Containers Usage on the ATLAS Grid Infrastructure

A. C. Forti[1], A. Filipcic[2], P. Nilsson[3],T. Maeno[3], P. Love[4], A. DeSilva[5], L. Heinrich[6], A. De Salvo[7]

1. University of Manchester, 2. Jozef Stefan Institute, 3. Brookhaven National Lab, 4. Lancaster University, 5. TRIUMF, 6. NYU, 7. INFN Roma

Summary

Containerization is a lightweight form of virtualisation that allows reproducibility and isolation responding to a number of long standing use cases in running the ATLAS software on the grid. The development of Singularity, in particular with the capability to run as a standalone executable, allows for containers to be integrated into the WLCG infrastructure. Container technology enables a decoupling of Analysis Software and Site upgrade schedules. Further, it can efficiently address use-cases around software preservation and its distribution on sites without access to the CVMFS file system or the Grid middleware as well as improve the users development experience. While Singularity is easy to run, the variety of grid sites configurations and workflows still makes it a challenge to use it everywhere seamlessly. As usual the answer is to maintain a flexible system.

Flexibility vs Uniformity

One of the biggest challenges in distributed computing is to run user and production jobs anywhere on the infrastructure and produce results without the user caring where they run or what is underneath. This, on the grid, in practice translated to imposing an environment as uniform as possible on all the sites connected to simplify the already difficult operations of handling millions of jobs and files a day produced by hundreds of concurrent users. However both resources and user requirements are diversifying and the rigidity of a uniform distributed system cannot be supported anymore without loss of resources and analysis techniques. Using containers introduces the flexibility of environment needed. In particular it would address the following use cases:

Use cases

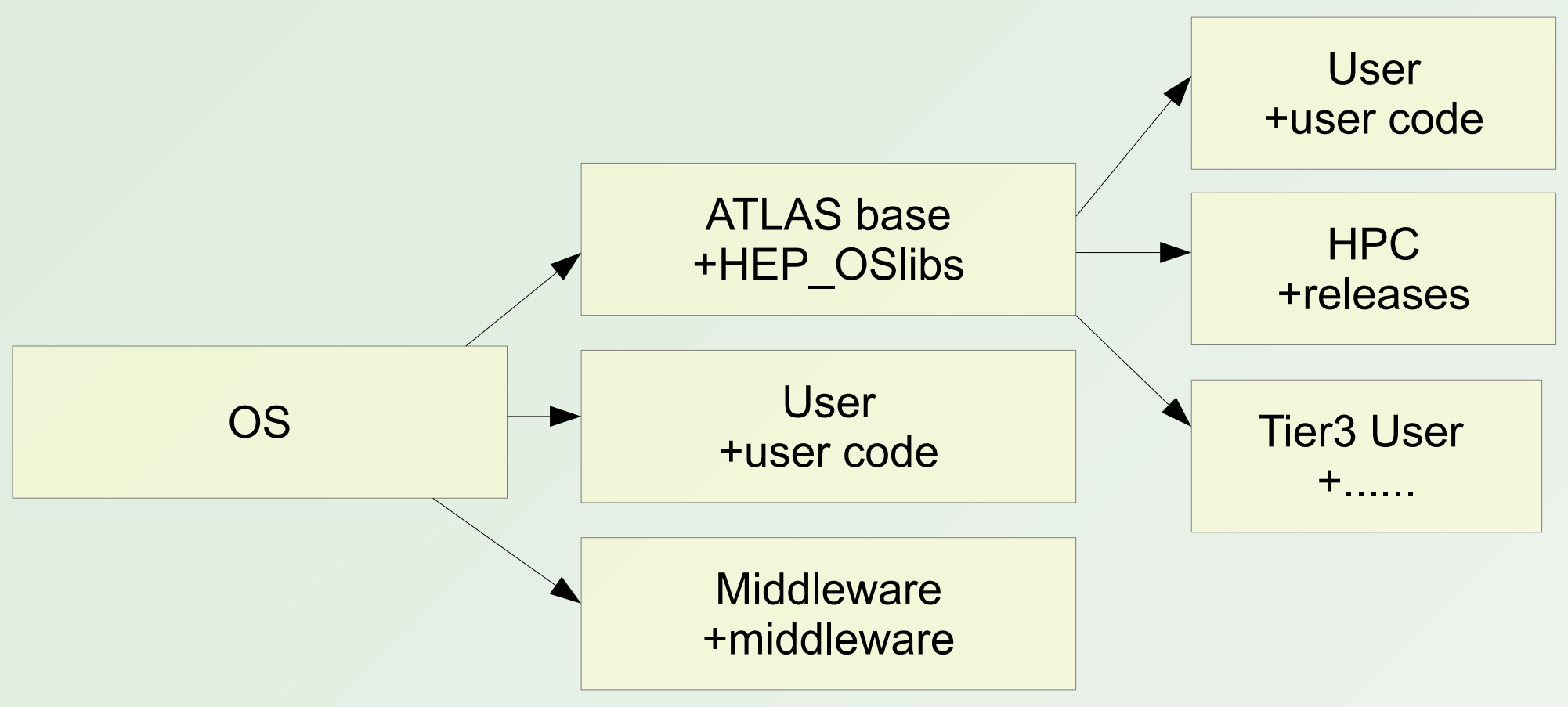
- Installation of different OS: ATLAS code can only run on RH family linux. Linux containers can be used to run different OS than the host machine OS. This allows ATLAS to run on sites that support a different distribution
- OS upgrades: even using the same OS everywhere, OS upgrades have always been a challenge and required a big coordinating effort between experiments and sites
- Minimal installation on the nodes: manpower at sites is shrinking and there is an effort to reduce the amount of software that needs to be maintained by them
- Allows experiment to run tests with specific software or setups without requiring sites to install specific environments
- Allows software preservation: older releases don't need to be obsoleted anymore
- User analysis development cycle can be better integrated in the infrastructure and new types of analysis can be developed
- Offers another approach to software distribution to sites that don't support CVMFS and this is particularly important for HPC resources
- Can offer payload isolation if run by the PanDA (ATLAS WMS) pilot

