eResearch AUSTRALASIA2019

Embedded Nimrod: Straightforward HTC in HPC environments.

Zane van Iperen[*], David Green[*], Hoang Nguyen[*], and David Abramson[*]

[*]Research Computing Centre, The University of Queensland, QLD, Australia,
{z.vaniperen,david.green,h.nguyen30,david.abramson}@uq.edu.au

June 21, 2019

HTC (High Throughput Computing) is a class of research computing workload that typically consists of of large quantities of small (CPU/memory) footprint, short-lived, independent jobs. The goal is to churn through them as quickly as possible. Many HTC workloads involve parameter sweeps whereby the same processing task is performed for a range of values of one, or more, input parameters. One example of this is gathering a statistical sample by running the same stochastic task and collecting multiple outcomes.

HTC is generally regarded as different from HPC (High Performance Computing), due to its smaller footprint, independence of tasks, and shorter walltimes. However, it is not necessarily a clear distinction, because in some situations such as rapid design prototyping, a larger footprint HPC jobs could be employed repeatedly with short walltimes to rapidly optimize a design.

Many of the common HPC workload managers, such as PBSPro, Torque, and SLURM, do not efficiently handle such jobs. For jobs with walltimes in the order of minutes and seconds, scheduling and setup overhead can often outpace the duration of the job itself. Job arrays are available to provide a more efficient way for handling very large numbers of independent tasks; however, there is still an overhead associated with the commencement and completion of each batch system job, or job array element. They are also fairly limited in their support for parameter sweeps.

The independent nature of each job lends itself to the jobs becoming scattered across the HPC system. Traditional HPC workloads required much larger chunks of computing resources and small footprint, short walltime jobs can hold up the larger footprint workloads. From a resource management standpoint, it is more desirable to consolidate fragmented workloads such as job swarms into a subset of the HPC resources and to better manage the high turnover of tasks with a mechanism that has much more fine-grained control over the workload.

Embedded Nimrod is a software tool created to address this problem. Instead of having to schedule the numerous HTC jobs through the existing batch system, only one is scheduled; Embedded Nimrod handles the scheduling and execution from within this pool of resources. Similar to its predecessor Nimrod/G [1], Embedded Nimrod is capable of elegantly handling parameter sweeps. Yet it requires minimal changes to existing job submission scripts.

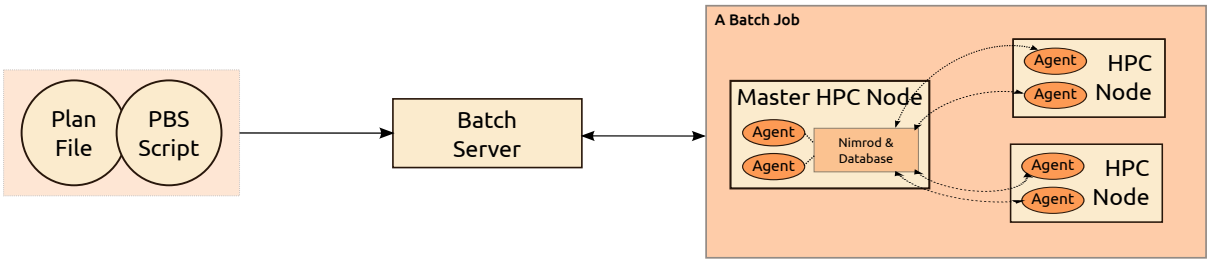

Figure 1: Embedded Nimrod Architecture Diagram

Embedded Nimrod has been used on the University of Queensland Research Computing Centre's HPC clusters (including Tinaroo, Awoonga, and FlashLite) and on Barcelona Supercomputing Center's NordIII. It has been used by several projects such as the Threatened Species Index, the first attempt to create an index for threatened species in Australia [2], and for inland drayage research at UQ, i.e. optimising the transportation of cargo containers inland from the Port of Brisbane [3].

References

[1] D. Abramson, J. Giddy, and L. Kotler, "High performance parametric modeling with Nimrod/G: Killer application for the global grid?" In *Proceedings 14th International Parallel and Distributed Processing Symposium. IPDPS 2000*, IEEE, 2000, pp. 520–528.

[2] NESP Threatened Species Recovery Hub. (2018). TSX – A threatened species index for Australia, Threatened Species Recovery Hub, [Online]. Available: `https://tsx.org.au/` (visited on 06/07/2019).

[3] Research Computing Centre, "Embedded Nimrod: The truck stops here," Dec. 12, 2018. [Online]. Available: `https://rcc.uq.edu.au/article/2018/12/embedded-nimrod-truck-stops-here` (visited on 06/07/2019).