Continuous Delivery of Research Application in a Distributed Environment

Annotated Bibliography

Sakhile Masoka (851667@students.wits.ac.za)
Witwatersrand University

May 4, 2015

References

[blo 2012] *Status and future perspectives of CernVM-FS*, volume 396. IOP Publishing, 2012. International Conference on Computing in High Energy and Nuclear Physics 2012.

> **Aim:** To present a new approach to stage updates and changes into the file system, which aims to reduce the delay in distributing a software release.
> **Style/Type:** Technical.
> **Summary:** CVMFS is a read-only file system used to access High Energy Physics experiment software and conditions data. Files and directories are hosted in global namespace (/cvmfs) and grid worker nodes on remote sites are able to mount these CernVm-FS repositories. To further reduce the time to publish delay, a read-write interface is created using the union file system, stacking CernVM-FS read-only file-system with a writable scratch area to track changes on the local node. Also, the publishing changes on the file systems is distributed to multiple machines, which improves the performance. With these improvements, CernVM-FS can be used as an exclusive software distribution system.

[Boettiger 2015] Carl Boettiger. An introduction to docker for reproducible research. *ACM SIGOPS Operating Systems Review*, 49:71–79, 2015.

> **Aim:** To present how Docker addresses challenges facing reproducible research
> **Style/Type:** Conference article
> **Cross Reference:** A contrast is made between two paradigms that currently have emerged, work-flow and virtual machines. Both have struggled in the scientific community to be successful.
> **Summary:** Reproducible research has gained interest in scientific communities and the public at large. The paper presents four main technical challenges that poses substantial barriers to reproducing the original scientific results, namely “Dependency Hell”,Imprecise documentation, Code rot and Barriers to adoption and reuse in existing solutions. Current approaches such as work-flow, virtual machines and “DevOps” does not effectively and elegantly address these challenges, but Docker which offers several promising features for reproducibility can. Docker images can resolves “Dependency Hell”, Dockerfiles can resolves imprecise documentation and the work-flow docker provides, eliminates code rot and barriers to adoption and re-use. Docker is presented as portable and has the potential to address shortcomings of certain existing approaches to reproducible research challenges that stem from recreating complex computational environments.

[bun 2010] *CernVM a virtual software appliance for LHC applications*, volume 219. IOP Publishing, 2010. 17th International Conference on Computing in High Energy and Nuclear Physics.

> **Aim:** CernVM aims to provide a complete and portable environment for developing and running LHC data analysis on any end-user computer (laptop, desktop) as well as on the Grid, independently of Operating System platforms (Linux, Windows, Mac OS).
> **Style/Type:** Conference article
> **Summary:** CernVM is a lightweight Virtual Software Appliance intended to run on any operating system platform providing consistent and effortless installation of

experiment software. CernVM runs a file system (CVMFS) which is optimized for software distribution. Its makes the directory tree stored on the web-server(CernVM) look like a local read-only file system on the client side. This allows to centrally install and maintain software on few CernVM instances that run CVMFS and allow many clients to connect to access the software. File level caching on local disks further assists with client scalability. Compared to Andrew File System (AFS), CVMFS shows competitive performance figures, especially when the local disk is fully cached. Further improvement have been implemented in experimental versions to reduce https overheads and transfer volumes. By leveraging the standard https protocols, this technology is able to become a software distribution service.

[jen 2015] *Jenkins Continuous Integration*. `http://jenkins-ci.org`, 2015. Accessed 2015-04-20.

Aim: Official website for Jenkins, a continuous integration tool
Style/Type: Website
Summary: official website for Jenkins, an open source continuous integration tool. The software monitors executions of repeated jobs, such as building/testing a software project continuously and monitoring executions of externally-run jobs like cron-jobs. For each commit, Jenkins runs, test and reports back with feedback on you application deployment. These deployment can be directly to production or test environment which are distributed. Jenkins can be extended using plugins for additionally supporting processes not native to it. Further more, integration with gitHub is made easy.

[joa] *Developing knowledge systems with continuous integration*. ACM, New York. Proceedings of the 11th International Conference on Knowledge Management and Knowledge Technologies.

Aim: To transfer ideas of Continuous Integration from Software Engineering to Knowledge Engineering and demonstrate the implmentation of Continuous Integration tool into a Knowledge Engineering workbench.
Style/Type: Conference Paper
Summary: Continuous Integration (CI) proposes a collection of practices for the engineering of software systems in order to improve the overall quality of the system. These practices which are (1) User of code repository (2) Automated tests on all levels of the software (3) Automated building of the software (4) Frequent and timely integration of new code (5) Easy access to the latest builds, can be implemented in most Knowledge Engineering tools. With the integration of CI dashboards for visualization of the development process, risk is reduced and providing a running system at any time of the development process becomes possible. The paper ends with an implementation of CI to Knowledge Engineering tool called Semantic Wiki.

[lui 2014] *The research and implementation of cloud computing platform based on docker*. IEEE, 2014.

Aim: To describe Dockers applications and advantages in a distributed environment.
Style/Type: Conference article
Cross Reference: The report compares Docker with Vmware ESXi vmw [2015], with the one difference of virtualized applications includes not only the application and the necessary binaries and libraries, but also an entire guest operating system while the Docker Engine container comprises just the application and its dependencies which

makes it more portable and efficient.
Summary: Docker is an open-source project that automates the deployment of applications inside software containers, by providing an additional layer of abstraction and automation of operating system. Mainly used by developers and systems administrators, Dockers enables applications to be quickly assembled, from components and eliminates the friction between development, QA, and production environments. Docker's architecture, allows for portability and is lightweight compared to virtual machines as the engine allows containers to share the kernel while maintaining their isolation. Docker makes use of the following technologies to gain its advantages, namely, Namespaces, Control group (cgroup) and Union File systems (UFS). The article continues with an example implementation in a cloud environment Platform-As-A-Service configured.

[Meyer and CI 2014] Mathias Meyer and Travis CI. Continuous integration and its tools. *IEEE Software*, 31:14–16, 2014.

Aim: To present the basics of continuous integration and its tools
Style/Type: Journal Article
Summary: The paper describes continuous integration (CI) as a set of principles that apply to the daily work-flow of development teams. First, all code must be kept in a repository. Secondly, when code is checked in the repository, the system checks the code and runs test to verify that the change is good. The core practice of CI is that small changes are committed to the main-branch daily instead of big changes once in a while. For these principles to work, the continuous integration server has to be unbiased and tools such as Jenkins ensures this. The principles combined with the tools brings culture of responsibility and ensures a safe development and deployment of applications.

[vmw 2015] *VMware ESXi.* `http://www.vmware.com/products/esxi-and-esx/overview`, 2015. [Online; accessed 2015-04-20].