Singularity and Docker

There are several types of containers on the market. In HEP environment the main choices are 3: Docker, Singularity and Shifter. Docker is the most developed and used by users, but requires daemons and some integration with the batch systems to run and is therefore difficult to deploy with a certain flexibility on the grid. Singularity is instead a really light weight, easy to install executable, that can run as a non-root process. It has a simple configuration that sites can deploy without much trouble. In addition Singularity images can be easily generated using Docker images. Shifter is similar to Singularity and is used on some of the HPC systems.

Images

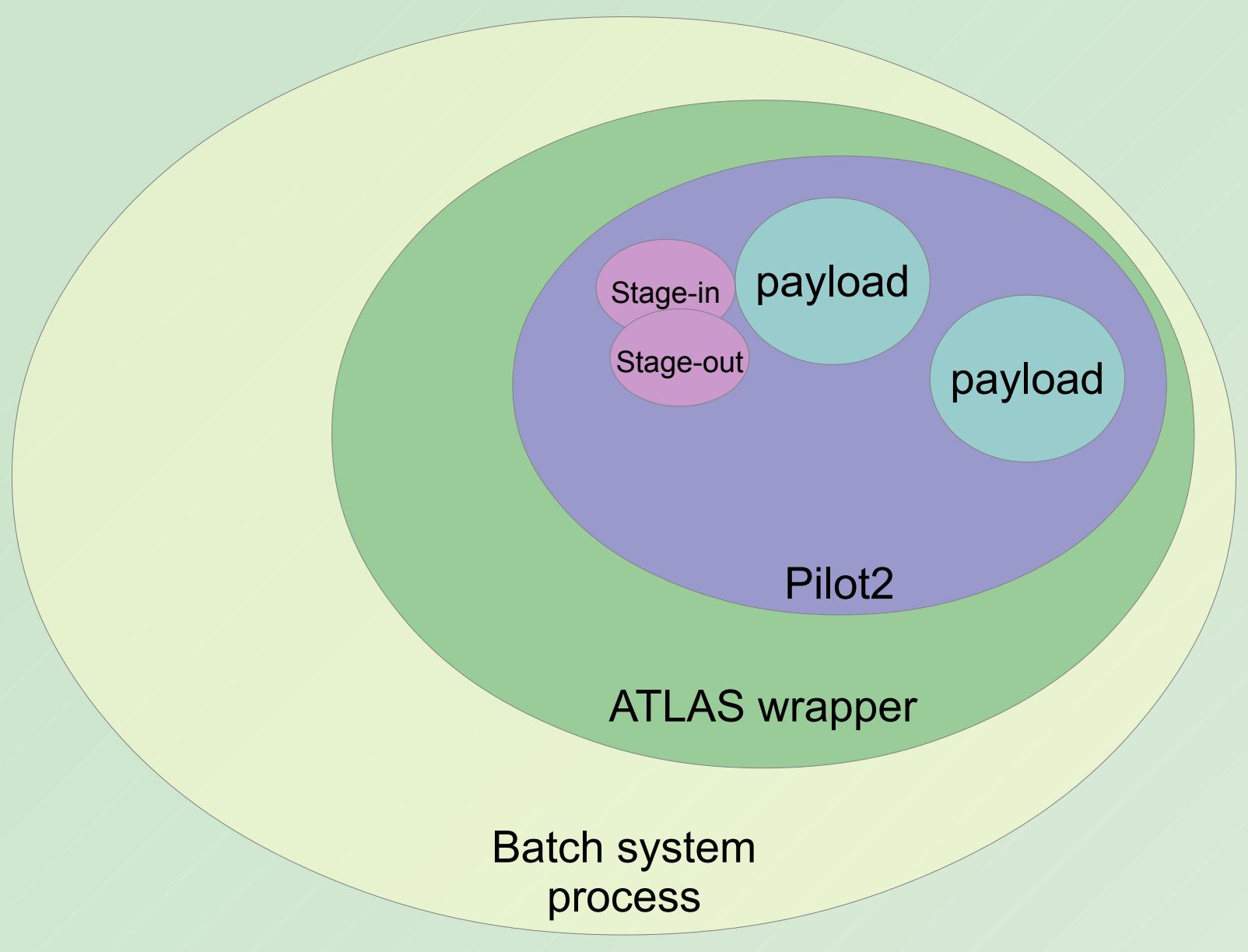
Images have to be as small as possible and at the same time satisfy each particular use case. To avoid putting everything in each image we decided to have a hierarchy of images. And for each use case we add only what is required.



