New Ideas For PanDA Packaging

Sean Wilkinson
University of Texas at Arlington
BigPanDA TIM, 25 April 2018

My background

In undergrad and early grad school, I studied various forms of math: computational and applied math, statistics, and pure math.

I ended up with a PhD in biomedical engineering for work on “web computing for bioinformatics applications” that I performed as a trainee in Pathology.

QMachine (https://www.qmachine.org/)

My “claim to fame”: a volunteer computing service that can use web browsers as compute nodes, coordinated by an HTTP API server backed by any of several databases.

It is the most accessed paper of all time in *BMC Bioinformatics*, so I’m sure all of you have read it ;-)

Convention

NOTE: When I am talking about QMachine (QM) throughout this presentation, I am referring to a complete system, which includes an API client and an API server connected to storage of some kind.

I use the same convention when I’m talking about PanDA.

So I join this group and start learning PanDA

Everyone says “Go look at the Twiki” but that doesn’t work for me, because

- I want to try it out and play with it live, and
- I want to see the code, but
- I don’t have CERN or ORNL accounts when I start, and
- I am not a High Energy Physicist!

First try: my personal laptop

I have a Mac laptop with Git and Homebrew, but I find that there is no documentation for

- building PanDA from source,
- downloading a PanDA installer, or
- downloading a pre-installed container or VM image.

Second try: ATLAS Tier 3 cluster via SSH

I wait to get an account from CERN to try this.

I can use our Tier 3 cluster with CVMFS in order to try it live, but I can’t clone the source from GitHub because the Tier 3 doesn’t have Git.

Our Tier 3 has to clone the PanDA source code by using SVN or zip archives.

Why is this so complicated?

So, as long as I have proper accounts, I can try out the PanDA workflow while I use my laptop to read the code, but

- Why am I having to work so hard, just to try it out?
- Would other “outsiders” like me even try to navigate any of this?
- How do the developers test new release candidates?

Wait, wait!

Please keep in mind that I am not telling you that PanDA is a bad project.

I am telling you that significant obstacles face “outsiders” who might otherwise try PanDA and like it.

The same things that will help outsiders will also help us!

The Outside World

I’m coming from the world of web development and cloud computing, where Git and package managers that use Git are commonly used.

[Live demonstration of cloning, building, and running QM.]

QMachine project structure

Parent “super-repository” links to exact versions of project component repositories as “sub-repositories”.

browser-client @ badf00d	Some Quanah-inspired changes for the JS
chrome-hosted-app	Bumped version number to 1.2.4
homepage	Bumped version number to 1.2.4
node-app	Bumped version number to 1.2.4

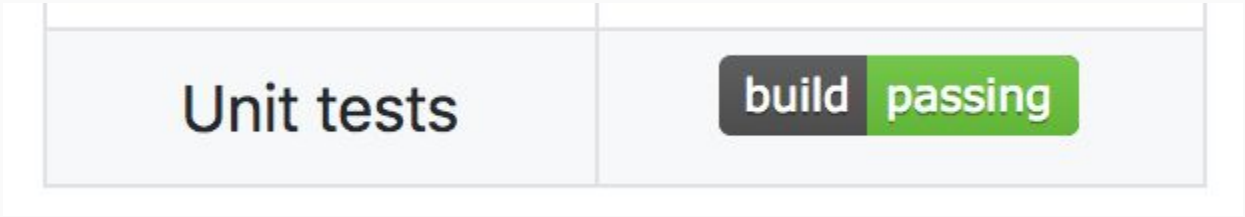
QMachine project structure

There are LICENSE and README files for every repository, and release versions are indicated by matching Git tags.

LICENSE	Added Apache License 2.0
Makefile	Stop deleting locally cached images for docs
README.md	Removed comments from Markdown

QMachine project structure

Every commit triggers documentation to rebuild and unit tests to run, and current statuses are publicly visible.



This screenshot of the README shows the Travis CI badge.

QMachine project structure

Project components are published to their respective global package managers (Bower, NPM, Gem, et al.).

```
$ bower install qm # for browser JavaScript client

$ gem install qm # for the Ruby server implementation

$ npm install qm # for the Node.js server
```

Long-term goal

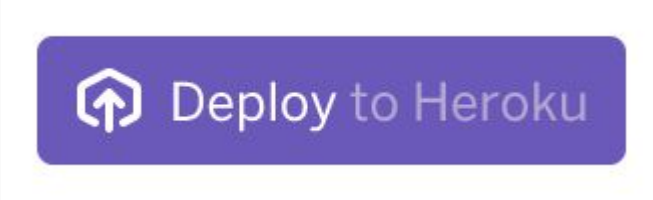
My vision for PanDA is to be able to install it from PyPI, the Python Package Index, via

```
$ pip install pandawms
```

Other components would follow a naming convention like “pandawms-foo”.

One-button deploy to the cloud

Well-described, self-contained apps become easy to port to a variety of environments. Platform-as-a-Service vendors like Heroku (https://www.heroku.com) have streamlined the deployment process all the way to a single button click.



QMachine Ruby turnkey app

A teaching version of QM is hosted as a repository on GitHub which can be installed to Heroku with one click:

https://github.com/qmachine/qm-ruby-turnkey

[Live demonstration of the Heroku Ruby turnkey app]

Turnkey app security warning

When your app is well-described by package manifests and the like, you get free help from the open source community!

⚠ We found potential security vulnerabilities in your dependencies.

Some of the dependencies defined in `./Gemfile.lock` have known security

Review vulnerable dependencies

Short-term goal

Docker (https://www.docker.com) is an open platform for developers and sysadmins to build, ship, and run distributed applications, whether on laptops, data center VMs, or the cloud.

PanDA could be ported to run inside Docker more quickly.

Benefits and advantages to Docker strategy

It will allow us to hide the ugly installation details from users while we streamline the process.

We already have some experience running PanDA in Docker on Titan.

New users can still try PanDA with just a few commands.

QM turnkey app as a Docker stack

So now, here is the same QM app I keep showing, but this time running as a Docker stack inside containers:

https://github.com/qmachine/qm-ruby-turnkey-docker

[Live demonstration]

Doesn’t PanDA already run on Docker?

Yes, a containerized version of PanDA is running at ORNL, but its code has diverged significantly from the sources.

That version will still be useful, however, as a template for constructing a general Docker stack.

Goals for PanDA on Docker

- Builds from the latest sources automatically
- Requires only Docker to be installed on host machine(s)
- Installs either from GitHub-hosted files or DockerHub
- Uses a general purpose Pilot (maybe mini-pilot)
- Runs offline once installed
  - SchedconfigDB skips AGIS

DockerHub as “package repository”

There is a test image at https://hub.docker.com/r/qmachine/qm-ruby-turnkey/.

Docker as “package manager”

Images can be updated in-place as the result of Git commits that pass unit tests. (Thanks, webhooks!)

Images can be downloaded automatically and run via

```
$ docker run user/project:tag
```

Rucio uses DockerHub

Rucio already publishes to DockerHub: https://hub.docker.com/u/rucio/

This immediately raises a lot of interesting ideas in my own head, although I haven't personally learned anything about Rucio yet. Maybe PanDA + Rucio in Docker someday?

Summary

Nice to meet you!

For PanDA to go mainstream, it needs to be streamlined.

Docker is my proposed short-term strategy for simplifying BigPanDA into a “MiniPanDA” with some catchy name.

PanDAMachine, maybe? :-)

Thanks!

Questions?