



WALTy: A User Behavior Tailored Tool for Evaluating Web Application Performance

Giancarlo Ruffo, ruffo@di.unito.it

with

R.Schifanella, M. Sereno (University of Turin)

and

R. Politi (CSP)



Introduction

- Popular Web sites need to be carefully scaled up and out, especially if they provide customers service;
- A set of metrics must be measured (e.g., served requests per seconds, throughput, Round trip time, Errors, etc.) to evaluate overall performance;



Introduction (cont.)

- To detect system bottlenecks, resource usage must be monitored during load and stress test:
 - CPU and RAM utilization;
 - I/O Disks access;
 - Network traffic;
- SNMP Agents should record data to be post processed and analyzed

August 30th, 2004

IEEE NCA04 - Cambridge, MA, USA



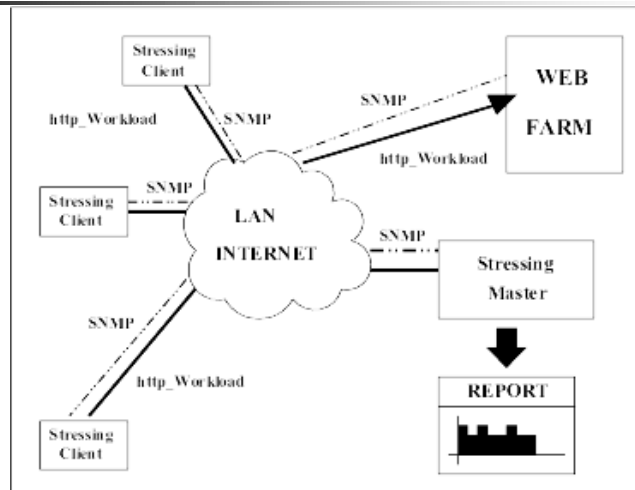
Overview of the talk

- Stressing framework and related work:
- CBMG (Customer Behavior Model Graph);
- CBMG-based request stream;
- Description of WALTy;
- Conclusions and Future Work.

August 30th, 2004

IEEE NCA04 - Cambridge, MA, USA

Stressing framework



August 30th, 2004


IEEE NCA04 - Cambridge, MA, USA

Stressing framework (cont.)

- Synthetic workload is based on a given *workload characterization*:
 - Trace based
 - File list based
 - Analytical distribution based

August 30th, 2004

IEEE NCA04 - Cambridge, MA, USA



Characterization of the Request Stream

- **Trace based:** the characteristics of the Web workload is based on pre-recorded trace logs;
- **File list:** the tools provide a list of Web objects with their access frequencies. During workload generation, the next object to be retrieved is chosen on the basis of its access frequency.
- **Analytical distribution driven:** the Web workload characteristics are specified by means of mathematical distributions.

August 30th, 2004

IEEE NCA04 - Cambridge, MA, USA



Tools and Approaches

- SpecWeb99 (www.spec.org)
- Surge (Barford, Crovella, 98)
- OpenSta (open source - www.opensta.org)
- TPC-W (Transaction Processing Performance Council – www.tpc.org/tpcw)
- WebStone (Mindcraft)
- Load Runner (Mercury Interactive)
- **HTTPERF** (HP LAB)

August 30th, 2004

IEEE NCA04 - Cambridge, MA, USA

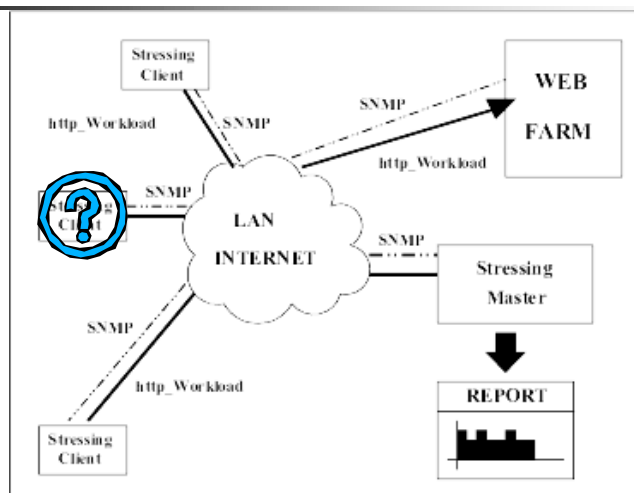
Comparison of Alternatives

- **Surge** (analytical distribution driven) generates representative workload but only for static web services.
- **SpecWeb99** (file list) is largely available, but it is designed to evaluate performance on different architectures for comparison purposes.
- **TPC-W** (trace based) produces benchmark for e-business and e-commerce sites.
- **Load Runner** (trace based) is a very expensive commercial tool.
- **Comparison paper** [*Andreolini, Cardellini and Colajanni, Performance 2002 Tutorial Lectures*]

