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

Eduardo Tomasi 1,2, César Fuguet 1, Christian Fabre 1, 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

This work was funded by the French National Research Agency (ANR) under grant agreement ANR-21-CE25-0008 (MAPLURINUM)

CONTEXT

SIMULATION OF A DISTRIBUTED SYSTEM

QEMU

QEMU is an open-source machine emulator. Multiple architectures can be executed in a single host. It provides a virtual model of an entire machine (CPU, memory, and devices). We chose it for three main reasons:

1. **IT IS FAST**

   It uses dynamic binary translation (DBT) to reach very high simulation speed. Also, QEMU’s scalability on SMP machines is good [1].

2. **TCG PLUGIN**

   QEMU plugins provide interfaces to extend the simulator and add proper instrumentation. Plugins provide a mechanism to subscribe to events during translation and execution of instructions [2].

3. **RISC-V 128**

   QEMU already has support for RISC-V [3]. Global memory in HPC might, in the next decade, exceed 2MB bytes. It allows us to rethink the software architecture of supercomputers, including memory virtualization [4].

Scalability

Functional Results

Table I. Number of syscalls during MPI calls of NPB-I5.

Table II. Number of syscalls during MPI calls of NPB-DT.

Table III. Impact of the plugins during execution of NPB-I5.

Fig 1. An example of a typical clustered architecture, as can be found in high performance computing (HPC).

Fig 2. Shared Memory is used for parallelism inside a cluster, and Message Passing for parallelism between clusters.

Fig 3. How to use QEMU to profile MPI calls.

Within a cluster, memory coherence is ensured by hardware protocols, such that CPUs and accelerators communicate through a shared memory.

Beyond a cluster, coherence cannot be efficiently guaranteed. Processes in different clusters do not have access to each other’s address space. Data can only be shared by message passing through the communication network.

A global address space offers a simplified programming model for distributed systems. It provides a high-level abstraction of the memory, hiding the complexity associated to its management.

Programming Models

Non-Intrusive Workload Analysis

To retrieve the metrics of interest, we need to monitor the code during execution. We need to know beforehand the virtual addresses of the MPI functions in the binary code. Through a QEMU plugin, we can monitor the translation blocks executed and profile the calls.

Results

We can gather metrics such as: number of instructions during a MPI call, number of system calls and number of memory accesses.

References


