

FinecoVita

Stress Test sull'applicazione CITCall

Milano

Hacking Team S.r.l.	http://www.hackingteam.it
<i>Via della Moscova, 13 20121 MILANO (MI) - Italy</i>	info@hackingteam.it
<i>Tel. +39.02.29060603</i>	<i>Fax +39.02.63118946</i>

STORIA DEL DOCUMENTO		
Versione	Data	Modifiche Effettuate
1.0	16/03/2005	Prima Stesura

INFORMAZIONI	
Data di Emissione	16/03/2005
Versione	1.0
Tipologia Documento	Allegato Tecnico
Numero di Protocollo	//
Numero Pagine	7
Numero Allegati	0
Redatto da	Federico Guerrini
Approvato da	Gianluca Vadruccio

INDICE

1	Obiettivo.....	4
2	Ambiente di riferimento	4
3	Analisi dei requisiti	4
3.1	Vincoli.....	5
4	Descrizione della metodologia	5
4.1	Definizioni.....	5
4.2	Sollecitazione del sistema.....	6
4.3	Analisi del sistema.....	6
4.4	Elaborazione dei risultati	6
5	Descrizione degli impatti	7
6	Planning	7

1 Obiettivo

FinecoVita ha richiesto una analisi delle prestazioni (*Stress Test*) sull'applicazione CITCall, allo scopo di determinare il massimo carico che il sistema è in grado di gestire ed, eventualmente, di fornire indicazioni sugli aspetti più critici dal punto di vista delle performance.

Questo documento descrive la soluzione proposta da Hacking Team, sia dal punto di vista tecnico, sia dal punto di vista dell'effort previsto.

2 Ambiente di riferimento

Sulla base delle indicazioni fornite dal Committente, il sistema target è costituito dai seguenti elementi architetturali.

- **Interfacciamento con centralino telefonico:** l'attivazione del browser e la richiesta dell'URL per l'accesso alle funzionalità dell'applicazione a fronte di ogni telefonata inbound sono gestiti in modo automatico, mediante opportuno plug-in per MSIE, sviluppato da terzi.
- **Front end:** CITCall espone agli operatori del call center una interfaccia web.
- **Application server:** la business logic di CITCall è sviluppata con tecnologia Java. Come servlet container è utilizzato Tomcat versione 4.1.
- **Data layer:** è costituito da un database Oracle attestato su un cluster di due nodi, che garantisce l'alta disponibilità del servizio.

3 Analisi dei requisiti

Il committente richiede una analisi delle performance del sistema CITCall simulando i seguenti livelli di carico:

- 15 utenti impegnati in conversazione per il 50% del tempo disponibile;
- 15 utenti impegnati in conversazione per il 75% del tempo disponibile;
- 15 utenti impegnati in conversazione per il 100% del tempo disponibile;
- 20 utenti impegnati in conversazione per il 50% del tempo disponibile;
- 20 utenti impegnati in conversazione per il 75% del tempo disponibile;
- 20 utenti impegnati in conversazione per il 100% del tempo disponibile;
- 25 utenti impegnati in conversazione per il 75% del tempo disponibile;
- 25 utenti impegnati in conversazione per il 100% del tempo disponibile;
- 30 utenti impegnati in conversazione per il 75% del tempo disponibile;
- 30 utenti impegnati in conversazione per il 100% del tempo disponibile;

- 60 utenti impegnati in conversazione per il 75% del tempo disponibile;
- 60 utenti impegnati in conversazione per il 100% del tempo disponibile.

3.1 Vincoli

La presente attività di testing prevede che l'interazione con il sistema CITCall avvenga mediante l'interfaccia HTTP che essa espone al browser dell'utente finale. La generazione del carico avverrà mediante opportune sequenze di richieste HTTP verso l'application server su cui CITCall è attestato, o verso altri server HTTP su cui risiedono eventuali funzionalità esterne. In particolare, il carico generato non potrà simulare in modo completo interazioni verso altri sistemi (centralino telefonico, sistemi host).

4 Descrizione della metodologia

Questo paragrafo descrive la modalità di esecuzione del test funzionale.

4.1 Definizioni

La metodologia adottata da Hacking Team si fonda sui seguenti concetti.