August 30th, 2004

IEEE NCA04 - Cambridge, MA, USA

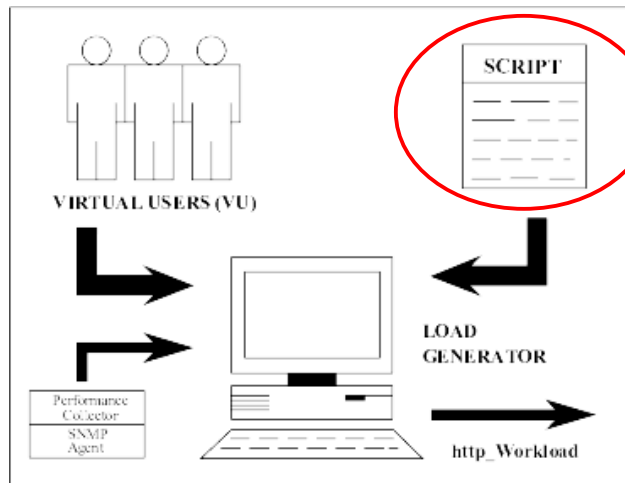
Workload generation



August 30th, 2004

IEEE NCA04 - Cambridge, MA, USA

Stressing client



August 30th, 2004

IEEE NCA04 - Cambridge, MA, USA

Test based stressing client

- The script describes the behavior of each Virtual User;
- It emulates a browser, which sends the server a sequence of GET (or POST) *requests* for pages and embedded objects;
- Between successive requests, the virtual user can be configured for *waiting* a given interval of time (client think time);
- HTTPERF syntax used.

August 30th, 2004

IEEE NCA04 - Cambridge, MA, USA



Artificial sessions

- Trace based stressing clients replicate a (set of) *artificial session(s)* – or navigational patterns.
- These sessions can be:
 - ~~Randomly generated;~~ Not realistic
 - ~~Manually generated;~~ Subjective
 - ~~Extracted from log files.~~ Not scalable

August 30th, 2004

IEEE NCA04 - Cambridge, MA, USA




Profile based sessions

- An acceptable solution: each Virtual User is given a profile. It behaves following its profile.
- Log files are used to extract (few) user profiles, not (many) user sessions:
 - More realistic;
 - Not subjective;
 - Scalable.

August 30th, 2004

IEEE NCA04 - Cambridge, MA, USA



Customer Behavior Model Graph (CBMG)

- We used CBMG (Menascè, Almeida, et al. - 1999);
- Workload characterization based on user profiling by means of Markov Chains;
- Profiles are clusters of “similar” navigational patterns extracted from log files.

August 30th, 2004

IEEE NCA04 - Cambridge, MA, USA



CBMG

- Each web service function is a **state**. A set of static and dynamic pages belong to a state.
- CBMG is a pair (P, Z) of $N \times N$ matrices, where N is the number of states.
- $P = [p_{ij}]$ contains the transition probabilities between states;
- $Z = [z_{ij}]$ represents the average server think times between successive requests (i.e., average time elapsed since the server completed a request until it receives the next request).

