of distributed algorithm executions (coordinated checkpointing and message logging). Typically, these approaches are used from the beginning to the end of the execution, considering that a parallel execution always behaves as a distributed algorithm. However HPC applications are by far more complex than a single distributed algorithm and rely on a collection of global transformations that do not necessarily need complex fault tolerance algorithms. Based on this view of parallel executions, we derive a simple fault tolerance approach based on essential global transformations that considers transactions as a mechanism for fault tolerance in HPC parallel applications. This approach being still under its design stage, we only discuss its main principles.

Modular Redundancy for Soft-Error Resilience in Large-Scale HPC Systems

Christian Engelmann (Oak Ridge National Lab., US)

Recent investigations into resilience of large-scale high-performance computing (HPC) systems showed a continuous trend of decreasing reliability and availability. Newly installed systems have a lower mean-time to failure (MTTF) and a higher mean-time to recover (MTTR) than their predecessors. Modular redundancy is being used in many mission critical systems today to provide for resilience, such as for aerospace and command & control systems. The primary argument against modular redundancy for resilience in HPC has always been that the capability of a HPC system, and respective return on investment, would be significantly reduced. We argue that modular redundancy can significantly increase compute node availability as it removes the impact of scale from single compute node MTTR. We further argue that single compute nodes can be much less reliable, and therefore less expensive, and still be highly available, if their MTTR/MTTF ratio is maintained.

Joint work of: Engelmann, Christian; Ong, Hong; Scott, Stephen

Scalable Massively Parallel I/O to Task-Local Files

Wolfgang Frings (Jülich Supercomputing Centre, DE)

Parallel applications often store data in multiple task-local files, for example, to remember checkpoints, to circumvent memory limitations, or to record performance data. When operating at very large processor configurations, such applications often experience scalability limitations when the simultaneous creation of thousands of files causes metadata contention or simply when large file counts complicate file management or operations on those files even destabilize the file system. SIONlib is a parallel I/O-library that addresses this problem by transparently mapping a large number of task-local files onto a small number of physical files via internal metadata handling and block alignment to ensure high performance.

Using SIONlib significantly reduces file creation overhead and simplifies file handling, while requiring only minimal source code changes and without penalizing read and write performance. We evaluate SIONlib's efficiency with up to 64K tasks and report significant performance improvements in two use cases.

Towards Support for Fault Tolerance in the MPI Standard

Richard L. Graham (Oak Ridge National Lab., US)

With the rapid increase in the size of current and planned High Performance Computing platforms and their associated large-scale simulation codes, and the emergence of long running loosely coupled mission critical parallel applications, the need to handle system failures while these computer codes continue to run correctly is essential. With the Message Passing Interface (MPI) being the most widely used parallel application process management and communications system, addressing MPI's support for Fault Tolerance is necessary. This talk will discuss ongoing work to add support for Fault Tolerance to the MPI standard. This work is aimed at the MPI-3 standard.

Keywords: MPI, Fault Toleracne

Berkeley Lab Checkpoint/Restart (BLCR): Status and Future Plans

Paul Hargrove (Lawrence Berkeley National Laboratory, US)

This will be a very informal presentation about the current status and future plans for the software package Berkeley Lab Checkpoint/Restart (BLCR).

As a kernel-level implementation of classic rollback recovery, BLCR is inherently very I/O intensive. In this talk I will discuss with the

Keywords: BLCR checkpoint scalability CPR rollback

Transparent Fault Tolerance in Message Passing Systems

Thomas Herault (Université Paris Sud, FR)

In this talk, I will present the study that was conducted in the Grand-Large project/team of the INRIA at Orsay, and the Parallelism team of the University Paris-Sud on transparent fault tolerant protocols for message passing systems (in particular MPICH). The talk will present all the protocols that were studied during the last years, the results we obtained, and ongoing research done today in collaboration with the university of Tennessee. I will also present recent works done to port MPI on institutional grids, and issues that were raised by such study.