- **Use Case:** è una operazione che può essere effettuata da un utente mediante l'applicazione. Tale operazione può comprendere un numero arbitrario di interazioni client-server, può avere natura transazionale e può implicare la creazione di una "sessione" che in ogni istante descrive lo stato dell'interazione fra utente ed applicazione.
- **Virtual User:** è la sorgente di una successione di richieste verso l'applicazione che nel loro complesso realizzano uno Use Case.
- **Curva di carico:** descrive le caratteristiche dinamiche del carico generato (transitori iniziali e finali per passare dalle condizioni di carico nullo alle condizioni di carico previsto, durata della fase di regime, etc). La curva di carico è definita quando sono noti, istante per istante, il numero di virtual user attivi e gli use case che essi eseguono. Quando i virtual user vengono attivati ad intervalli di tempo costanti, la curva di carico può essere descritta tramite un unico parametro scalare, il *ramp-up time*, ovvero il tempo impiegato per attivare *tutti* i virtual user previsti dal test case.
- **Test Case:** è un livello di carico a cui si sottopone il sistema. Un test case è definito da:
 - un insieme di use case;
 - numero di virtual user che implementano ogni test case;
 - curva di carico;

L'intera attività di stress test può dunque essere definita dall'insieme di test case per i quali si sono misurate le performance del sistema.

4.2 Sollecitazione del sistema

Lo stress test viene svolto interagendo con l'applicazione nella stessa modalità prevista per gli utenti finali, cioè via web. Allo scopo di testare il sistema in modo esaustivo, esso viene sollecitato con numerose sequenze di richieste HTTP che simulano le più frequenti modalità di interazione (use case) da parte degli utenti finali.

Si utilizzano più test case, ciascuno caratterizzato dallo stesso insieme di use case, ma da un numero progressivamente crescente di virtual user che li eseguono.

4.3 Analisi del sistema

Il comportamento del sistema viene analizzato utilizzando un approccio ibrido black box/white box. Nell'analisi di tipo black box il sistema (infrastrutture di rete, server, componenti software) è visto come un componente monolitico che espone una interfaccia HTTP. L'analisi si basa quindi sui codici di risposta (HTTP 200 "OK", HTTP 500 "Internal Server Error", ecc.) e sui tempi di risposta ottenuti sottomettendo al sistema sequenze di richieste HTTP.

Nell'analisi di tipo white box il sistema viene visto come un insieme di diverse componenti interagenti, ognuna delle quali può essere analizzata separatamente dalle altre. Per svolgere questo tipo di analisi, durante il test vengono monitorate le prestazioni ed il livello di occupazione delle risorse del front-end web e del back-end (livello di occupazione delle risorse di CPU e memoria RAM).

4.4 Elaborazione dei risultati

I risultati dello stress test sono ottenuti dall'analisi di:

- **Tempi di risposta.** I tempi di risposta dell'applicazione alle richieste HTTP vengono loggati durante l'esecuzione del test. Poiché il tempo di risposta ad una singola richiesta è fortemente dipendente dalle condizioni istantanee del sistema, una corretta interpretazione di tali dati richiede una loro elaborazione: il tempo di risposta viene calcolato per mezzo di operazioni di media (tempi medi per completare una data operatività, tempi medi per particolari richieste HTTP, etc.).
- **Codici di risposta HTTP.** Eventuali anomalie vengono individuate e segnalate in base ai codici di errore (HTTP 404 "Not Found", HTTP 500 "Internal Server Error", ecc.) nelle risposte del sistema.

Utilizzo delle risorse hardware e software. Vengono riportati i dati sui livelli di occupazione delle risorse di processore/memoria sui web server e sul database server.

5 Descrizione degli impatti

La preparazione e l'esecuzione del test comportano la necessità di interagire con l'applicazione secondo le stesse modalità degli utenti. Gli impatti sul sistema target sono pertanto i seguenti.

- **UtENZE di test:** sarà necessario creare un insieme di utenze di test, le cui caratteristiche saranno definite durante le prime fasi dell'attività.
- **Database di test:** poiché l'interazione con l'applicazione può comportare la modifica della base dati, sarà necessario predisporre una istanza di test, che verrà utilizzata sia per la preparazione sia per l'esecuzione dell'attività.
- **Ripristino del database di test:** l'esecuzione del test può richiedere che la base dati sia riportata ad uno stato noto definito durante la fase di preparazione. Sarà pertanto necessario prevedere le opportune operazioni di backup/restore.
- **Isolamento:** se il database server ospita anche dati di altre applicazioni, sarà necessario definire opportune strategie per evitare che i risultati del test siano influenzati da condizioni di carico anomale (più o meno elevate) causate da fattori esterni a CITCall.

6 Planning

Di seguito, l'elenco delle attività di cui ai paragrafi precedenti.

- Information gathering
- Definizione degli use case
- Testing use case
- Svolgimento del test
- Analisi dei risultati
- Documentazione