

Tsung GIS

Rodolphe Quiédeville

Tests de charge des systèmes d'informations géographiques

22 mai 2014

Rodolphe Quiédeville

- ▶ 15 ans d'admin/sys tendance devops
- ▶ 18 ans de libertés dans l'IT
- ▶ venu aux SIG par freegis
- ▶ contributeur OpenStreetMap
- ▶ contributeur Tsung
- ▶ freelance

Principe du test de performance

Jouer un scénario métier sur un système applicatif, et cela sous différentes charges afin de mesurer les temps de réponses, de détecter d'éventuels problèmes, déterminer le point de rupture

...

Principe du test de performance

Jouer un scénario métier sur un système applicatif, et cela sous différentes charges afin de mesurer les temps de réponses, de détecter d'éventuels problèmes, déterminer le point de rupture

...

- ▶ test de charge

Principe du test de performance

Jouer un scénario métier sur un système applicatif, et cela sous différentes charges afin de mesurer les temps de réponses, de détecter d'éventuels problèmes, déterminer le point de rupture

...

- ▶ test de charge
- ▶ test de performance

Principe du test de performance

Jouer un scénario métier sur un système applicatif, et cela sous différentes charges afin de mesurer les temps de réponses, de détecter d'éventuels problèmes, déterminer le point de rupture

...

- ▶ test de charge
- ▶ test de performance
- ▶ test de capacité, montée en charge

Principe du test de performance

Jouer un scénario métier sur un système applicatif, et cela sous différentes charges afin de mesurer les temps de réponses, de détecter d'éventuels problèmes, déterminer le point de rupture

...

- ▶ test de charge
- ▶ test de performance
- ▶ test de capacité, montée en charge
- ▶ test aux limites

Principe du test de performance

Jouer un scénario métier sur un système applicatif, et cela sous différentes charges afin de mesurer les temps de réponses, de détecter d'éventuels problèmes, déterminer le point de rupture

...

- ▶ test de charge
- ▶ test de performance
- ▶ test de capacité, montée en charge
- ▶ test aux limites
- ▶ test de vieillissement

Tsung

Pourquoi Tsung

- ▶ multi protocoles
- ▶ scalable (validé à 10 Millions d'utilisateurs)
- ▶ extensible par plugins
- ▶ scenarios XML avec validation XSL
- ▶ scenario dynamique avec chargement de modules
- ▶ 13 ans de code, un projet robuste
- ▶ publié sous GNU GPL V2
- ▶ ouvert à contribution
- ▶ modèle Stochastic avec distribution de Poisson

Utilisateurs de Tsung

- ▶ Total
- ▶ CMA-CGM
- ▶ Ministère des finances
- ▶ Dailymotion
- ▶ Skyrock
- ▶ LibertySurf
- ▶ Grid'5000

Skeleton de scénario

Example

```
<?xml version="1.0"?>
<!DOCTYPE tsung SYSTEM "/usr/share/tsung/tsung-1.0.dtd">
<tsung loglevel="notice" version="1.0">
  <clients />
  <servers />
  <options />
  <load />

  <sessions>
    <session>
      <transaction name="foobar"/>
      <transaction name="barfoo"/>
    </session>
  </sessions>
</tsung>
```

Si tsung permet de réaliser des tests de performance, il n'a pas de dimension géographique.

Si tsung permet de réaliser des tests de performance, il n'a pas de dimension géographique.

Tsung GIS apporte cette dimension géographique à Tsung

Si tsung permet de réaliser des tests de performance, il n'a pas de dimension géographique.

Tsung GIS apporte cette dimension géographique à Tsung

- ▶ modules de calculs

Si tsung permet de réaliser des tests de performance, il n'a pas de dimension géographique.

