A directory/cache for leveraging the efficient use of distributed memory by task-based runtime systems

A directory/cache for leveraging the efficient use of distributed memory by task-based runtime systems

Tiberiu Rotaru, Bernd Lörwald, Nick Brown, Mirko Rahn, Olivier Aumage, Vicenc Beltran, Xavier Teruel, Jan Ciesko, Jakub Šístek

As the community progresses towards exascale the appearance of much more complex architectures than we are currently used to leveraging is likely. For instance, machines which are highly parallel, with a large number of multi-core nodes, deep memory hierarchies and complex interconnect topologies are around the corner. Efficiently programming these new systems will be a huge challenge, where programmers will not only need to fundamentally rethink their code to increase the level of parallelism by at least an order of magnitude, but to also address other issues such as resilience.

Task-based models [1] are one possible solution to this challenge, where parallelism is decomposed into many tasks. By rethinking their parallelism in the paradigm of tasks, one can significantly reduce synchronisation which is key for achieving high levels of concurrency. The underlying task model decouples the management of parallelism from computation. This relieves the application developers from dealing with lower level details such as scheduling, memory management and resilience concerns that are tricky to manage in large computing systems.

However task-based models are not a silver bullet and this paradigm often focuses around providing the abstraction of a single shared address space to the application programmer. To scale beyond a single physical memory space then some sort of distribute technology (e.g. MPI or GASPI) must be combined. This interoperability is either at the task based runtime level (and implicit to the programmer) or involves explicit communications calls provided by the programmer within tasks of their application code.

We have developed an API for a Directory/Cache [2] which can be integrated with task-based runtimes and seamlessly (to the applications programmer) provides the abstraction of a single shared address space. Supporting interoperability between task-based models and distributed memory technologies, whilst memory is physically distributed amongst the nodes the Directory/Cache enables this to be presented to the applications programmer as a single, unified memory space. The directory tracks what data is physically stored where and the cache is used for performance to avoid frequently retrieving the same piece of remote data. The main purpose of the Directory/Cache is to provide a set of services that support task-based runtime systems efficiently running distributed applications, while being able to consistently manage data stored in distributed memory or in local caches.

We have developed a reference implementation of our Directory/Cache API which, to illustrate the abstract and generic nature of our API, has been or is being integrated with the runtimes of the OmpSs, StarPU, GPI-Space and PaRSEC task-based models. The Directory/Cache API allows runtimes to be completely independent from the physical representation of data and from the type of storage used. This facilitates access through the same interface to an extendable list of transport implementations using different communication libraries such as GASPI and MPI and even on-disk storage or tiered memory.

In this talk we will describe the main concepts behind our API and the underlying architecture of the reference implementation. We will also present the results of integrating the Directory/Cache with popular task-based models and specifically the performance and scalability that this affords to real-world applications utilising this technology.

References

[1] Fernández, A., et al. "Task-based programming with OmpSs and its application." European Conference on Parallel Processing. Springer International Publishing, 2014.

[2] Hedo, J. "Run-time support for multi-level disjoint memory address spaces." PhD thesis, 2015