êtes-vous prêts pour le succès ?

Huh !? Qui ça ?

Steven VAN POECK

autodidacte, 12 ans de PHP, entre autres

stevenvanpoeck

joind.in

http://joind.in/4352

should i stay or should i go* ?

* the clash, 1981

le début

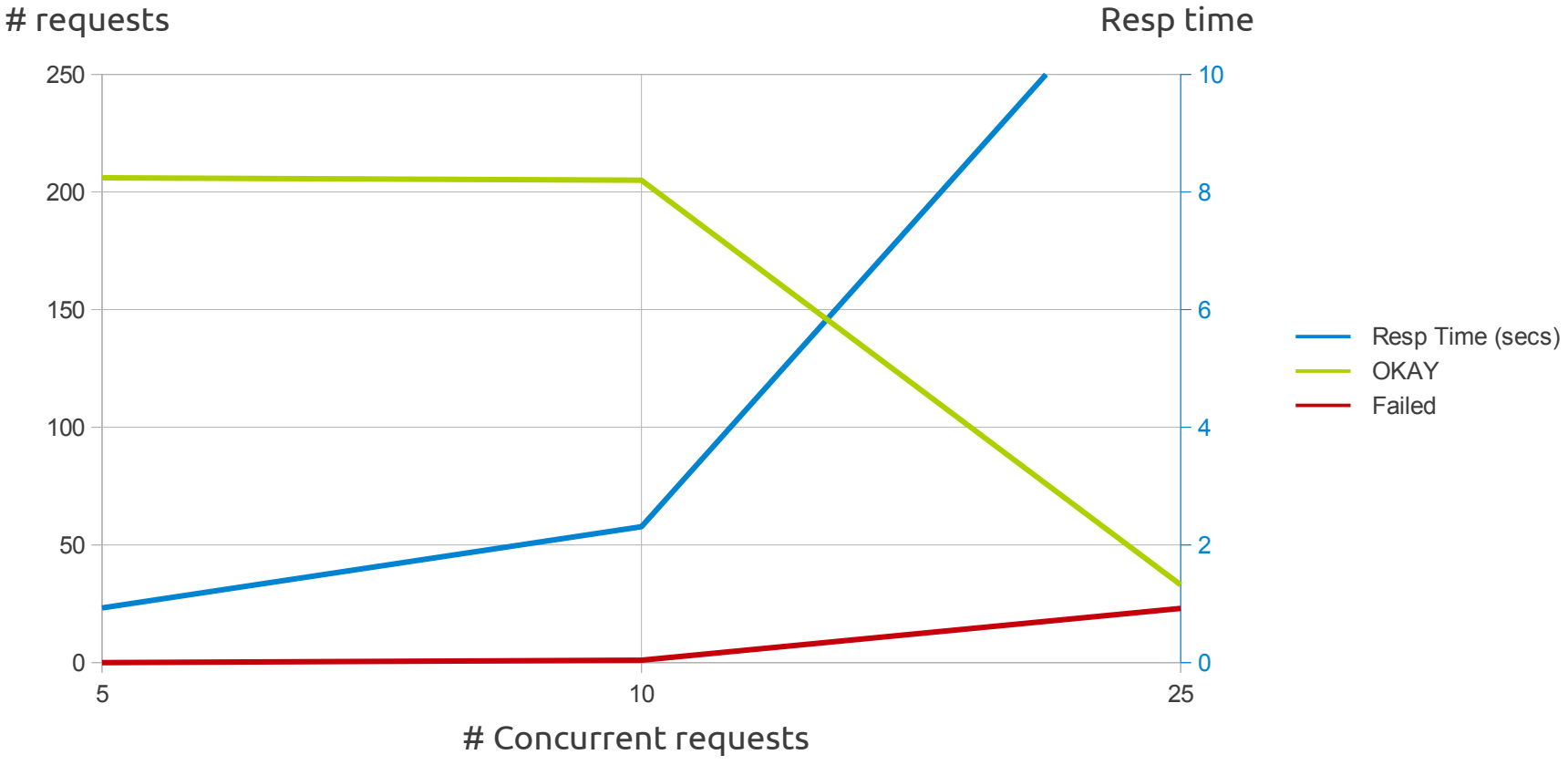
quelques utilisateurs

Apache
2.2.16 + mod
PHP 5.3.3 +
MySQL 5.1.49

Performance :

Disponibilité :

Coût :



