Towards Simulation of an Unified Address Space for 128-bit Massively Parallel Computers

Eduardo Tomasi[1,2], César Fuguet[1], Christian Fabre[1] and Frédéric Pétrot[2*]

[1]Univ. Grenoble Alpes, CEA, List, F-38000 Grenoble, France
[2]Univ. Grenoble Alpes, CNRS, Grenoble INP, TIMA, F-38000 Grenoble, France

Abstract

High Performance Computing (HPC) supercomputers are composed of up to ten thousand nodes, each one having hundreds of cores organized around a shared memory. These nodes communicate through a high-performance communication network. Applications for HPC have increasing needs, both in terms of computational speed and the size of datasets to be processed. To follow these needs, memory in supercomputers is increasing at a rate such that, in the next decade, it will likely exceed 2^{64} bytes. The RISC-V 128-bit ISA gives us the opportunity to rethink how memory is addressed and virtualized at the scale of a supercomputer. We are working on a platform to simulate a distributed 128-bit system with a global address space shared by the whole supercomputer.

Introduction

Massively parallel supercomputers can process large amounts of data by distributing the computation over hundreds or thousands of independent compute nodes. Each node contains its own processing units organized around a shared memory and running its own instance of an operating system, and they communicate through a high bandwidth and low latency communication network and a dedicated networking software stack. These supercomputers are used, for instance, for Cloud Computing and for High Performance Computing (HPC).

An application consists of multiple processes that communicate by passing messages through the communication network. Inside a node, processes spawn threads that share data within a coherent memory space. Despite the complexity of mixing two programming models, MPI (Message Passing Interface) and OpenMP have become the de facto standard for high performance inter- and intra-node parallelism, respectively (Figure 1).

A typical workload for HPC is computationally intensive scientific applications, whose needs, both in terms of computational speed and the size of dataset to be processed, have been constantly increasing. To follow these needs, global memory in supercomputers is increasing at a rate such that, in the next decade, it might exceed 2^{64} bytes. The RISC-V community has already set the basis for a 128-bit architecture, allowing us to rethink the software architecture of supercomputers. It gives us an oportunity to reassess memory virtualization, in a way such that it is globally shared by all nodes participating in one application.

*Corresponding author: eduardo.tomasiribeiro@cea.fr

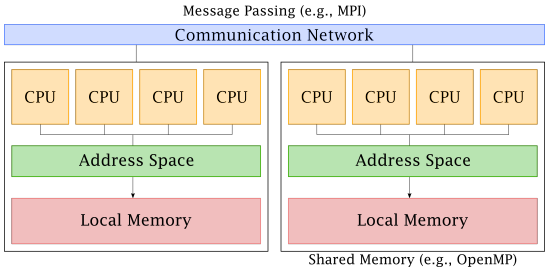

Figure 1: *Shared Memory is used for parallelism inside a node, and Message Passing for parallelism between nodes.*

Programming Models

Shared Memory

Within a node, memory coherence is ensured by hardware protocols, such that the CPUs and accelerators within a node communicate through a shared memory. This allows developers to parallelize a program with multiple threads sharing a single address space.

OpenMP is one of the main API specifications for shared memory parallel programming. Parallel execution can be achieved with a high level of abstraction, as communication and synchronization points are implicit. However, this approach is only viable as long as memory coherence scales, up to a few dozen cores, limiting its usage to internode parallelism.

Message Passing

Beyond a node, memory is physically distant and coherence cannot be efficiently guaranteed. Processes running the same application in different nodes do not have access to each other's address space, so they communicate through the communication network. This means that data can only be shared by message passing. Pointers, however, cannot be shared.

