

ZeroMQ

Jeff Abrahamson

Jellybooks

Ce qui est bon :

- Parfois mieux intégrer, donc plus vite
- Souvent facile à déployer

Ce qui est moins bon :

- Couplage étroit
- Passer à l'échelle souvent difficile
- Échec == échec total

Ce qui est bon :

- Compréhensible, facile à développer (par service)
- Déploiement indépendant
- Passer à l'échelle
- Fiabilité (en principe)
- Diversité technologique

Ce qui est moins bon :

- Plus de devops
- Parfois plus lent
- Une complexité différente
- Eventual consistency
- Distribution

ØMQ — C'est top !

“ØMQ is unbelievably cool—if you haven't got a project that needs it, **make one up**”

Jon Gifford

We took a normal TCP socket, injected it with a mix of radioactive isotopes stolen from a secret Soviet atomic research project, bombarded it with 1950-era cosmic rays, and put it into the hands of a drug-addled comic book author with a badly-disguised fetish for bulging muscles clad in spandex. Yes, ZeroMQ sockets are the world-saving superheroes of the networking world.

I/O, non-blocking

Composants dynamiques?

- client/serveur (stable?)
- serveur/serveur
- reconnection

Wire format

- cadrage (framing)
- buffer overflow
- efficacité pour petit et grand message

Quoi faire en attente?

- impossibilité de livraison
- attente de réponse

Quoi faire avec nos files d'attente?

Et s'il y a trop de messages?

Et les messages perdus?

- attendre
- redemander
- couche de fiabilité

Abstraction de couche de transport

- inproc
- ipc
- tcp
- pgm

Routing

- Envoie de messages à multiples clients
- réponse à qui?

Génération d'API pour d'autres langages?

Changement d'architecture

Erreurs de réseau

Un exemple vrai

tyrannosaur
rex.py

Exemple

hello world client/server

Exemple

hello world pub/sub (météo)

Exemple

task pipeline

- répartition de charge
- fair queuing

Exemple

proxy

CurveMQ

Questions?

`https://github.com/JeffAbrahamson/talks/`

`jeff@purple.com`