Max connexions simultanées : 9

~4 500 utilisateurs

un peu plus d'utilisateurs

Apache
2.2.16 + mod
PHP 5.3.3 +
MySQL 5.1.49

Performance :

Disponibilité :

Coût :

un peu plus d'utilisateurs

Apache
2.2.16 + mod
PHP 5.3.3 +
MySQL 5.1.49

Performance :

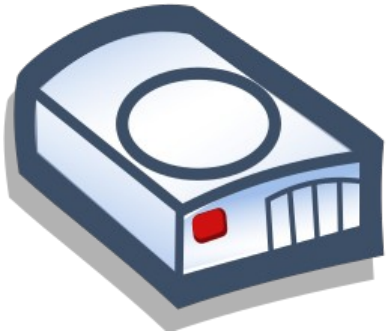
Disponibilité :

Coût :

- solutions
  - utilisez les dernières versions stables des briques logicielles
  - distribution de charges : séparation serveur web / db
  - côté disponibilité : FastCGI pour PHP
  - côté ressources : nginx
  - tuning serveur Web (Nginx)
    - worker_processes → 1 par CPU mais plus de 4 est inutile
    - worker_connections → 2048
    - keepalive_timeout → entre 10 et 20
    - access_logs off → réduction I/O (utilisez Google Analytics pour stats)
    - error_log <LOGFILE> error|crit → réduction I/O (désactivez avec error_log /dev/null crit)

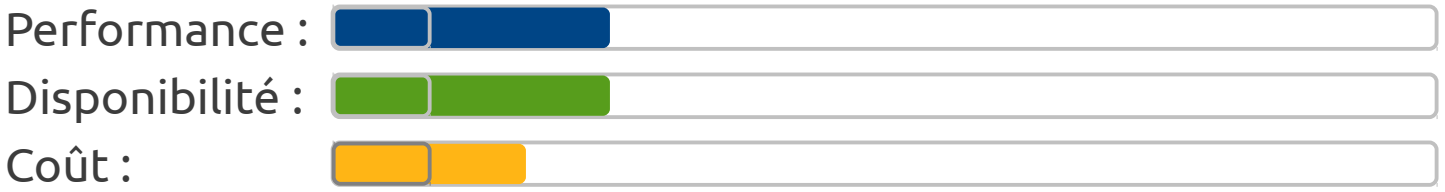
- solutions (suite)
  - tuning PHP
    - cache op-code PHP (APC, eAccelerator, Zend Server)
    - utilisez la dernière version majeure stable !
    - template caching (ex : Smarty)
    - compilation « maison » sans extensions inutilisées
    - always_populate_raw_post_data = Off (utilisez php://input)
    - max_execution_time → le plus petit possible
    - memory_limit → adapté à vos scripts
    - realpath_cache_size → plus si beaucoup de fichiers
    - realpath_cache_ttl → le plus élevé possible
    - enlever Suhosin patch

