Computing and networking challenges in supporting streaming applications

Rajkumar Kettimuthu and Ian Foster
Argonne National Laboratory. Contact: kettimut@anl.gov

Data streaming applications require compute resources at a specific time, for a specific period with a high degree of reliability. Such requirements are hard to meet on current HPC systems, which are typically batch-scheduled under policies in which an arriving job is run immediately only if enough resources are available, and is queued otherwise. Other approaches are possible: for example, commercial clouds leverage the law of large numbers and some degree of overprovisioning to enable immediate scheduling of at least modest-sized tasks. But cloud resources alone may not be sufficient to support all streaming applications. For example, some applications might require large amount of resources that is available only at leadership computing facilities and big supercomputers. NERSC's Cori supercomputer supports real-time queue for jobs of modest size. Still, a number of challenges remain in supporting streaming applications on batch-scheduled HPC systems.
In addition, a congestion-free network path is required to stream data from data source to compute resource at a rate that is same as the data generation rate. Though significant work has been done in the past on dynamic provisioning of network resources, several challenges still remain.

Computing challenges:

What changes will be required to the scheduling policies, architecture, and implementation of next-generation (and current) supercomputers if they are to support streaming science workloads effectively? What kind of allocation and charging policies are suitable to accommodate real-time jobs?

What are implications for other batch workloads? Streaming jobs, in order to meet their requirements, will sometimes have to run before batch jobs that were submitted earlier. Hence, accommodating more streaming jobs will negatively impact the performance of batch job. In order to meet the constraints of streaming jobs, batch jobs may have to be preempted. The overhead involved in preempting and restarting batch jobs will, in turn, negatively impact the system utilization.

What hardware and system-level support can be beneficial? Boot time could be expensive on supercomputers; for example, boot time is as high as 5 minutes for large numbers of nodes. The boot time comprises more than ten factors, and the major costs are related to OS kernel verification and launch. We need to keep the preemption cost and boot time of a HPC machine really low in order to accommodate short-running streaming jobs.

Networking challenges:

NSF funded DRAGON and DOE funded OSCARS project enable the provisioning of wide-area network resources. The tools developed by these projects enable one to provisioning bandwidth from the border router of one institution to the border router of another institution. Projects such as TeraPaths and LambdaStation (both funded by DOE) and DYNES (funded by NSF) attempt to extend the circuit all the way to the end hosts. But still the process to setup an end-to-end circuit involves complex authentication process and manual steps including obtaining a certificate from a well known certificate authority, emailing wide-area network administrators and getting approval from the user's site network administrator. Because of this tedious process, only very few applications make use of the network reservation capabilities. Software-defined networking (SDN) can help address some of the issues but still a number of challenges remain. Particularly, it is unclear how an application can be given exclusive access to the reserved bandwidth.

The network allocation typically specifies a virtual circuit between two endpoint addresses, for example IP addresses and optionally TCP port numbers, assuming that the application component is directly connected to the IP address and specified ports. Network reservation services authenticate and authorize the user before allowing her/him to make a reservation. Once the reservation is completed, the network does not enforce the relationship between the user dataflow and the reservation. If the reservation is made using only the source IP

address and destination IP address, all applications transferring data between the source and destination during the reservation period will share the reserved bandwidth. And specifying the port numbers while making a reservation might work only for on-demand reservations and not for advanced reservations.

For streaming applications, when compute resources are reserved in advance, the exact IP addresses of the machines may not be known in advance. So, we need a mechanism that specifically binds the network virtual circuit to an application in an explicit manner, for example, through use of tokens rather than connection tuples. SDN technologies will ease the implementation and deployment of such a mechanism but research is required to identify the appropriate enforcement granularity and layers. For example, packet-level enforcement vs. control plane signaling.