Tsung GIS apporte cette dimension géographique à Tsung

- ▶ modules de calculs
- ▶ des squelettes de scenario

Une carte à base de tuiles dans une page web engendre un nombre élevé de requête

Exemple

```
/tile/france/6/33/21.png  
/tile/france/6/31/23.png  
/tile/france/6/34/23.png  
/tile/france/6/33/24.png  
/tile/france/6/30/21.png  
/tile/france/6/32/22.png  
/tile/france/6/30/24.png  
/tile/france/7/63/45.png  
/tile/france/7/62/46.png  
/tile/france/7/65/46.png  
/tile/france/7/66/45.png  
/tile/france/7/64/44.png  
/tile/france/8/126/90.png  
/tile/france/8/129/90.png  
/tile/france/8/127/92.png
```

Example

```
<request>
  <http url="/tile/france/6/33/21.png">
</request>
<request>
  <http url="/tile/france/6/31/23.png">
</request>
<request>
  <http url="/tile/france/6/34/23.png">
</request>
<request>
  <http url="/tile/france/6/33/24.png">
</request>
<request>
  <http url="/tile/france/6/30/21.png">
</request>
<request>
  <http url="/tile/france/6/32/22.png">
</request>
```

- ▶ scénarios longs et fastidieux

Problématiques

- ▶ scénarios longs et fastidieux
- ▶ scénarios sans approche métier

Problématiques

- ▶ scénarios longs et fastidieux
- ▶ scénarios sans approche métier
- ▶ scénarios difficilement maintenables

Problématiques

- ▶ scénarios longs et fastidieux
- ▶ scénarios sans approche métier
- ▶ scénarios difficilement maintenables
- ▶ scénarios non réutilisables

Problématiques

- ▶ scénarios longs et fastidieux
- ▶ scénarios sans approche métier
- ▶ scénarios difficilement maintenables
- ▶ scénarios non réutilisables

Il est possible d'améliorer la situation en utilisant un module.

Qu'est-ce qui caractérise un SIG d'un SI ?

Qu'est-ce qui caractérise un SIG d'un SI ?

- ▶ les actions sont limitées

Qu'est-ce qui caractérise un SIG d'un SI ?

- ▶ les actions sont limitées
- ▶ appels HTTP sont prédictifs

Mouvements

Les actions utilisateurs sont au nombre de 6.

4 mouvements de base, un par direction.

- ▶ déplacement vers le nord
- ▶ déplacement vers le sud
- ▶ déplacement vers l'ouest
- ▶ déplacement vers l'est

Plus 2 actions de zoom

- ▶ zoom_more
- ▶ zoom_less

Fonctions spéciales

Ces 6 actions de bases sont complétées avec 1 action d'initialisation.

Et 3 actions aléatoires pour augmenter l'interactivité du scénario.

- ▶ `move_first` : initialisation de la carte
- ▶ `action_random` : action aléatoire
- ▶ `move_random` : déplacement aléatoire
- ▶ `zoom_random` : zoom aléatoire

Les différents modules qui composent Tsung GIS

Des modules utilitaires

- ▶ map
- ▶ randomcoord
- ▶ slippymap
- ▶ reproj

Les modules de protocoles

- ▶ tms
- ▶ wfs
- ▶ wkb
- ▶ wms
- ▶ xyz

Génération de données géographiques aléatoires.

- ▶ `rcoord_array`
- ▶ `url`

rcoord_array/1 retourne un tableau de coordonnées **lat,lon**

Example

```
10> randomcoord:rcoord_array({Pid, Dynvars}).  
["70.205974", "33.121879"]
```

rcoord_array/1 retourne un tableau de coordonnées **lat,lon**

Example

```
10> randomcoord:rcoord_array({Pid, Dynvars}).  
["70.205974", "33.121879"]
```

rcoord_array/1 peut être configuré pour renvoyer des données au sein d'une bbox donnée.

url/1 retourne une url de base pour générer des données géographiques aléatoires

url/1

```
4> randomcoord:url({Pid, Dynvars}).  
"lat=88.969257&lon=47.267150"
```

slippymap rassemble des fonctions de conversions essentiellement utilisées par les modules plus orientés fonctionnels.

slippymap rassemble des fonctions de conversions essentiellement utilisées par les modules plus orientés fonctionnels.