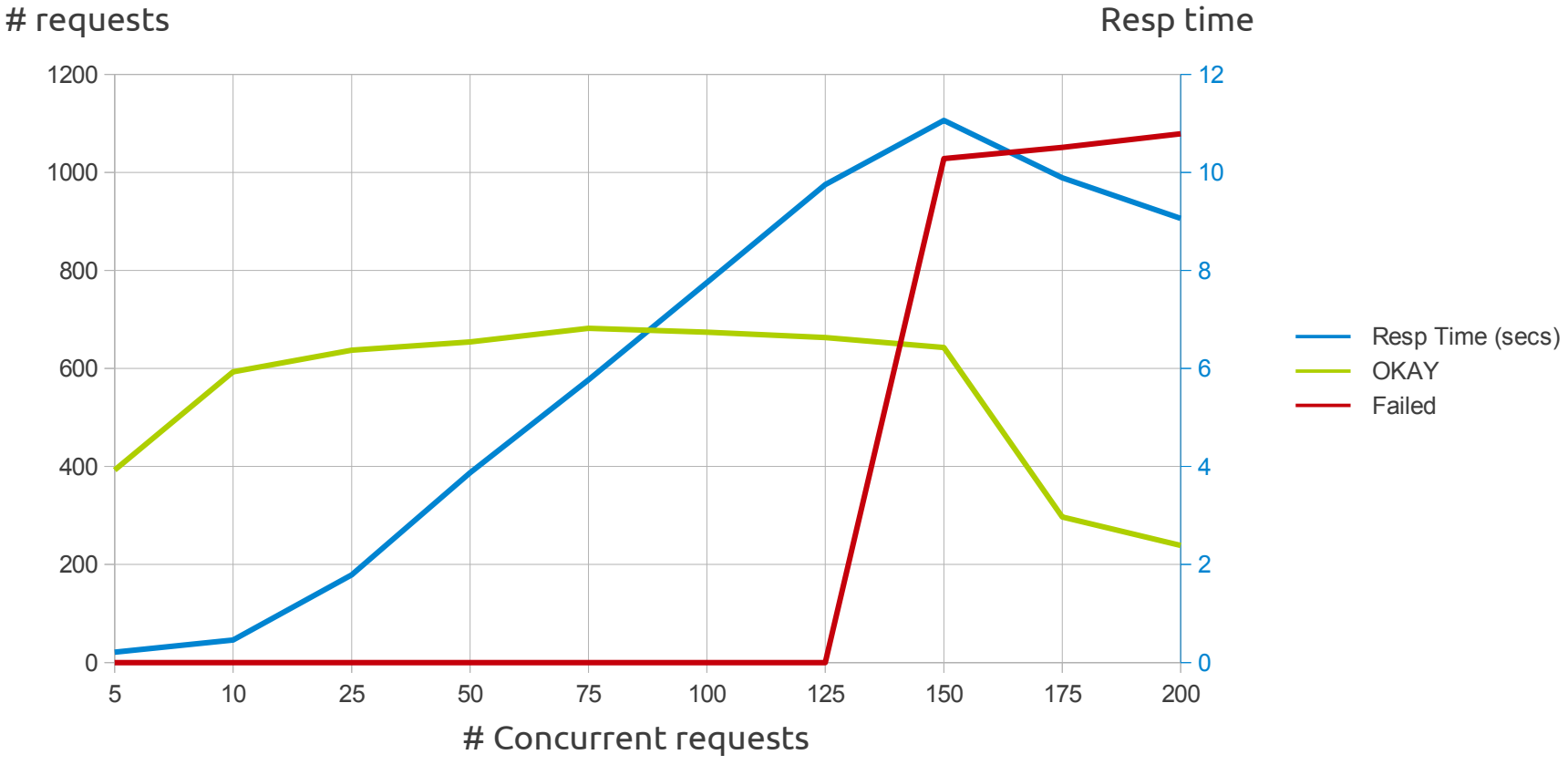
un peu plus d'utilisateurs



MySQL
5.5.16

Nginx 1.0.8 +
FastCGI PHP
5.3.8

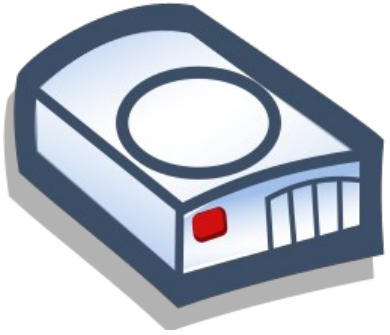




Max connexions simultanées : 27

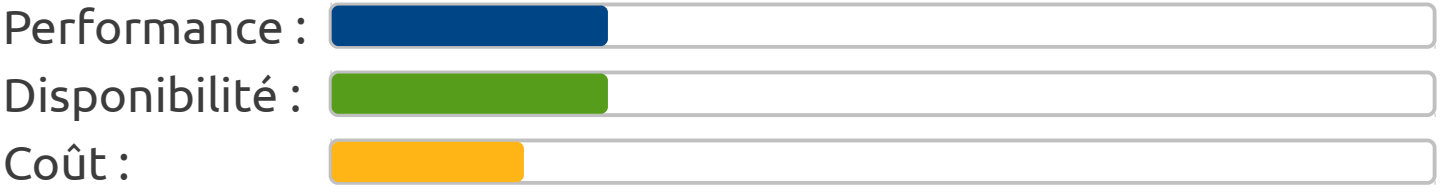
~13 000 utilisateurs

encore plus d'utilisateurs



MySQL
5.5.16

Nginx 1.0.8 +
FastCGI PHP
5.3.8



encore plus d'utilisateurs

Performance :

Disponibilité :

Coût :

