Tryton WSGI Deployment
A field report

Robin Baumgartner
robin.baumgartner@leuchterag.ch

Tryton Unconference Leipzig 2014
Outline

1. About me
2. Why WSGI?
3. Using Tryton with uWSGI
4. Problems, Solutions & Tricks
5. Discussion
About me

- Working at Leuchter Open Source Solutions (LOSS), Lucerne, Switzerland
  - Tryton Hosting & Development
  - Customization
  - Support
- Studying computer science (1 semester to go :)
- Main developer of LOSS Open Source Tryton modules\(^1\)
- Occasional contributer of bugfixes for Tryton
- Tryton package maintainer for Arch Linux AUR

\(^1\)https://bitbucket.org/loss
Why WSGI?

- It is a standardized Python interface (PEP 3333) for communication with web server
- It has been invented to improve portability of web applications
- Potent WSGI server often provide additional features that Tryton lacks (and is not meant to provide), such as:
  - Advanced process management
  - Routing
  - Monitoring
The Patch

• Cédric initially wrote a patch for WSGI support\(^2\) in August 2011, which has been further improved in the meantime
  • Adds factory method for WSGI entry points:
    `trytond.protocols.wsgi.get_jsonrpc_app`
  • Removes multi_server configuration option (removed in 3.4 anyway)
• Still in code review, but “it is no longer a prototype”\(^3\)
• Currently not applicable on trunk without errors, but easy to fix (errors can be ignored)
• Used in production at LOSS since spring 2013

\(^2\)http://codereview.tryton.org/92001/
\(^3\)according to post on tryton-dev mailing list
Introducing uWSGI

Our WSGI Server of choice: uWSGI (but there are many others to choose from)

- Full stack (WSGI) application server
  - Scalability
  - Process monitoring
  - Load balancing
- Targeted at hosting providers
- Versatile, high-performance and reliable (though a bit complex)
- Available on Python Package Index (PYPI)\(^4\)

\(^4\)https://pypi.python.org/pypi/uWSGI
Particularly interesting uWSGI features for Tryton hosting:

- **Emperor**: Run multiple instances of Tryton
- **Cheaper Subsystem**: Adaptive worker process spawning
- **Subscription Service**: Dynamically register new Tryton instances
- **Stats Server**: Get statistics of your Tryton instances
- **Python Tracebacker**: Print current traceback of a running worker process
- **Alarm Subsystem**: Get notified about unhandled exceptions
- **Harakiri**: Kill hanging processes

…and there is probably a lot more I haven’t yet discovered :(
Showtime
uWSGI Configuration

Minimal configuration file for uWSGI (tryton.ini):

```
[uwsgi]
master = True    # start in master mode
http = :8000    # bind to port 8000
workers = 4     # run 4 worker processes
virtualenv = path/to/virtualenv
file = path/to/wsgi.py
env = TRYTOND_CONFIG=path/to/config
```

WSGI startup file (wsgi.py):

```
from trytond.protocols.wsgi import \
    get_jsonrpc_app
application = get_jsonrpc_app()
```
Running uWSGI

Start uWSGI using the config file from the last slide

```bash
$ uwsgi --ini tryton.ini
```

and connect using the client and port 8000

- The configured number of workers are started
- Load balancing is done automatically
- There are a lot more config options for advanced scenarios
Scalable Hosting Architecture

- **nginx**
- **Reverse Proxy**
- **uWSGI**
- **HTTP/uwsgi**
- **Subscription Server**
- **App Server**
- **PostgreSQL**
- **Database**

5 Binary, high performance protocol
Problems, Solutions & Tricks (1)

**Problem**  Tryton Cron does not run

**Solution**  Run Cron as external service using a separate script (following TrytonServer.run method)

**Trick**  With uWSGI you can start the Cron script as an external service together with the WSGI instance
Problems, Solutions & Tricks (2)

**Problem**  Module installation/update leads to inconsistent worker states

**Solution**  An instance reload is required

**Trick**  uWSGI can automatically trigger a reload in code, but this requires patching the server
Discussion

Starting questions:

- Would you like to see official WSGI support in Tryton?
- Are there technical difficulties that prevent WSGI support?
- How do you see the future of Tryton deployment?