- ▶ deg2num/3
- ▶ num2deg/3
- ▶ num2bbox/3
- ▶ tile2lat/2
- ▶ tile2lon/2

deg2num/3 conversion d'une coordonnées lon,lat en un numéro de tuile contenant cette coordonnées à un niveau de zoom donné.

deg2num/3

```
10> slippymap:deg2num(48.84098,2.58741,18) .  
{132956,90202}
```

reproj permet de faire les conversions entre systèmes de coordonnées.

- ▶ tmstowms/1
- ▶ reproj/3

reproj utilise proj4erl qui est un bindings erlang de la librairie PROJ.4

wkb est plus orienté Postgis que web, il regroupe les fonctions de manipulations d'objet encodé en Well Known Binary

- ▶ wkb_point/1
- ▶ wkb_linestring/1

Module encore au stage alpha

xyz est constitué de toutes les actions de base relatives à une carte implémentée dans un contexte web avec par exemple OpenLayers ou Leaflet.

xyz est constitué de toutes les actions de base relatives à une carte implémentée dans un contexte web avec par exemple OpenLayers ou Leaflet.

- ▶ move_first
- ▶ move_north
- ▶ move_south
- ▶ move_east
- ▶ move_west
- ▶ move_random
- ▶ zoom_more
- ▶ zoom_less
- ▶ zoom_random
- ▶ action_random

Initialisation

L'initialisation d'un scenario consiste à définir la taille de la carte, le point central et les différentes couches.

L'initialisation d'un scenario consiste à définir la taille de la carte, le point central et les différentes couches.

dimensions

```
<setdynvars sourcetype="value" value="960">  
  <var name="map_width" />  
</setdynvars>
```

```
<setdynvars sourcetype="value" value="640">  
  <var name="map_height" />  
</setdynvars>
```

L'initialisation d'un scenario consiste à définir la taille de la carte, le point central et les différentes couches.

dimensions

```
<setdynvars sourcetype="value" value="960">  
  <var name="map_width" />  
</setdynvars>
```

```
<setdynvars sourcetype="value" value="640">  
  <var name="map_height" />  
</setdynvars>
```

centre

```
<setdynvars sourcetype="value" value="8/109/121">  
  <var name="first_url" />  
</setdynvars>
```

La dernière étape est l'utilisation de **move_first** qui correspond à l'arrivée sur le site.

Example

```
<setdynvars sourcetype="erlang" callback="xyz:move_first"
  <var name="list_url" />
</setdynvars>
```

```
<transaction name="move_first">
  <foreach name="url" in="list_url">
    <request subst="true">
      <http url="/%_%url%.png" />
    </request>
  </foreach>
</transaction>
```

Mouvement de base

Une fois la carte initialisée on peut utiliser les toutes les autres actions un déplacement vers l'est se fera en utilisant ...

Mouvement de base

Une fois la carte initialisée on peut utiliser les toutes les autres actions un déplacement vers l'est se fera en utilisant **xyz:move_east**

Example

```
<setdynvars callback="xyz:move_east" sourcetype="erlang"
  <var name="list_url" />
</setdynvars>

<foreach name="element" in="list_url">
  <request subst="true">
    <http url="/%%_element%%.png" />
  </request>
</foreach>
```

ENTITY

Comme XML et Tsung le permettent on va factoriser un peu le scenario pour le rendre plus lisible.

On va définir en début de scénario des entités xml pour chaque action de base

ENTITY

```
<?xml version="1.0"?>
<!DOCTYPE tsung SYSTEM "/usr/share/tsung/tsung-1.0.dtd"
[
<!ENTITY move_first SYSTEM "xyz_move_first.xml">
<!ENTITY move_east SYSTEM "xyz_move_east.xml">
<!ENTITY move_north SYSTEM "xyz_move_north.xml">
<!ENTITY move_south SYSTEM "xyz_move_south.xml">
<!ENTITY move_west SYSTEM "xyz_move_west.xml">
<!ENTITY zoom_more SYSTEM "xyz_zoom_more.xml">
<!ENTITY zoom_less SYSTEM "xyz_zoom_less.xml">
]
>
<tsung loglevel="notice" version="1.0">
```

Scénario initial

Pour rappel le scénario initial était :