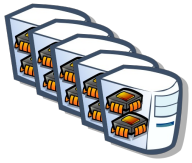
- solutions
  - ajout de RAM sur le serveur MySQL
  - tuning MySQL
    - utilisez InnoDB (rowlock)
    - connection_timeout → le plus bas possible
    - skip_name_resolve = 1 → réduction appels réseau
    - log = /dev/null → réduction I/O
    - slow_query_log = 1 → surveillance
  - cache résultats brutes des requêtes
    - solution ad-hoc dans le code
    - extension AdoDb + memcache
  - optimisation du code
    - exit(ORM);
    - traque et élimination des requêtes dans des boucles

exemple solution ad-hoc dans le code :

```
public function query($sql) {
    static $qCache = array(); // Stockage

    $qSig = md5($sql); // Cle unique pour la requete

    // On retourne le resultat directement
    if (array_key_exists($qSig, $qCache)) {
        return $qCache[$qSig];
    }

    $rs = $this->db->query($sql); // Requete en base

    $qCache[$qSig] = $rs; // Mise en cache

    return $rs; // On retourne le resultat
}
```

encore plus d'utilisateurs

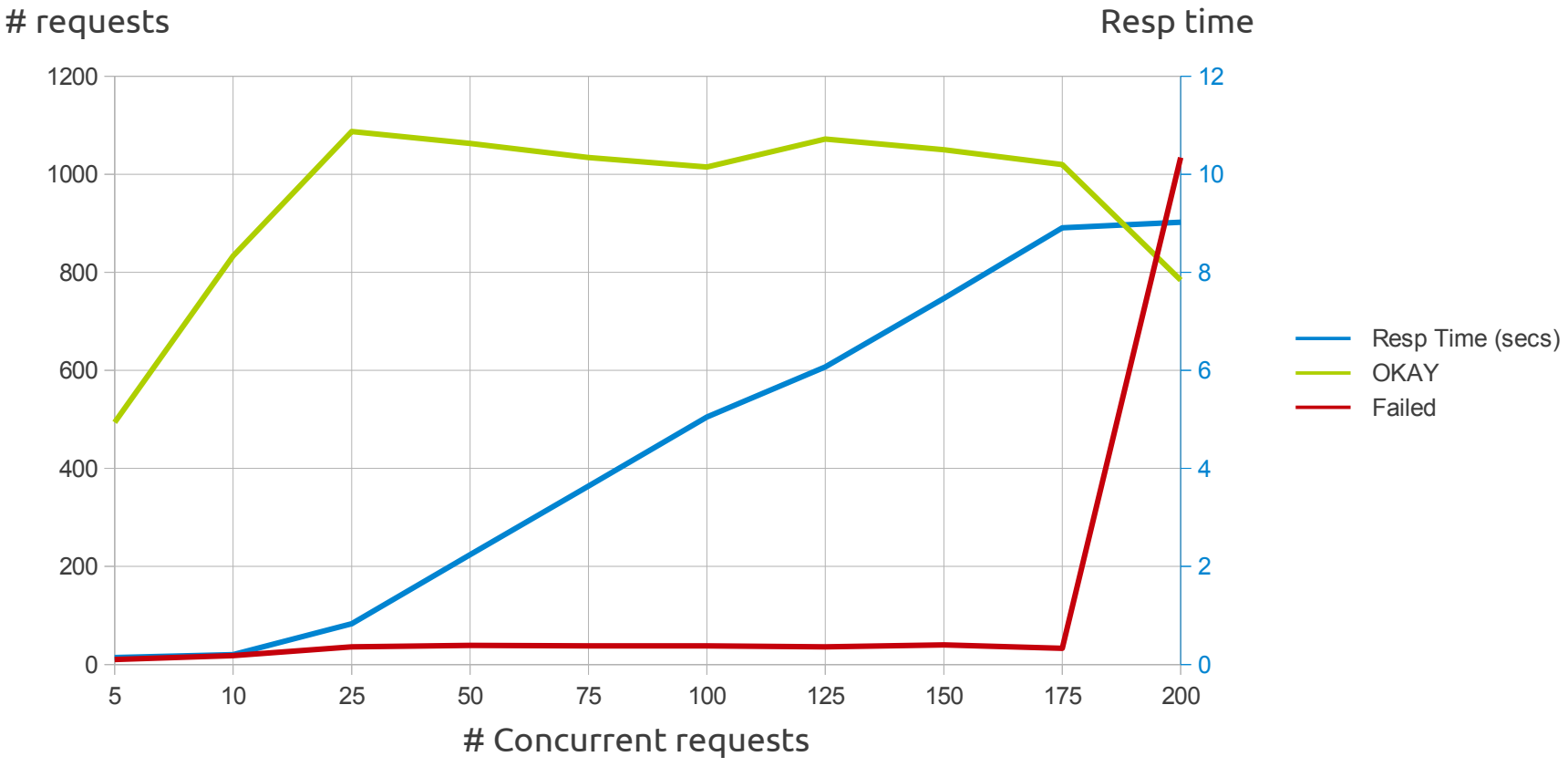
memcache



Performance :