There is also a problem with image distribution and image format. On the grid images will be distributed via CVMFS, if available, and will be unpacked as CVMFS directory trees; where CVMFS is not available we will have to distribute files as if they were data. Users also maintain their code on Dockerhub which Singularity can read directly creating the images on the fly.

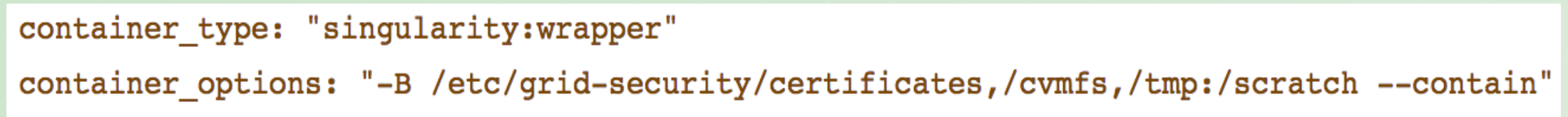
Atlas Infrastructure

Singularity can be called in different layers of the infrastructure with different use cases supported at different levels. If the site runs the containers it still satisfies the first 2 use cases but doesn't allow the experiments the flexibility needed to chose the images. The most flexible level to cover all the use cases run the container at the level of the pilot. This will allow to isolate the payload from the pilot and, since the new PanDA Pilot2 (now in late stage development) is multithreaded, to even run the payload and the middleware on a different OS. It will also allow to isolate the payloads from each other in case of multi payload pilots.



Containers Options in AGIS

To allow maximum flexibility to select the layer to run, type of container, if to run the middleware in a container and pass options to the container, we have introduced two new fields in AGIS (ATLAS Grid Information System): container_type and container_options.



```
container_type: "singularity:wrapper"
container_options: "-B /etc/grid-security/certificates,/cvmfs,/tmp:/scratch --contain"
```

In most cases the container type will indicate to run Singularity at pilot level, but it will be possible to have other combinations. The main reason the options have been added is to explicitly list bind paths. Bind paths are the directories that have to be mounted from the host system for Singularity to run. For most sites on the grid this might not be needed because pilot2, instead of calling Singularity directly, calls it through ALRB (AtlasLocalRootBase) system of scripts and ALRB works out what is needed from environment variables and sets up the ATLAS appropriate environment within the container. However there are sites with extra number of directories like caches that might need to be added or other options the site want the pilot to run and this can be configured on a per PandaQueue basis. We would like to keep this to a minimum though to avoid sites having to communicate internal details that shouldn't be known and also minimize the number of directories that should be listed in the image if overlayfs cannot be used, so other solutions are being developed in cooperation with WLCG with whom also security considerations are being evaluated.

Pilot2 Development and Testing

The containers initial development work was done at wrapper level thus demonstrating proof of concept after containers run successfully jobs at a couple of sites. It is now carried out using Pilot2 which is a complete rewrite of the pilot currently used by ATLAS. Pilot2 is currently running containers successfully in all 3 systems ATLAS uses to submit jobs the pilot factories both in Europe and in the US and at a site that is behind the aCT (ARC Control Tower). Regular tests have been setup so a small containerized payload is run at the participating sites and changes to the system (switching options on and off, Singularity upgrades etc) can be tested easily. Every couple of weeks we add an extra site or two to expand the range of site configurations and running conditions (single core, multi core, high memory, analysis) and see if the system needs changes. So far the queues have been production and we have now started to test also user analysis. To support user requirements, the pilot was modified to accept the image as an option rather than selecting it automatically from the architecture of the release.

HPC and user analysis have are different uses cases, but what they have in common is they need more customized images not necessarily available in CVMFS. The pilot selects the image automatically using the release architecture, but, as for users, also for HPC it will be able to override that selection. A preliminary naming scheme for the images mirroring the releases naming scheme was also discussed for HPC, a second iteration with the users is needed.