Introduction:

As someone who deals with Puppet quite a lot at work, I had the great pleasure of speaking to longtime open source pundit James Turnbull, who recently co-authored his latest book "Pro <http://www.apress.com/9781430230571>
Puppet" through <http://www.apress.com/9781430230571> Apress Media with colleague Jeffrey McCune of Puppet Labs. This is his fifth technical book about open source software. "Pro Puppet" is an in-depth book about how to install, use, and develop Puppet, the popular open source systems management platform used by organizations including Twitter, Rackspace, Digg, Genentech and more.

Q&A with James Turnbull

1. What in your estimation is the number of servers (including virtual instances) that run Puppet at any level of capacity?

A: This is a question that I ponder every few months. Our largest installation is around 50,000 nodes and we have several more at the 25,000 to 50,000 node range. Given the size of the community, I think we've quite easily reached the million plus node mark.

2. Throughout your work on the book, have you had a chance to measure/survey the operating systems on which Puppet is deployed? Have you any insight regarding the distribution of usage?

A: Puppet Labs did a survey earlier in the year and gathered some data about usage. Based on that and interactions with the community I think we can pretty comfortably say that our core operating systems are Linux-based with Red Hat (and derivatives) and Ubuntu/Debian being the biggest platforms. The next largest block is Solaris with a smaller number of OSX, *BSD, HP UX and pSeries/AIX systems also being represented.

3. There is a common perception that Free/Open Source software suffers from deficient documentation and lack of support (despite this being the business model of many companies). How do you challenge these types of allegations?

A: This is a common perception that regularly makes me laugh. I usually respond that *all* software has deficient documentation and lacks support! It's true some open source tools lack documentation but others, for example MySQL, have exemplary documentation. Some open source software communities are hard to get help from and others

fall over themselves to help people out. I'm always immensely proud of how the Puppet community, which is largely made up of some of the busiest people in IT - sysadmins, goes out of its way to help newcomers and share knowledge.

Of course this same problem is present across enterprise and commercial software. Otherwise authors wouldn't be able to sell books offering insights into using commercial software. :) It's even perhaps somewhat worse for enterprise software where submitting a bug request can lack transparency and where examples of how others have solved issues can be hard to find or perceived as proprietary information.

4. How can your book address or assist a crowd of people with no prior knowledge of UNIX/Linux and how can it assist those who are familiar with everything but Puppet?

A: Pro Puppet is aimed at users with some Linux/Unix knowledge, albeit at a fairly basic level -- a few friends and I created an earlier book called Pro Linux System Administration designed to teach someone with zero Linux knowledge how to be a Linux sysadmin. Pro Puppet is aimed at junior and mid-level sysadmins looking to get started with Puppet and take them through to advanced topics like scaling and extending Puppet.

5. What impact do you foresee the licensing changes from the GPL to the Apache licence as having?

A: Both the GPL and Apache licenses are free and open source licenses and we're very much staying true to our open source roots. However where we are with Puppet now we need a license that people, for whatever reasons, consider easier to integrate with. In the open source world that license is Apache and we're already starting to see Puppet being used heavily as an integrated tools in Cloud and Infrastructure/Platform as a service (IAAS, PAAS) offerings as a result.

6. Manual operators of Puppet seem to rely mostly on the initial setup. What proportion of the work would you say a Puppet expert needs to invest in setting up the software compared to the overall lifetime of a box and its operation?

A: With Puppet, the large proportion of the work you need to do to get started is up front. Once you've done that work setting up new boxes becomes a routine and easy task.

Maintaining and managing them is also fast and simple. Indeed, one of the benefits of Puppet is that not only do you get fast and automated setup, but you can make sure they stay the way you configured them for as long as you need. That ability to stem the tide of configuration drift and limit the potential for human error and entropy causing issues is an enormous timesaver.

7. What is the most eccentric/fascinating/uncommon use of Puppet that you have come across?

A: One that fascinated me recently is the Deutsche Flugsicherung, the German air traffic control network, who use Puppet to ensure all the operator workstations and tower servers are up to date. They have a very strict and structured work flow and an interesting deployment model where any configuration drift is anathema. I also find Air Traffic Control really interesting (I'm a geek it's true) so it was pretty exciting to see Puppet being used in such an interesting arena.

8. Puppet functionality lags behind in platforms such as Windows. What would you advise organisations that choose to run it on this platform?

A: We're actively working on Microsoft Windows support but we're not there yet. What we'd love to see is people telling us what they need. I'm not primarily a Windows guy so I actually don't know what the pain points are for Windows sysadmins. If a few of them could tell us "If you automated these 4, 5, 10 things that would make my job easier!" then that would help us structure that future support.

9. How does Puppet compare to its proprietary counterparts?

A: I think the key difference is time to value or as I prefer "how long before I'm doing something useful". Often when you install one of the larger proprietary tools it can take significant time and people to deliver value or to get things done. We find people can download Puppet, install it and be doing something useful in a matter of minutes or an hour rather than months.

10. If one receives proper training or learns from your book, how would the difficulty of using Puppet compare to the difficulty of using other products that are out in the market?

A: I think Puppet is pretty easy to use (but I'm also biased!). It does have rough edges and things that are hard to get

your head around though. One thing I think we do really well in the book is build on knowledge. You can start simple and grow into the more complex topics. I think having that sort of resource makes it really easy for people to learn how to use Puppet. The other resource I'm really excited about is a new section in the documentation called Learning Puppet (http://docs.puppetlabs.com/learning/) that offers a similar "grow into using it" experience.

I think as a result of having the book plus documentation and training available that makes Puppet a lot less difficult to understand than some of the alternatives out there.

11. How would you say the Puppet learning curve compares if a programmer and non-programmer were both faced with the task of learning it?

A: I recently came to the conclusion that I now spend more time cutting code than I do being a sysadmin which is a big change in my life. As a result I've been thinking about how both groups approach learning and problems. I think for a lot of sysadmins Puppet is very easy to engage with. Puppet's language is a logical extension for people use to dealing with configuration files and scripts.

For developers that's perhaps not as natural a progression and some have struggled in the past with learning Puppet. One of the new features in Puppet 2.6.0 though was a Ruby DSL for Puppet. This allows any developer (and sysadmins too) to write their Puppet manifests in Ruby. This approach is something that may make more sense and make it easier for developers to learn Puppet.

As a result of this Ruby interface (which we cover in the book too) I think the learning curve for both non-developers and developers is rapidly approaching parity.

Conclusion:

We would like to thank James for being available for this interchange of insights and we hope his literature will spread Puppet to more and more companies, aiding the spread of Free/open source in systems management. Puppet sure helps the company that I work for.