Disponibilité :

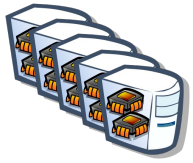
Coût :



Max connexions simultanées : 45

~22 000 utilisateurs

toujours plus d'utilisateurs

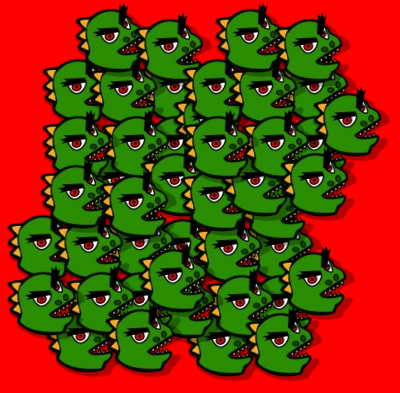


Performance :

Disponibilité :

Coût :

toujours plus d'utilisateurs



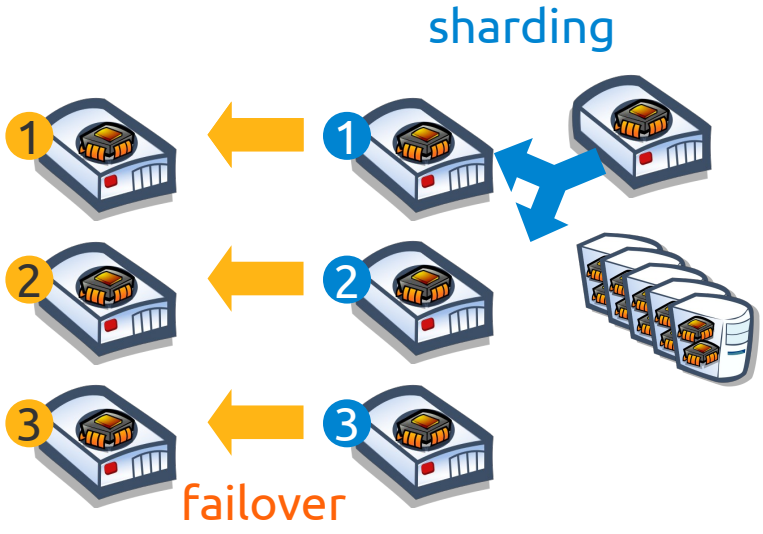
Performance :

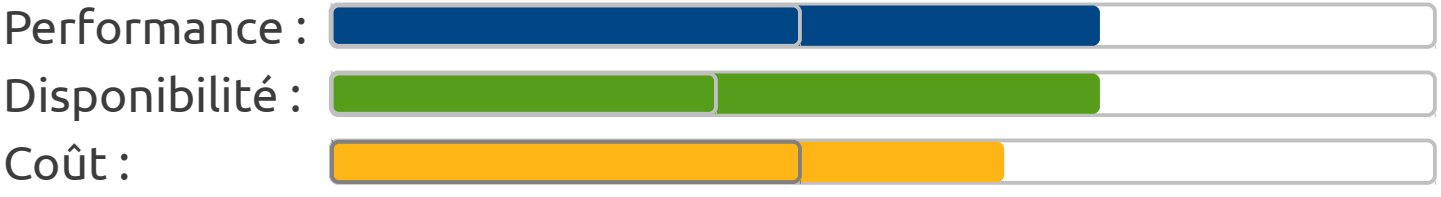
Disponibilité :