Object-based Over-Decomposition can enable powerful Fault Tolerance Schemes

Laxmikant Kale (University of Illinois - Urbana, US)

HPC systems for Computational Science and Engineering have almost reached the threshold where they cannot do without some for of fault tolerance. Although system-level checkpoint-restart keeps things simple for the application developer, they lead to high overhead; application-level schemes are effort intensive. Schemes based on smart runtime systems appear to be at the right level for effecting fault tolerance. Further, as fault frequencies rise and MTBF approaches ideal checkpointing period, it becomes harder for applications to make forward progress. I will discuss a series of techniques that can help in this situation. In particular, my pet technique of object based over-decomposition (as implemented in Charm++ and AMPI) appears to be (surprise, surprise) key to many novel fault tolerance ideas — I'll explain why. The techniques include proactive fault tolerance, distributed checkpoints, and message-logging with parallel recovery.

I'd like to involve the expertise of the community in combining new ideas with object-based techniques to overcome challenges of future, which include minimizing forward-path (i.e. no-failure) overhead of scalable fault tolerance schemes.

Keywords: Object-based overdecompostion, proactive fault tolerance, message logging, parallel recovery

Towards HPC Resilience standardization

Chokchai Leangsuksun (Louisiana Tech University, US)

A new paradigm in distributed computing is emerging, with systems nearing the million core mark, the probability of any single component failing is no longer merely a possibility but a guarantee. To address the issue of imminent failure, resilient computing has emerged as the champion that will not only react to expected failures, but also cope with failures that are unexpected. With the mission to ensure that a job makes progress in spite of failures, resilience is difficult to define; it attempts to capture both proactive and reactive fault tolerance, while also attempting to quantify whether an application is making progress toward its intended goal. A key aspect of resilient computing is that it is application centric, the sole focus being to ensure the application makes progress.

There are many challenges associated with resilience, and many organizations and researchers are attacking various aspects. With resilience's change in perspective from system level to job/application level, the semantics related to reliability need to be addressed. For instance, in HPC environments, what does availability mean? Does availability require all resources to be available? Or only those required by the application? In addition, knowing more detailed information about the state of the system is required, as resilience mechanisms should respond differently based on the state of the system. Some potential states for the system globally as well as resources within it are: production mode, planned downtime, and engineering mode. It is also important to clarify what exactly constitutes a failure; from the application point of view, a failure could be called an interrupt or it may be called a failure by the current methodology. These subtle differences should be clarified now, rather than later to ensure that the HPC community has a clear vision and language for expressing resilient computing research. The benefits of addressing these challenges now expand beyond resilient computing, and have the potential to enhance and clarify the research of all HPC researchers.

To address these challenges, we are actively working to formalize and justify existing terminology and metrics as well as thier semantics. We seek to document the nuances when moving between perspectives, and work towards a whitepaper that clearly states what the research community agrees are, the accepted terms and what the intended meaning should be based on the context. We plan to have a draft available in a public repository, and will encourage all researchers to propose modifications.

Keywords: Resilience HPC standardization Reliability

The plan for the new Frankfurt Green-IT HPC Computer and its requirements on fault tolerance infrastructure

Volker Lindenstruth (Universität Heidelberg, DE)

A new Green-IT computer center is planned in Frankfurt with a cooling ouverhead of less than 20%. This is realized by water cooled racks, which are mounted in a 3D steel structure like a high rise rack store. The computer itself is planned as a cluster, connected with a high performance low latency network and utilizes hardware accelerators, such as GPUs and the like. The presentation will give an outline of the datacenter architecture and its fault tolerance requirements. It turns out that a number of design parameters are typically chosen such, that they provide a certain level of system security at the cost of energy usage during normal operation. The Frankfurt system is planned differently, but at the cost of additional requirements for system support. A second subject is the growing demand for hardware accelerators. Many of the existing applications of the discussed system can utilize such devices. However, they may be either somewhat unreliable because they do not implement ECC memories or they are quite expensive. We want to investigate whether it is possible to cope with this issue by fault tolerant algorithms. The scale of a distributed system with thousands of nodes, which are themselvse a structured system will be unreliable in itself. On average we expect one to two node failures per day. Fault tolerance infrastructure has to be available in order to handle such failing systems, for instance by failing over the application to another node.