August 30th, 2004

IEEE NCA04 - Cambridge, MA, USA

CBMG: an example

CBMG states for a generic Web service with a restricted access area

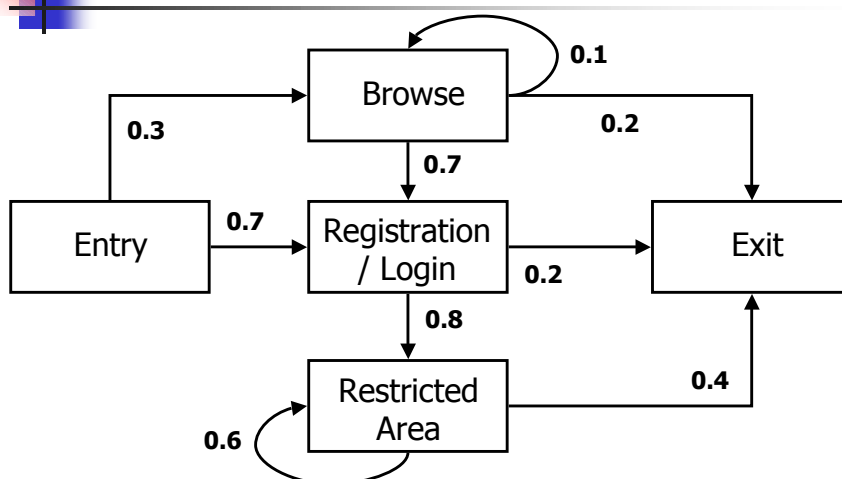
N = 5

State	Function
0	Entry
1	Browse
2	Registration
3	Restricted Area
4	Exit

August 30th, 2004

IEEE NCA04 - Cambridge, MA, USA

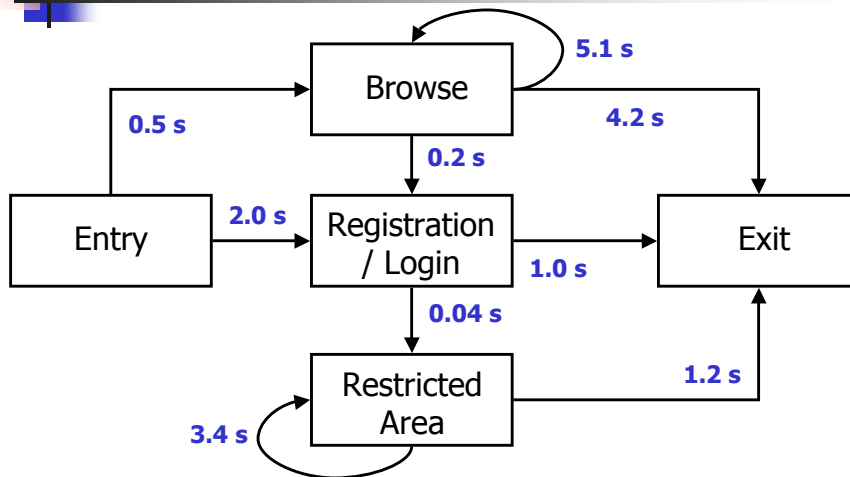
A CBMG profile: graphical representation of matrix P