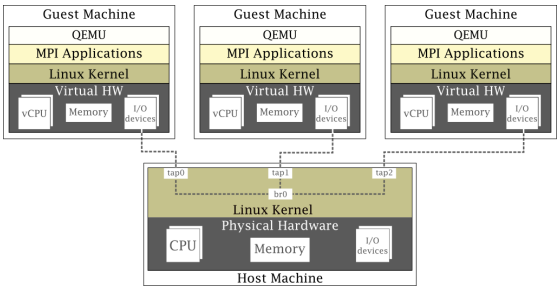

Figure 2: *Simplified diagram of the platform. The machines are connected using TUN/TAP interfaces.*

The main standard for parallel programming on distributed systems is MPI. It requires developers to explicitly distribute the workload and tasks, and to explicitly define the communication and synchronization points. As memory is not coherent, the programmer is also responsible for guaranteeing that all nodes have the most recent value of shared data.

Contributions

We developed a platform for simulation of a distributed machine with the 128-bit RISC-V base ISA to explore different programming models and architectures based on a distributed memory space (Figure 2). We chose QEMU for three main reasons: it is fast; it provides convenient interfaces to extend the simulator and add appropriate instrumentation; it supports the RV128 ISA [1]. However, there is not, to this date, any toolchain that supports this ISA.

Due to the lack of 128-bit applications, we started using RV64 and existing benchmarks, and will slowly extend it to the 128-bit solution.

This simulator allows users to have an early prototype of their distributed applications, so developers can tune them and prevent performance bottlenecks. In addition to the possibility of exploring different network topologies, multiple metrics will be provided to guide the programmer (e.g. number of atomic operations, cost in terms of instructions to perform MPI communications and number of system calls).

For our exploration of memory architectures for a 128-bit supercomputer, we will use this simulator to evaluate the cost of MPI calls, and the potential speedup when passing to a shared global address space.

Related Work

Some research teams studied this problem in the 1990s with Distributed Shared Memory (DSM) approaches, such as Ivy [2] and Munin [3]. These solutions failed to achieve scalability, due to a large run-time overhead due to software-based coherence protocols, and they were quickly replaced by multicore architectures.

Similarly, Jia et al. [4] addressed this problem with a distributed hypervisor that provides a many-to-one virtualization of the underlying hardware to enable distributed CPU, memory and I/O virtualization. However, it makes use of DSM algorithms to achieve memory aggregation, experiencing some of the overhead issues of other DSM implementations.

Wang et al. [5] proposed a RISC-V extension for scalable global addressing in HPC. It provides support for direct accesses to remote shared memory, aiming to reduce user libraries' and drivers' overheads. Our approach aims at achieving this at the hardware/software interface instead, rethinking memory virtualization and adding the necessary hardware support.

Conclusion

If today's rate continues, 64-bit supercomputers will likely run short of addresses in the following decades, if it all resided on the same address space. The RV128 ISA gives us the opportunity to start thinking about the solutions to this problem, and the challenges that may come with 128-bit architectures. Current programming models, however, do not seem to address the challenges of programming such supercomputers.

We think that this problem should not be addressed only at the software level, nor at the hardware level. We want to address it at the hardware/software interface, rethinking memory virtualization in the operating system, and adding the necessary hardware support. We have built a simulator of distributed systems, that allow us to profile MPI workloads and do the instrumentations to get to a 128-bit global address space.

Acknowledgment

The authors would like to thank the partners of the ANR Maplurinum project and acknowledge the financial support of the French Agence Nationale de la Recherche under grant ANR-21-CE25-0016.

References

[1] Fabien Portas et al. "Fast simulation of future 128-bit architectures". In: 2022 DATE.

[2] Kai Li et al. "Memory coherence in shared virtual memory systems". In: *ACM Trans. Comput. Syst.* 7.

[3] John B. Carter et al. "Implementation and performance of Munin". In: *SIGOPS Oper. Syst. Rev.*

[4] Xingguo Jia et al. "GiantVM: A Novel Distributed Hypervisor for Resource Aggregation with DSM-aware Optimizations". In: *ACM Trans. Archit. Code Optim.*

[5] Xi Wang et al. "xBGAS: A Global Address Space Extension on RISC-V for High Performance Computing". In: 2021 IEEE IPDPS.