LHCB’s Puppet 3.5 to Puppet 4.9 migration

Three weeks long hackathon

H. Mohamed\textsuperscript{1}, F. Sborzacchi\textsuperscript{2}, T. Colombo\textsuperscript{1}, L. Brarda\textsuperscript{1}, N. Neufeld\textsuperscript{1}
R. Schwemmer\textsuperscript{1}, M. Daoudi\textsuperscript{1}
1. CERN, Geneva, Switzerland
2. INFN, Rome, Italy
hristo.mohamed@cern.ch

Objectives
- Migrate our entire code base from Puppet 3.5 to Puppet 4.9
- Put high availability on every component of our Puppet infrastructure
- Scale our Puppet infrastructure to be able to initiate a site-wide run in predictable time
- Scale out MCollective installation
- Do all of the above in the time frame of a month!

Introduction
Up until September 2017 LHCB Online was running on Puppet 3.5 a Master/Client non-redundant architecture. As a result, we had problems with outages, both planned and unplanned, as well as with scalability issues (unable to predict amount of time needed to complete site-wide run). What’s more, Puppet 5.0 was stable and around the corner!

For the redo of our code base, we used the following principles
- Always use modules from the forge(if available), never write your own
- (API) Specific data for our infrastructure goes to Hiera
- Write every module like an API
- Reuse profiles whenever possible
- Start using C/CED pipeline

New Infrastructure\textsuperscript{1}

Performance tuning and other considerations
Below are some performance tuning we used in our infrastructure
1. Test for yourself if deadline/cfq/noop performs better in your environment
2. Enable G1GC garbage collector with 4GB+ Heaps
3. Monitor PuppetDB performance drop and regularly initiate a vacuum\textsuperscript{2}
4. For Foreman running behind Passenger ensure maximum number of instances from startup (process spawning is expensive)\textsuperscript{3}
5. If using Passenger with Apache use MPM and a large number of workers (due to blocking I/O)\textsuperscript{3}

Performance tuning of Puppet Masters - JRuby
1. Puppet Server uses JRuby instances. Default value are between 1 and 4 (num-cpus - 1)
2. Minimum 512 MB of memory per JRuby with light catalogs / small number of environments
3. Ensure your system has adequate entropy

Results
1. We now have over 2500 hosts running our highly available Puppet
2. We can tolerate failure in every component
3. On average our infrastructure needs 17-20 seconds catalog compilation time
4. Full run of Puppet on our entire infrastructure takes less than 30 minutes!

Conclusion
Puppet’s initial release was in 2005. Since then it has come a long way to become a complete configuration management solution. Nowadays for a successful Puppet Open Source deployment, multiple components are needed as well as following soft engineering principles.

References

Additional materials used

\begin{itemize}
\item Pro Puppet
\item Turnbull, James, McCane, Jeffrey
\item http://www.brendangregg.com/linuxperf.html
\item Brendan Gregg
\end{itemize}