Superclouds: Scalable High Performance Nonstop Infrastructure for AI and Smart Societies

Bill McColl(✉)

Huawei Research, Paris, France
bill.mccoll@huawei.com

Abstract. We describe a new approach to building high performance nonstop infrastructure for scalable AI and cloud computing. With AI as the technological driving force behind future smart cities and smart societies, we will need powerful new nonstop AI infrastructures at both the cloud level and at the edge.

Keywords: AI · Smart cities · HPC Big Data

1 Introduction

Future smart cities and smart societies will require powerful new nonstop AI infrastructures at both the cloud level and at the edge. Unfortunately, today's cloud computing architectures are not capable of delivering what is required, as they lack support for high performance general purpose parallel computing with fault tolerance and tail tolerance.

2 The Problem – Fault Tolerance and Tail Tolerance

In many scenarios in large-scale parallel computing, it is not possible to simply run the computation again, or to use simple methods such as checkpointing, to handle faults or long latencies (tails). For example, with petaflop and exaflop computations it will be very costly to "just run again or use checkpointing". Another example is where we are running infinite continuous nonstop computations. In such a case, "just run again or use checkpointing" is impossible, as it is in the case where we are running realtime computations.

Many modern large-scale parallel computing applications are both highly iterative and communication-intensive. For example, Big Data Analytics, Graph Computing, Machine Learning, Deep Learning, Artificial Intelligence, HPC, Modelling, Genomics, Network Optimization, Simulation.

Large-scale parallel computing in the cloud requires new software architectures that:

- Can be used to efficiently run any parallel computation, at any scale
- Can be used on cost-effective commodity architectures
- Can be used to efficiently run computations continuously, without interruptions
- Offers high performance

R. Mehmood et al. (Eds.): SCITA 2017, LNICST 224, pp. 3–5, 2018.
https://doi.org/10.1007/978-3-319-94180-6_1

- Offers high availability, with automatic fault tolerance and tail tolerance, self-healing and self-optimizing

Cloud computing offers parallelism, scale and cost-effectiveness, but clouds are very unpredictable. Maximum latency is often more than 100 times greater than average latency for identical tasks. In some cases the multiple can even be 1000 times or more. These long tails are a major problem, and quite different from the related problem of handling faults.

However, with modern software container technology, containers can be relaunched very quickly – in seconds rather than the minutes normally required to relaunch a virtual machine or physical server. So containers provide a means of restarting computations quickly, but there remains the challenge of deciding when to restart.

In this new era of large-scale cloud computing with frequent long tails, traditional checkpointing-based approaches to recovery are inadequate for a variety of simple reasons:

- The latency of writing to, and reading from, resilient storage is very high
- We need to choose frequent checkpointing or long recovery times - both are very bad, and there is no good tradeoff
- We need to choose frequent recovery or long tail limits - both are very bad
- With checkpointing, nonstop performance or continuous realtime performance are both impossible, due to stopping and restarting.

Are there alternatives to checkpointing for parallel computing with fault tolerance and tail tolerance?

3 General Purpose Parallel Computing - Computing in Rounds

Most large-scale parallel algorithms “compute in rounds”, irrespective of whether the actual software they are written in is MPI, BSP, MapReduce, Spark, Pregel, Giraph, Petuum, or other parallel programming models and systems. This style of parallel computing in rounds is normally referred to as Bulk Synchronous Parallel (BSP) computing. With BSP style parallel algorithms and software, the basic computational model is:

1. Compute on data in local memory
2. Globally communicate across the network
3. Synchronize
4. Repeat

Simple parallel models such as MapReduce provide an adequate framework for parallel computations that involve only a small number of rounds. Other models such as Spark provide an adequate framework for parallel computations of limited scale, where the low performance obtained by automatic management of communications is acceptable. For general purpose parallel computing at large scale, BSP style parallelism, either using MPI or BSP message passing software, has proven to be capable of delivering the highest levels of performance in all kinds of applications, including: Dense Linear

Algebra; Sparse Linear Algebra; Spectral Methods, e.g. FFT; N-Body Methods; Structured Grids; Unstructured Grids; Monte Carlo; Graph Computing; Dynamic Programming; Combinatorial Search, e.g. Branch-and-Bound; Machine Learning; Discrete Event Simulation.

So BSP, or "computing in rounds" is a proven method for general purpose parallel computing that can handle any type of large-scale parallel application. Moreover, many of the most important modern large-scale commercial parallel applications such as machine learning, deep learning, AI, network optimization, and graph analytics, are highly iterative, involving thousands of rounds, and can be very naturally and easily expressed as BSP computations.

4 Superclouds

In this talk I will show that BSP provides not only a foundation for the development of algorithms and software for HPC, Big Data Analytics, Graph Computing, and Machine Learning, but that is can also be adapted to provide a foundation for the design of next-generation nonstop "superclouds" for AI, that offer not only high performance, but also fault tolerance and tail tolerance.