Keywords: Many core, multi core, HPC, fault tolerance, green-IT

Checkpoint I/O requirement for peta-scale applications

Xiaosong Ma (North Carolina State University, US)

Large-scale applications routinely write checkpoints. Many have treated checkpoint data the same way as the result data. However, these two types of output have very different semantics and requirements. In this talk, we discuss the opportunities in differentiating checkpoint data from result data, for more efficient parallel I/O and overall execution.

Keywords: Checkpointing, parallel I/O

Challenges in Fault Tolerance for High-Performance Computing

Frank Mueller (North Carolina State University, US)

Large-scale parallel computing is relying increasingly on clusters with (tens of) thousands of processors.

At such large counts of compute nodes, faults are becoming common place. Current techniques to tolerate faults focus on reactive schemes for recovery and generally rely on a simple checkpoint/restart mechanism. Yet, they have a number of shortcomings. (1) They do not scale and require complete job restarts. (2) Projections indicate that the mean-time-between-failures is approaching the overhead required for checkpointing. (3) Existing approaches are application-centric, which increases the burden on application programmers and reduces portability.

To address these problems, we discuss a number of techniques and the level of maturity (or lack thereof): (a) Scalable network overlays track node failures and recoveries lifting part of the burden from the programmers. (b) Mechanisms for on-the-fly recovery without a need to restart compute jobs conserve large-scale resources much in contrast to today's techniques. (c) An approach for proactive fault tolerance that complements reactive schemes further reduces resource requirements. (d) Minimal API support for fault tolerance increases portability without requiring vendors to implement extensive functionality.

Keywords: Fault tolerance, high performance computing

Supporting Fault-Tolerance in Modern High-End Computing Systems with InfiniBand

Dhabaleswar K. Panda (Ohio State University, US)

InfiniBand architecture is emerging as a modern interconnect for designing large-scale High-End Computing (HEC) systems. This architecture provides novel mechanisms for performance, network fault-tolerance and Quality of Service (QoS). Examples of some of these mechanisms include: Remote DMA (RDMA), Automatic Path Migration (APM), Service Levels (SL) and Virtual Lanes (VL). We will present novel schemes using these mechanisms to provide various levels of fault tolerance at the MPI layer on InfiniBand clusters. Our experiences in designing multiple levels of fault-tolerance (such as network-level fault-tolerance using APM, process-level fault-tolerance using Checkpoint-Restart, I/O aggregation to reduce checkpoint-restart overhead, efficient process migration using RDMA and virtualization, etc.) in MVAPICH2 library will be presented. Challenges in using these schemes on large-scale HEC systems will be discussed.

Keywords: Clusters, InfiniBand, Fault-Tolerance, Checkpoint-Restart, MPI and QoS

Joint work of: Panda, Dhabaleswar K. (DK)

Implementing Fault Tolerant Services on Structured Overlay Networks - Part 1 of the Messy Details

Alexander Reinefeld (K. Zuse Zentrum Berlin, DE)

Structured overlay networks provide scalability and availability with unreliable components. They are designed for operation in hostile environments where nodes may crash, leave or join at any time.

In the first part of our talk we investigate all the nitty gritty details that are needed for implementing a stable data structure in the face of unreliable components. We discuss how consistent data access can be guaranteed with replicated data stores on Internet-connected servers.

In the second part of the talk we discuss the advantages of using functional programming languages like Erlang in the design and implementation of fault tolerant, large-scale distributed applications.

Keywords: Structured overlays, Paxos, transactions, Erlang

Implementing Fault Tolerant Services on Structured Overlay Networks - Part 2 of the Messy Details

Florian Schintke (K. Zuse Zentrum Berlin, DE)

Structured overlay networks provide scalability and availability with unreliable components. They are designed for operation in hostile environments where nodes may crash, leave or join at any time.

In the first part of our talk we investigate all the nitty gritty details that are needed for implementing a stable data structure in the face of unreliable components. We discuss how consistent data access can be guaranteed with replicated data stores on Internet-connected servers.