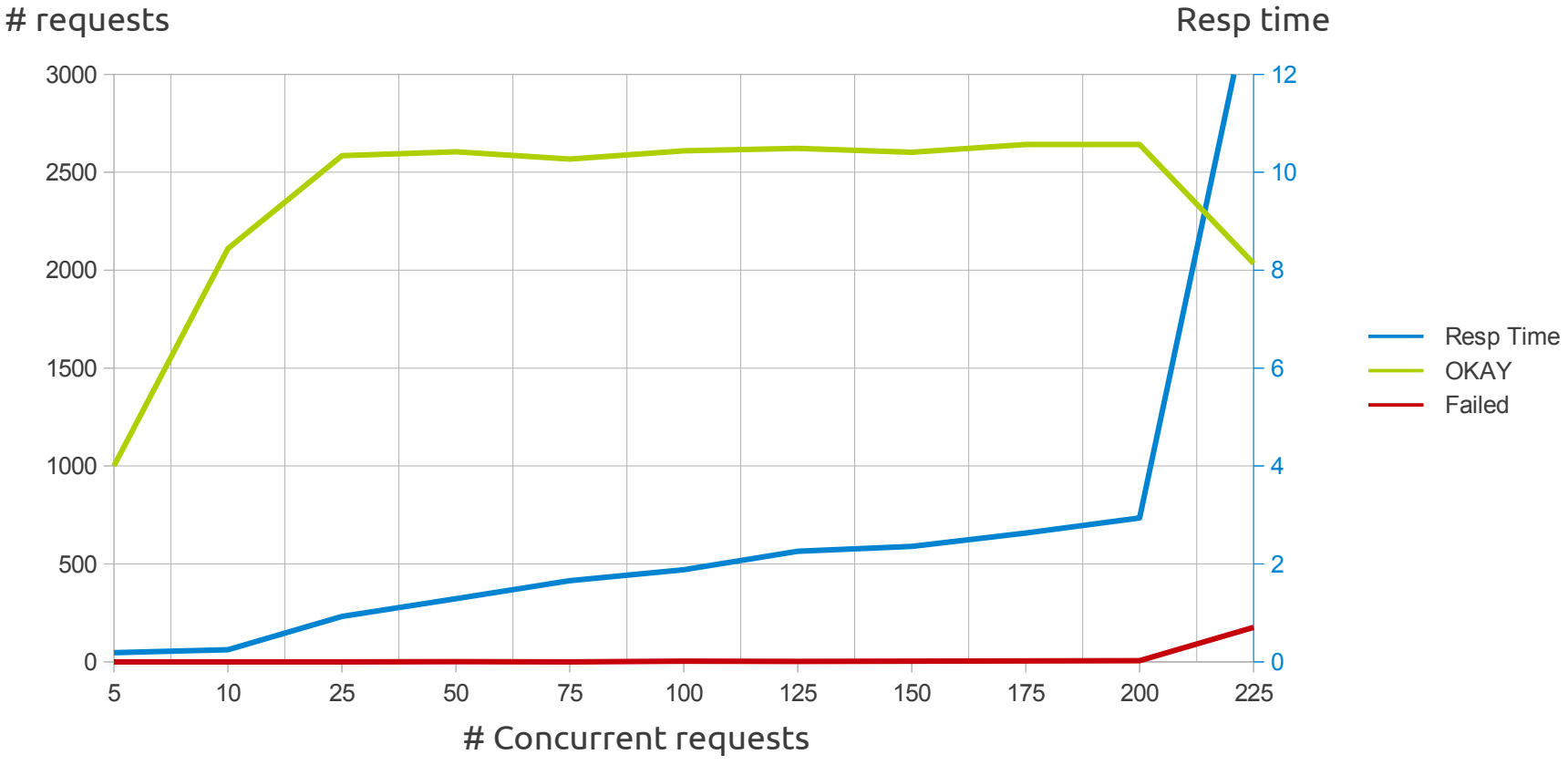
Coût :

- solutions
  - ajout de serveurs frontaux
  - load balancer
  - sharding pour la base de données
  - failover pour les MySQL

toujours plus d'utilisateurs







Max connexions simultanées : 110

~50 000 utilisateurs

le succès est au rendez-vous !

Performance :

Disponibilité :

Coût :

le succès est au rendez-vous !

