Improving Performances with Pydio Booster

CHARLES DU JEU - PYDIO LEAD DEVELOPER - CS3 2017

@CHARLESDUJEU

Web & Mobile access
Desktop sync client
All sort of shares
Storage agnostic (FS, object, remote...)
Server in PHP5.6+
Clients in python/qt and native mobile (swift3, java).
We love PHP...

- Not glamour but really mature
- PHP7 = high performance gains
- PSR's + frameworks
- easily hackable => good for OSS
- Well suited for structured data management (users, roles, rights, etc).

... but it has limitations

- Handling files data and I/O's is a serious matter
- Server-to-clients real time events not adapted
- Long-running operations easily reach request timeout / memory limits
Inherent Bottlenecks => legacy solutions

- Send back info to users / inform sync clients of changes: polling
- Optimize code, make sure to use streams everywhere
- Delegate some tasks to native tools (e.g. long copy, zips), or to PHP CLI.
- Add caching at many layers
- Reduce server load: scale-out / scale-up
2016: time for alternatives?

Micro Services
Direct access to the storage?
Adapt underlying hardware to service
Language more adapted to doing i/o?

Real-time push
Websockets technologies
Existing PHP websocket server was not stable
WS references are written in Nodejs

Ease Deployment
Provide easy to deploy and cross-platform services for the server.
E.g. replace CRONTAB?

Rewrite all? Drop PHP? Which language?
Introducing PydioBooster

- One-click websocket server
- Proxy for uploads / downloads directly from clients to storage
- Built-in scheduler triggering Pydio server master command
- Messaging service (NSQ, by bit.ly)

- Written in Go
- Open source
Why Golang?

- Server-oriented language
- Developed / supported by Google
- Dependency-free binary
- Learning curve: keep it hackable
- Maturity: docker, minio
- Concurrency by-design, pipes concept

Golang + PHP in a micro-service architecture = best of both world!
Proxy protocol overview

**Uploads**
- HTTP Upload
- Authenticate
- Storage location
- Read request & pipe to storage
- Finished
- Server response
- HTTP Response

**Downloads**
- DL requested
- Authenticate
- Storage location
- Read file & send to client
- HTTP Response
- Booster
- PHP Server
- Storage

Similar to X-SendFile
Benefits

- **WebSockets**
  - Drastically reduce load on server
  - Sync client can receive changes on the fly
  - Web UI is much more responsive for the user => good opportunity to send more actions as a background tasks

- **Downloads**: alternative subdomain, not-blocking connection,

- **Uploads**: no move_uploaded_file operation or buffering.

- *Pydio still works perfectly without Booster* (no new mandatory dependency).
Thank you!

Learn more on Pydio 7 on pydio.com