Poster: Damaris - Using Dedicated I/O Cores for Scalable Post-petascale HPC Simulations

Matthieu Dorier

SRC: Damaris - Using Dedicated I/O Cores for Scalable Post-petascale HPC Simulations

Matthieu Dorier
ENS Cachan, Brittany - IRISA
Rennes, France
matthieu.dorier@eleves.bretagne.ens-cachan.fr
Advisor: Gabriel Antoniu, INRIA Rennes, France, gabriel.antoniu@inria.fr

As we enter the post-petascale era, scientific applications running on large-scale platforms generate increasingly larger amounts of data for checkpointing or offline visualization, which puts current storage systems under heavy pressure. Unfortunately, I/O scalability rapidly fades behind the increasing computation power available, and thereby reduced the overall application performance scalability. We consider the common case of large-scale simulations who alternate between computation phases and I/O phases. Two main approaches have been used to handle these I/O phases: 1) each process writes an individual file, leading to a very large number of files from which it is hard to retrieve scientific insights; 2) processes synchronize and use collective I/O to write to the same shared file. In both cases, because of mandatory communications betweens processes during the computation phase, all processes enter the I/O phase at the same time, which leads to huge access contention and extreme performance variability.

Previous research efforts have focused on improving each layer of the I/O stack separately: at the highest level scientific data formats like HDF5 [4] allow to keep a high degree of semantics within files, while leveraging MPI-IO optimizations. Parallel file systems like GPFS [5] or PVFS [2] are also subject to optimization efforts, as they usually represent the main bottleneck of this I/O stack.

As a step forward, we introduce Damaris (Dedicated Adaptable Middleware for Application Resources Inline Steering), an approach targeting large-scale multicore SMP supercomputers. The main idea is to dedicate one or a few cores on each node to I/O and data processing to provide an efficient, scalable-by-design, in-compute-node data processing service. Damaris takes into account user-provided information related to the application, the file system and the intended use of the datasets to better schedule data transfers and processing. It may also respond to visualization tools to allow in-situ visualization without impacting the simulation.

We tested our implementation of Damaris as an I/O backend for the CM1 atmospheric model[1], one of the application intended to run on next generation supercomputer BlueWaters [1] at NCSA. CM1 is a typical MPI application, originally writing one file per process at each checkpoint using HDF5. Deployed on 1024 cores on BluePrint, the BlueWater's interim system at NCSA with GPFS as underlying filesystem, this approach induces up to 10 seconds overhead in checkpointing phases every 2 minutes, with a high variability in the time spent by each process to write its data (from 1 to 10 seconds). Using one dedicated I/O core in each 16-cores SMP node, we completely remove this overhead. Moreover, the time spared by the I/O core enables a better compression level, thus reducing both the number of files produced (by a factor of 16) and the total data size. Experiments conducted on the French Grid5000 [3] testbed with PVFS as underlying filesystem and a 24 cores/node cluster emphasized the benefit of our approach, which allows communication and computation to overlap, in a context involving high network contention at multiple levels.

[1]This work was done in the framework of a collaboration between the KerData INRIA - ENS Cachan/Brittany team (Rennes, France) and the NCSA (Urbana-Champaign, USA) within the Joint INRIA-UIUC Laboratory for Petascale Computing.

ICS'11, May 31–June 4, 2011, Tuscon, Arizona, USA.
ACM 978-1-4503-0102-2/11/05.

Categories and Subject Descriptors

D.1.3 [**Concurrent Programming**]: Parallel Programing; I.6.6 [**Simulations and Modeling**]: Simulation Output Analysis; E.5 [**Files**]: Optimization

General Terms

Design, Experimentation, Performance

Keywords

Exascale Computing, Multicore Architectures, I/O, Dedicated Cores

1. REFERENCES

[1] The Blue Waters Project. `http://www.ncsa.illinois.edu/BlueWaters/`.
[2] P. H. Carns, W. B. Ligon, III, R. B. Ross, and R. Thakur. PVFS: A parallel file system for Linux clusters. In *Proceedings of the 4th annual Linux Showcase & Conference - Volume 4*, pages 28–28, Berkeley, CA, USA, 2000. USENIX Association.
[3] Grid5000. https://www.grid5000.fr.
[4] NCSA. Hierarchical Data Format HDF5, `http://www.hdfgroup.org/HDF5/`.
[5] F. Schmuck and R. Haskin. GPFS: A Shared-Disk File System for Large Computing Clusters. In *In Proceedings of the 2002 Conference on File and Storage Technologies (FAST)*, pages 231–244, 2002.