Why Containers on HPC

MPhil. Barbara Krašovec, Jožef Stefan Institute

Containers in high-performance computing (HPC) offer a flexible and efficient alternative to traditional modular software. Containers permit researchers and engineers to package applications and their dependencies into a single file, thereby ensuring portability, reproducibility and consistent performance across different computing environments. The advent of AI has led to an increase in the number of industrial users and SMEs engaged in HPC, and containers have become an even more convenient option for these users as they can also meet the security requirements for isolating user workloads. As the HPC community embraces container technology, it is ushering in a new era of more efficient, repeatable and scalable research computing. Nevertheless, the adoption of containers in HPC has been relatively slow, largely due to a lack of understanding about their functionality and concerns about potential performance degradation, energy inefficiency and security risks. Recent publications and developments have demonstrated that these concerns are unfounded, and therefore can be dismissed.

In terms of performance, containers can offer near-native speed if the following requirements are met:
- The software is built with the underlying hardware capabilities in mind.
- HPC-optimised base images are employed, including the requisite libraries for the underlying interconnect and MPI libraries.
- The image is comprised of a minimal set of software components, only those that are truly necessary.

It is advisable to avoid the inclusion of unnecessary layers in the container image, in order to reduce complexity and the potential for performance bottlenecks.

Another benefit of using containers is that they provide security and isolation. This is achieved through namespaces and cgroups, which are kernel features that containers leverage. Namespaces provide a layer of isolation by restricting what a container can see and access. Meanwhile, cgroups limit and allocate resources such as CPU, memory, and I/O bandwidth to containers. Additional security features can be applied, such as using Seccomp profiles or more granular privileges using Linux Kernel Capabilities. However, the security focus should extend beyond the provision of a secure operational environment to encompass the secure supply chain, which encompasses the entire process of container image provisioning. This necessitates the execution of security checks before the container images are used in the production environment. These checks may include vulnerability scanning, secret scanning, or testing the container for common exploits.

When utilising the containers, it is imperative that users adhere to the following best practices:
- The utilisation of a container on the system is not distinguishable from the execution of any other process. Consequently, if a container image comprises a vulnerable application, it is akin to executing a vulnerable application on the host system. Therefore, it is of paramount importance that container images are kept up-to-date and include the latest security patches.
- It is of the utmost importance that users utilise verified, trusted images from reputable sources.
- It is imperative that container images never contain any sensitive or personal data.

REPUBLIC OF SLOVENIA
MINISTRY OF HIGHER EDUCATION, SCIENCE AND INNOVATION

This project has received funding from the European High-Performance Computing Joint Undertaking (JU) under grant agreement No 101101903. The JU receives support from the Digital Europe Programme and Germany, Bulgaria, Austria, Croatia, Cyprus, Czech Republic, Denmark, Estonia, Finland, Greece, Hungary, Ireland, Italy, Lithuania, Latvia, Poland, Portugal, Romania, Slovenia, Spain, Sweden, France, Netherlands, Belgium, Luxembourg, Slovakia, Norway, Türkiye, Republic of North Macedonia, Iceland, Montenegro, Serbia.

It is recommended that data required to run workloads should not be copied into the container image, but rather should be bind-mounted in the container.
It is also advised that environments requiring a large number of small files (e.g. Conda environments) should be used in containers. A large number of small files can result in performance degradation when installed on parallel systems (e.g. Lustre).
Container images should be kept simple and lightweight. It is not recommended that all research software is installed into a container. The effort required to install all the software with its dependencies would be considerable, and the resulting image would be of a large size. Transferring such a big file to different environments would be time consuming and costly. Furthermore, the complexity of maintaining such an image would be very high.

REPUBLIC OF SLOVENIA
MINISTRY OF HIGHER EDUCATION, SCIENCE AND INNOVATION

This project has received funding from the European High-Performance Computing Joint Undertaking (JU) under grant agreement No 101101903. The JU receives support from the Digital Europe Programme and Germany, Bulgaria, Austria, Croatia, Cyprus, Czech Republic, Denmark, Estonia, Finland, Greece, Hungary, Ireland, Italy, Lithuania, Latvia, Poland, Portugal, Romania, Slovenia, Spain, Sweden, France, Netherlands, Belgium, Luxembourg, Slovakia, Norway, Türkiye, Republic of North Macedonia, Iceland, Montenegro, Serbia.