Example

```
<transaction>
  <request>
    <http url="/tile/france/6/33/21.png">
  </request>
  <request>
    <http url="/tile/france/6/31/23.png">
  </request>
  <request>
    <http url="/tile/france/6/34/23.png">
  </request>
  <request>
    <http url="/tile/france/6/33/24.png">
  </request>
  <request>
    <http url="/tile/france/6/30/21.png">
  </request>
  <request>
```

Scénario initial

Va dorénavant pouvoir s'écrire

Example

```
<transaction>
```

```
  &zoom_more;
```

```
  &zoom_more;
```

```
  &move_east;
```

```
  &move_west;
```

```
</transaction>
```

Un scénario complet devient un ensemble de règle métier

Example

```
<session name="sess1" probability="100">
  &init_map;

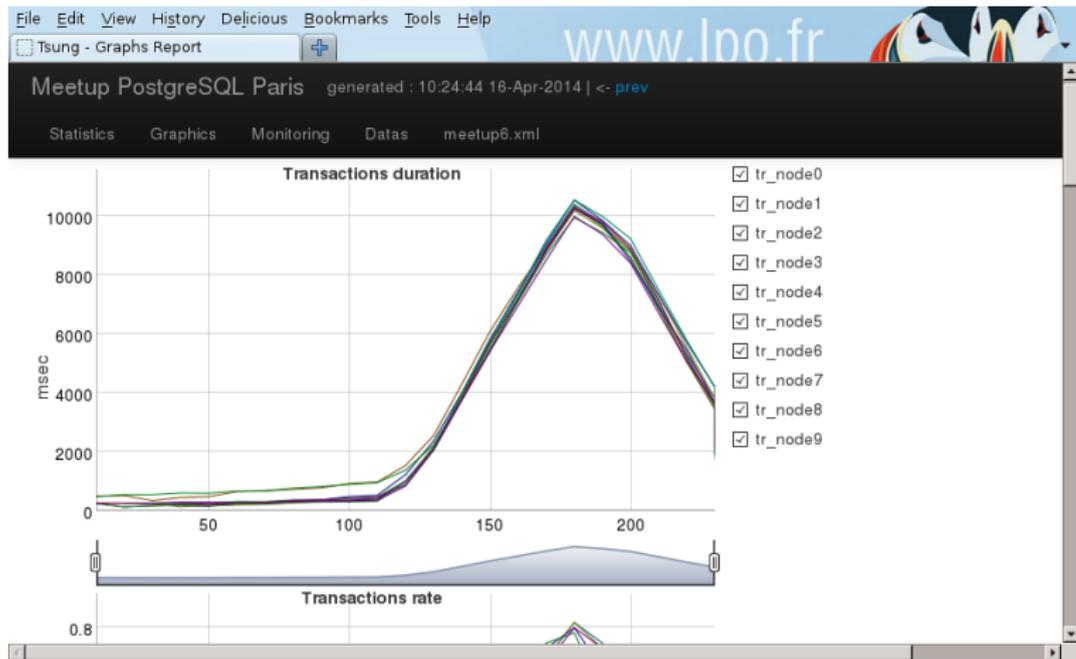
  <transaction name="start">
    &move_first;
  </transaction>

  <transaction name="nav">
    &move_random;

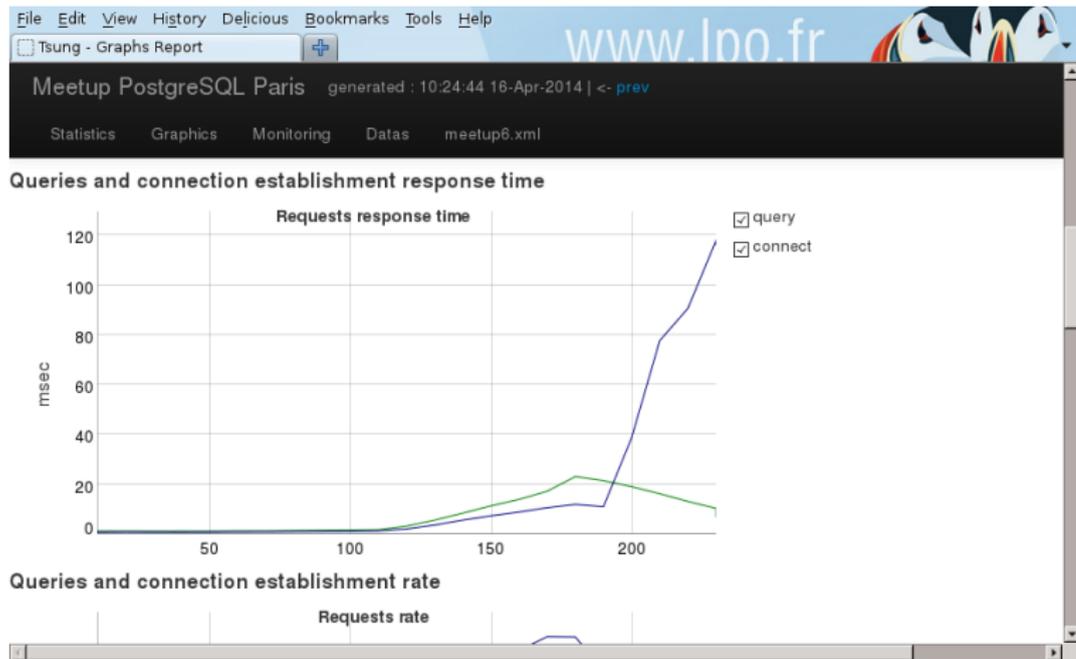
    &zoom_more;

    &move_north;
  </transaction>
</session>
```