In the second part of the talk we discuss the advantages of using functional programming languages like Erlang in the design and implementation of fault tolerant, large-scale distributed applications.

Keywords: Structured overlays, Paxos, transactions, Erlang

Who cares about the future (jobs)?

Joerg Schneider (TU Berlin, DE)

A lot of failure handling techniques have been developed to save running jobs. However, a resource failure can last unpredictably long. Therefore, submitted but not yet started jobs are affected by the resource outage as well. In case of advance reservations with a guaranteed end time, a simple postponing after the downtime is not sufficient.

In this talk approaches to remap the possibly affected jobs to other resources are discussed together with the effects to the Grid and the non-failed resources.

Joint work of: Schneider, Jörg; Linnert, Barry

Fault-tolerant Data Sharing in Grid Environments

Michael Schoettner (Universität Düsseldorf, DE)

Data replication is one of the basic building blocks for implementing fault tolerance. Furthermore, replication also improves application performance by providing local data access. The synergy of fault tolerance and locality is driving the design and implementation of our grid data sharing service complementing traditional message passing by transparent data access, automatic replica and consistency management. For scalability and fault tolerance reasons our communication infrastructure is built on top of structured overlay network techniques.

For severe errors we rely on a grid checkpointing service (GCS) that is able to save and restore distributed and parallel applications in heterogeneous environments. The GCS service is integrating existing kernel checkpointer solutions. A common kernel checkpointer API allows the GCS to use different checkpointers in a uniform way. Our architecture is open to support different checkpointing strategies that can be adapted according to evolving failure situations and changing application requirements.

Keywords: Fault tolerance, data sharing, data consistency, grid checkpointing

Joint work of: Schöttner, Michael; Morin, Christine

Fault-tolerance - RAS - Resiliency: "you guys start writing code and I'll go see what they want..."'

Stephen L. Scott (Oak Ridge National Lab., US)

Some of us have been evangelizing the cause for fault-tolerance, RAS (Reliability, Availability, Serviceability), and Resiliency for a quite a number of years. Lately people have noticed! That is the good news - the bad news is that the high performance computing community does not have any semblance of an agreement regarding what they want and how to achieve it. Perhaps we need to begin with something as "simple" as defining FT/RAS/Resiliency? (Let's hope not!) How does the view differ for folks in the application, systems, and hardware communities? What can we do to achieve community consensus so that our resilience efforts can move forward, receive accepted by the others, and ultimately help to achieve our high performance computing goals?

Handling MapReduce Failures in Dynamic Distributed Environments

Paolo Trunfio (University of Calabria, IT)

MapReduce is a programming model used for processing large data sets in a highly parallel way. MapReduce implementations are based on a master-slave model. The failure of a slave is managed by re-executing its task on another slave, while master failures are not effectively managed by current MapReduce implementations as designers consider failures unlikely in reliable environments, like clusters and Cloud systems. On the contrary, node failures (including master failures) are likely to happen in dynamic environments, like Grids and P2P systems, where nodes join and leave the network at an unpredictable rate. Therefore, providing effective mechanisms to manage master failures is fundamental to exploit the MapReduce model in the implementation of data-intensive applications in those dynamic environments where current MapReduce implementations could be unreliable. The goal of our work is investigating how to improve the master-slave architecture of current MapReduce implementations to make it more suitable for Grid-like and P2P dynamic scenarios. The extended model we introduce here exploits a P2P model to dynamically assign the master role, managing intermittent nodes participation and master failures in a decentralized but simple way

Keywords: MapReduce, P2P

Joint work of: Talia, Domenico; Trunfio, Paolo

A Proactive Resiliency Approach for Large-Scale HPC Systems

Geoffroy Vallee (Oak Ridge National Lab., US)

Reactive fault tolerance has been the privileged solution for system resilience since many years. However, this approach nowadays show its limits, mainly because of I/O issues. In this presentation, I present techniques and policies for proactive resiliency which aims at predicting failure in order to move applications away from nodes that are about to fail.

I will present advantages and limitations of such a solution, focusing on the system aspects.

Keywords: Proactive resiliency, system