Comparison of container virtualization tools for utilization of idle supercomputer resources

Wednesday, 10 June 2020 10:15 (15)

We propose a system to increase the effective load of supercomputer resources. The key idea of the system is that when idle supercomputer nodes appear, low-priority non-parallel jobs are started occupying these nodes until a regular job from the main queue of the supercomputer arrives. Upon arrival of the regular job, the low-priority jobs temporarily interrupt their execution and wait for the appearance of new idle nodes to be resumed there. This approach can be implemented by running low-priority jobs in containers and using the container migration mechanism to freeze these jobs and then run them from the point they were frozen at. While freezing a job, a stateful checkpoint is created that is a collection of files containing all the information for restoring the job execution (in the general case on another computing node).

Thus, the selection of a specific container virtualization system that is best suited to our goal is an important task. Preliminary analysis allowed us to choose Docker and LXC software products, which were compared in more detail.

When comparing the capabilities of Docker and LXC, it was noted that more lightweight Docker containers start and stop somewhat faster than LXC containers, the launch of which is more like starting a classic virtual machine. At the same time, the LXC project has the best support for the ZFS file system, which significantly speeds up the process of writing checkpoints to disk, as well as restoring containers from checkpoints. However, when testing the LXC, a recurring problem was discovered: the same container was correctly restored from a checkpoint only once. Attempts to checkpoint a container that was previously restored from a checkpoint resulted in an error with the loss of the container state. Since our project assumes a multiple checkpoint and restore of the same container as the main scenario, the above LXC feature prevents us from using this technology for the needs of our project. Thus, we opted for the Docker project, which stably and correctly checkpoints/restores any container multiple times while maintaining the current state of the processes. We implemented a prototype system to increase the effective load of supercomputer resources using Docker containers. Testing of the prototype proved the reliability and stability of the proposed approach.

Primary author(s) : DUBENSKAYA, Julia (SINP MSU); POLYAKOV, Stanislav (SINP MSU); NGUYEN, Minh-Duc; FEDOTOVA, Elena (SINP MSU)

Presenter(s) : DUBENSKAYA, Julia (SINP MSU)

Session Classification : Session 5