Parallelization of software pipelines using the mpififo tool

Nathan Weeks, Marina Kraeva, Glenn Luecke

High Performance Computing Group, Information Technology Services, Iowa State University

Summary

mpififo is an MPI-based middleware that enables inter-node communication between non-MPI processes, allowing processes in Unix pipelines to execute on separate nodes.

mpififo is available under an Apache 2.0 license at `https://bitbucket.org/nathanweeks/mpififo`.

Motivation

- An increasing number of CPU-bound, memory-bound, and I/O-bound scientific workflows could benefit from distributed computation on compute clusters; however, many are unable to use more than one node because they weren't designed to use MPI for inter-process communication (IPC)—and retrofitting MPI into these applications can be a costly endeavor.
- A subset of those workflows are software pipelines consisting of multiple processes that communicate via sequential I/O to files or Unix pipes. POSIX named pipes (aka fifos) provide a mechanism for IPC between processes on the same node. **mpififo is an MPI-based middleware that extends this functionality by allowing IPC between processes on different nodes via named pipes**.
- *mpififo* can facilitate scaling by allowing the processes in a pipeline to run on separate nodes.

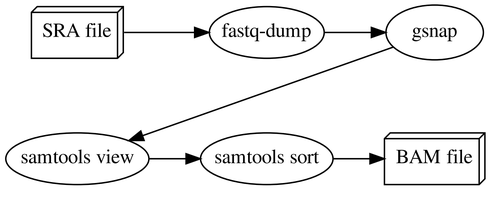

Figure 1: Single-node pipeline

Usage

One *mpififo* process must be launched per participating node using the following syntax:

```
[mpiexec...] mpififo [-cr] writefifo readfifo...
```

Data written to a *writefifo* on any participating node can be read from a corresponding *readfifo* on any participating node. *mpififo* will exit after each specified fifo has been opened and closed on any node.

Example

A simple bioinformatics sequence alignment pipeline is illustrated in figure 1:

fastq-dump (from the NCBI SRA Tookit[1]) converts an SRA file into FASTQ format.

gsnap [3] aligns the short reads to a genome, outputting alignments in the SAM format.

samtools view [2] converts the SAM to BAM format.

samtools sort outputs a sorted BAM file, ready for downstream analysis by a myriad of applications.

Figure 2 uses *mpififo* to facilitate the replication of compute, memory, and I/O-intensive stages on separate nodes. The following processes have been added to enable this:

partitioner A simple AWK script to distribute FASTQ reads in a round-robin fashion among the nodes.

samtools merge Merges the sorted BAM output from each node, writing a single sorted BAM file.

Disk I/O

Besides allowing concurrent execution of multiple processes, pipelines provide the benefit of minimizing disk I/O in workflows that can be structured to use them. In this example, disk I/O occurs only during the following stages:

- (**read**) SRA file by *fastq-dump*
- (**read**) Genome database (not shown) at startup of *gmap*
- (**read/write**) Temporary sorted files by *samtools sort* (can be avoided if sufficient free memory exists to store entire BAM output of *gsnap*—more feasible when using multiple nodes)
- (**write**) BAM file by *samtools merge*

Benchmark

mpififo was tested on cluster with compute nodes having the following specs:

- (4) 2.4 GHz quad-core AMD Opteron 8378 CPUs
- 64GB DDR2 memory
- 5-disk 3.1TB RAID-5 (local scratch space)
- RHEL 6.4 OS

The input SRA file contained 13.8 gigabases of soybean-seed single-end RNA-Seq data generated from an Illumina HiSeq 2000. On a single node, the pipeline took **8h 0m 17s**, whereas the multi-node version (4 nodes) took **3h 14m 29s**.

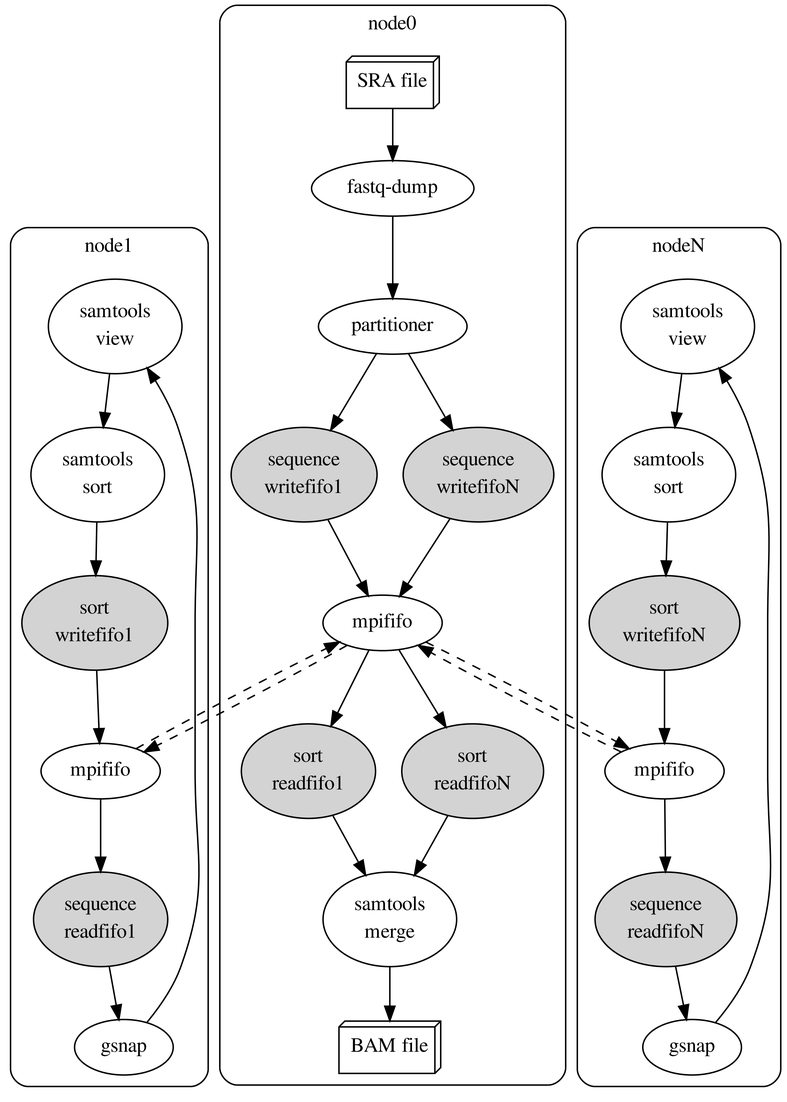

Figure 2: Multi-node pipeline

References

[1] R. Leinonen, H. Sugawara, and M. Shumway.
The sequence read archive.
Nucleic acids research, 39(suppl 1):D19–D21, 2011.

[2] H. Li, B. Handsaker, A. Wysoker, T. Fennell, J. Ruan, N. Homer, G. Marth, G. Abecasis, R. Durbin, et al.
The sequence alignment/map format and samtools.
Bioinformatics, 25(16):2078–2079, 2009.

[3] T. D. Wu and S. Nacu.
Fast and snp-tolerant detection of complex variants and splicing in short reads.
Bioinformatics, 26(7):873–881, 2010.