Enabling Application-Integrated Proactive Fault Tolerance

Dai YANG, Josef WEIDENDORFER, Carsten TRINITIS, Tilman KÜSTNER, Sibylle ZIEGLER

TU München, Lehrstuhl für Rechnertechnik und Rechnerorganization, Boltzmannstr. 3, 85748 Garching, Germany
LMU München, Klinik und Poliklinik für Nuklearmedizin, Marchioninistraße 15, 81377 München, Germany

Abstract. Exascale computing is the next major milestone for the HPC community. Due to a steadily increasing probability of failures, current applications must be made malleable to be able to cope with dynamic resource changes. In this paper, we show first results with LAIK, a lightweight library for dynamically re-distriutable application data. This allows to free compute nodes from workload before a predicted failure. For a real-world application, we show that LAIK adds negligible overhead. In addition, we show the effect of different re-distribution strategies.

Keywords. High Performance Computing, Parallel Programming Models, Application-Integrated Fault Tolerance, Data Distribution

1. Introduction

The performance increase of High Performance Computing (HPC) systems traditionally relies both on improvements of components as well as increased system size, with millions of compute units available nowadays. This results in increased importance of fault tolerance mechanisms, as the probability of single node failures becomes higher for applications running on larger systems. For future systems, Defense Advance Research Projects Agency (DARPA) expects that "traditional" fault tolerance technologies such as Checkpoint & Restart will require extensive amounts of resources, which is in clear contradiction to expected high efficiency [2]. An application-transparent approach for increased reliability is to provide fault tolerant environments using virtual machines [8] in combination with process-level live migration [10,16]. However, this often requires a significant amount of resources itself. Another option is to use fault tolerance techniques which cooperate with applications, e.g. by requiring them to dynamically adapt to resource change requests from the outside. This results in reduced overhead and better scalability.

In this paper we present a prototype of a library to help developers make their applications more dynamic, called LAIK (Lightweight Application-Integrated data distribution for parallel workers). It is based on our proposal presented in [17]. Given an “owner-computes” style, by taking over control of partitioning of data containers, LAIK can control the workload for compute units. This enables...