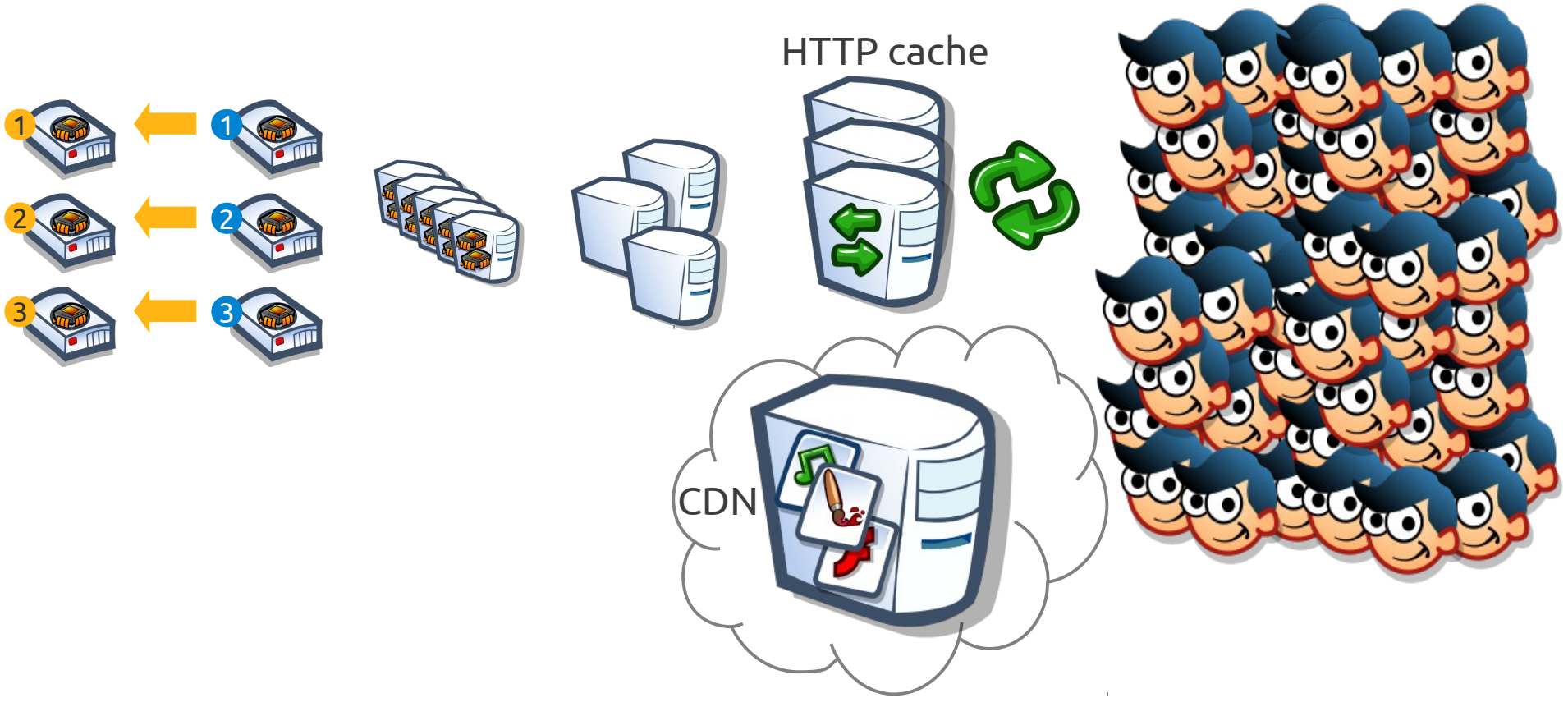
Performance :

Disponibilité :

Coût :

- solutions
  - perfomances & disponibilité
    - HTTP caching (Varnish, Squid, Nginx…)
    - content distribution network (CDN)