Analyse des résultats



Analyse des résultats



Analyse des résultats

File Edit View History Delicious Bookmarks Tools Help

Tsung - Stats Report

www.lpo.fr

127.0.0.1:8080/20140416-1005/report.html

DuckDuckGo

Meetup PostgreSQL Paris Statistics Graphics Monitoring Datas meetup6.xml generated : 10:24:44 16-Apr-2014 | << prev

Main Statistics

Name	highest 10sec mean	lowest 10sec mean	Highest Rate	Mean	Count
query	0.29 sec	0.774 msec	2445.3 / sec	14.32 msec	276643
session	1mn 2sec	0.29 sec	10 / sec	8.33 sec	913

Tsung Transactions Statistics

Name	highest 10sec mean	lowest 10sec mean	Highest Rate	Mean	Count
node0	12.25 sec	0.25 sec	1.4 / sec	5.93 sec	103
node1	12.57 sec	0.24 sec	1.1 / sec	6.59 sec	86
node2	12.56 sec	0.26 sec	1.6 / sec	6.77 sec	95
node3	13.08 sec	0.25 sec	1.1 / sec	6.71 sec	79
node4	12.59 sec	0.26 sec	1.2 / sec	5.72 sec	104
node5	13.10 sec	0.26 sec	1.4 / sec	6.33 sec	98

Scénario initial

Ce qui n'a pas été abordé

- ▶ support wms
- ▶ support tms
- ▶ support wfs
- ▶ la gestion des couches
- ▶ carte composite multi-protocoles

Code sous licence GPL ouvert à contribution.

- ▶ **Docs**, <http://tsung-gis.readthedocs.org/en/latest/index.html/>
- ▶ **Code**, <https://github.com/rodo/tsung-gis/>
- ▶ **CI**, <http://jenkins.quiedeville.org/view/TsungGIS/>

Code sous licence GPL ouvert à contribution.

- ▶ **Docs**, <http://tsung-gis.readthedocs.org/en/latest/index.html/>
- ▶ **Code**, <https://github.com/rodo/tsung-gis/>
- ▶ **CI**, <http://jenkins.quiedeville.org/view/TsungGIS/>

Si vous ne savez pas coder vous pouvez me payer pour le faire à votre place !

Questions ?

Rodolphe Quiédeville

rodolphe@quiedeville.org

<http://blog.rodolphe.quiedeville.org/>

Document publié sous Licence Creative Commons BY-SA 2.0