

Tsung API 測試流程

Tsung APIs

- 要執行 APIs 即為 requests，皆撰寫在 <sessions> 的 <request> 內
- API要加掛的參數為多數時，以&在參數間作連結
 - 如：/standard-platform/api/v1/updateUser?fb_id=23321&fb_token=123123

Tsung xml 注意事項

- <clients> 內的client，其host name不能是IP address，所以利用編輯 /etc/hosts，加入 host_ip_address host_name 的方式 (ex：54.168.154.216 tsung0)，再編輯 client host =“host_name” 來套用。
- 測試的負載主要在 <load></load> 及 <sessions></sessions> 2個區塊內撰寫
- <load>主要定義使用者部份；如每秒產生多少users，持續多久等設定
- <sessions>主要定義requests動作，2個requests間會插入時間間隔 thinktime
- requests 內的動態變數 dynamic variables 使用時，利用 <setdynvars> </setdynvars> 來加入，且 <request> 後要加入 subst="true" (<request subst="true">)，才能使用 %%_變數名%% 方式加以套用。
- 因為 tsung will end when all started users have finished their session。所以要測試 concurrent users 的架構，大多利用 for 迴圈讓每位 user 所要執行的動作持續一段不短的時間，進而達成 concurrent users 的目的。

Tsung xml 撰寫

- xml 文件 header 的部份，不用更動，copy&paste 即可。
- client 及 server 乃根據測試架構來撰寫，所以若測試架構不變，即只要最初設定完，就不用更動。
- 其他 monitor 及 options 2 區塊為 optional，可設可不設
- 在測試架構不變下，主要就是利用修改<load> 及 <sessions> 2 區塊來改變測試時的請求。

```
- <load>
  <arrivalphase phase="1" duration="30" unit="minute">
    <users interarrival="0.002" unit="second"></users>
  </arrivalphase>
</load>
```

- 每秒增加 500 ($1/0.002=500$) user，持續 30 分，總共有 $(500 \times 60) \times 30 = 900000$ users

```
- <load duration="20" unit="minute">
  <arrivalphase phase="1" duration="10" unit="minute">
    <users maxnumber="12000" arrivalrate="200" unit="second"></users>
  </arrivalphase>
</load>
```

- 每秒增加 200 users 的頻率，持續 10 分，最多累積達到 12000 users

```
- <setdynvars sourcetype="random_number" start="1"
  end="1000000000000000000000000">
  <var name="fb_rnd" />
</setdynvars>

  <request> <http url="/standard-platform/api/v1/activityInfo?
fb_id=1&activity_id=1" method="GET" version="1.1"></http> </
request>
  <thinktime value="2" random="true"></thinktime>
  <request subst="true"> <http url="/standard-platform/api/v1/
updateUser?fb_id=%_fb_rnd%&fb_token=%_fb_rnd%" method="POST"
version="1.1" ></http> </request>
```

- 其中 thinktime 是隨機的，但整體的平均值為 2s
- 產生 fb_rnd 動態變數，其值為 1~1000000000000000000000000 的隨機值

- request(API) 內要用 %_fb_rnd% 方式來使用此動態隨機變數值
- <request 後一定要加入 subst="true"，不然動態變數仍不能使用

requests 計數

- 利用 <load> 內的設定可計數出會新增的 users
 - interarrival : 1/ (interarrival) 取得 每 unit值 (如：秒)會增加的users
 - 配合 duration及其unit 取得最終會新增 users 總數
 - 然後再配合 <sessions> 內 <request> 數，得出會執行的 requests
- arrivalrate : (arrivalrate) / (unit)，每 unit 值增加 (arrivalrate) users
 - 大多會再設定 maxnumber 值與其配合
 - 配合 duration 及其 unit 取得最終可能會新增 users 總數，因為此總數值可能會受限 maxnumber 值
 - 如：以每秒增加 200 users 的頻率持續 10 分，則總數為 $200 \times 60 \times 10 = 120000$ ，若 maxnumber 為 100000，則最後總數為 100000 而非 120000
 - 可在 <sessions> 內加入 <for> 迴圈使其增加的 users 數變成 concurrent users 的形式，就是利用 for 迴圈使每個 users 不停地執行 requests
 - 可利用少數的 users 產生大量的 requests
 - 由於執行的 requests 量可能大增遠超過原 duration 時間，可在 <load> 後加入時間限制 (如：<load duration="20" unit="minute">)，這樣最多執行 20 分就會結束。

流程

- command : tsung start (tsung -f ~/.tsung/tsung.xml start)

- -f 後接要執行的xml script
- report command :
 - cd ~/.tsung/log/xxxxxxxx-xxxx (xxxxxxxx-xxxx 時間格式：西元年月日-時分)
 - /usr/lib/tsung/bin/tsung_stats.pl