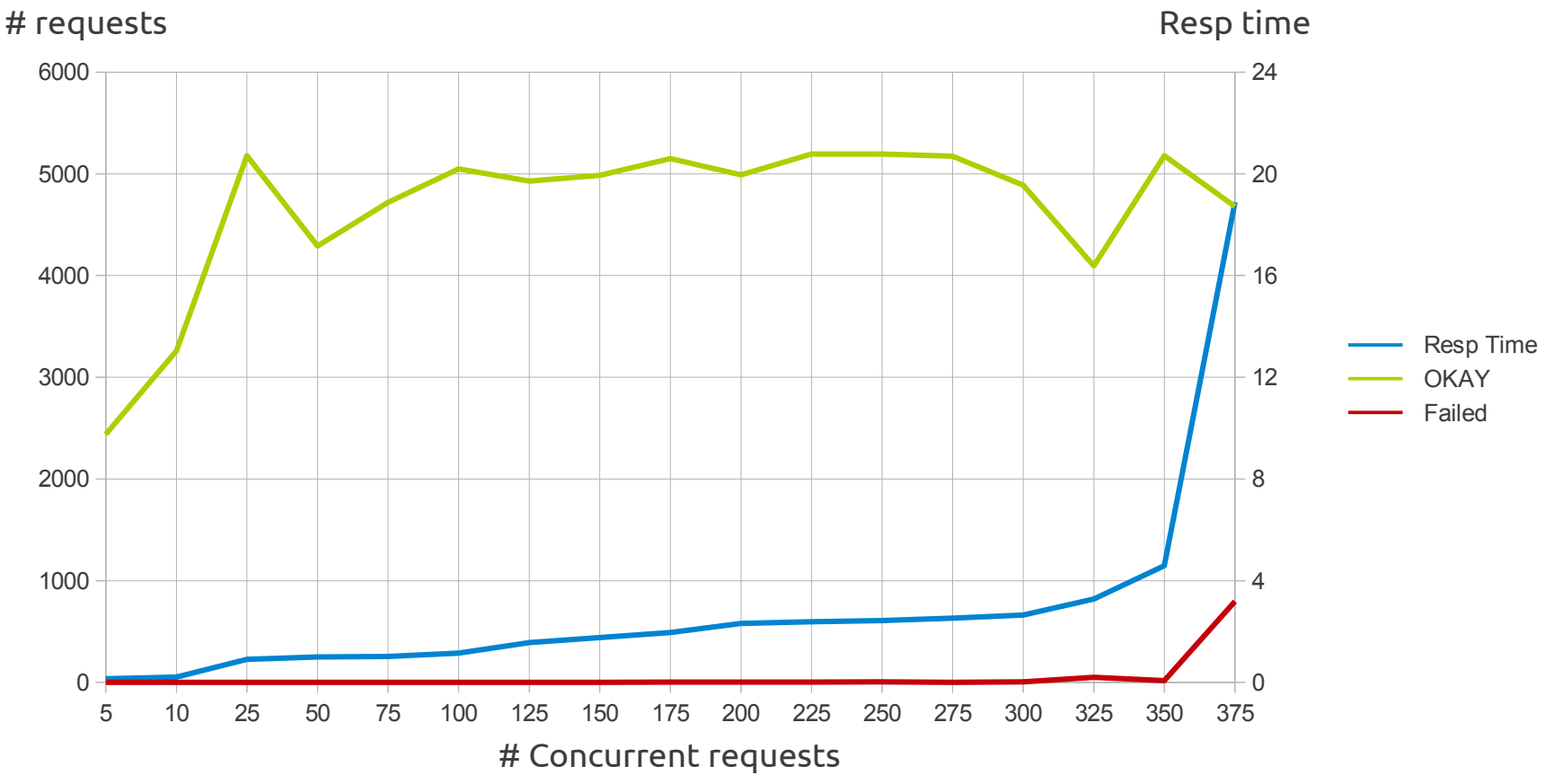
le succès est au rendez-vous !



Performance :

Disponibilité :

Coût :



Max connexions simultanées : 175

~85 000 utilisateurs

mais encore ?

- noSQL
  - MongoDb : http://www.mongodb.org/
  - CouchDb : http://couchdb.apache.org/
  - CouchBase : http://www.couchbase.org/
  - ...
- ze nuage (Amazon, Azure, Eucalyptus, Google apps...)

questions ?

- Références:
  - Dernières versions briques logicielles
    - Debian : http://dotdeb.org
    - PHP : http://www.php.net
  - Tuning PHP / Nginx :
    - http://blog.martinfjordvald.com/2011/04/optimizing-nginx-for-high-traffic-loads/
    - http://php.net/manual/en/ini.core.php
    - http://talks.php.net/show/perf_tunning
    - http://phplens.com/lens/adodb/docs-adodb.htm#memcache
  - Load balancing : http://en.wikipedia.org/wiki/Load_balancing_(computing)
  - Optimisation MySQL : http://www.slideshare.net/ligaya/dpc-tutorial
  - Sharding : http://en.wikipedia.org/wiki/Shard_(database_architecture)
  - Failover : http://en.wikipedia.org/wiki/Failover
  - Memcached : http://memcached.org/
  - CDN : http://en.wikipedia.org/wiki/Content_delivery_network
  - HTTP cache :
    - http://en.wikipedia.org/wiki/HTTP_accelerator
    - https://www.varnish-cache.org/
    - http://nginx.net/
    - http://www.squid-cache.org/