August 30th, 2004

IEEE NCA04 - Cambridge, MA, USA

A CBMG profile: graphical representation of matrix Z

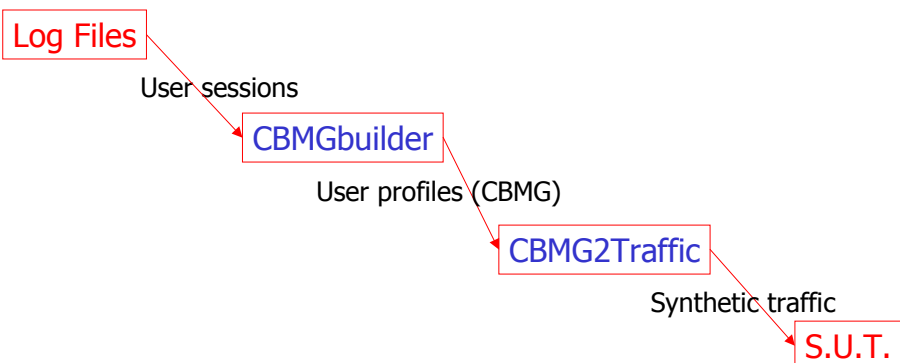


August 30th, 2004

IEEE NCA04 - Cambridge, MA, USA

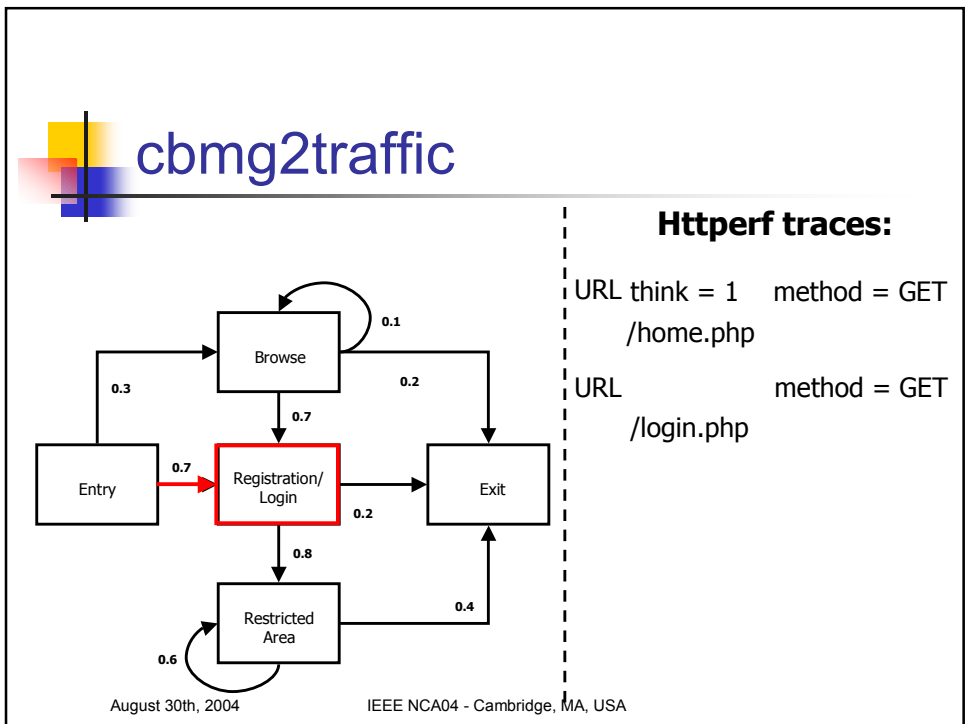
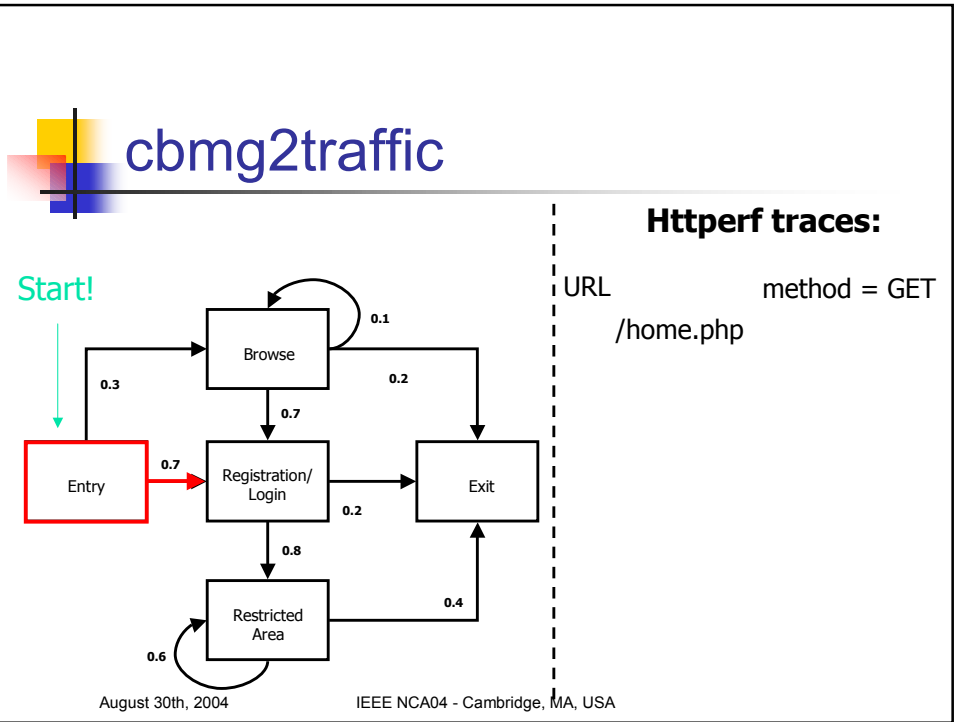
CBMG based request streams

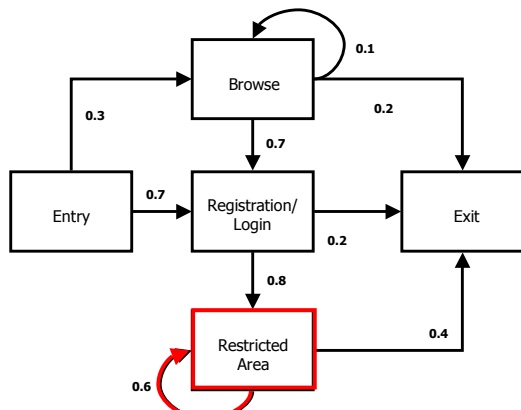
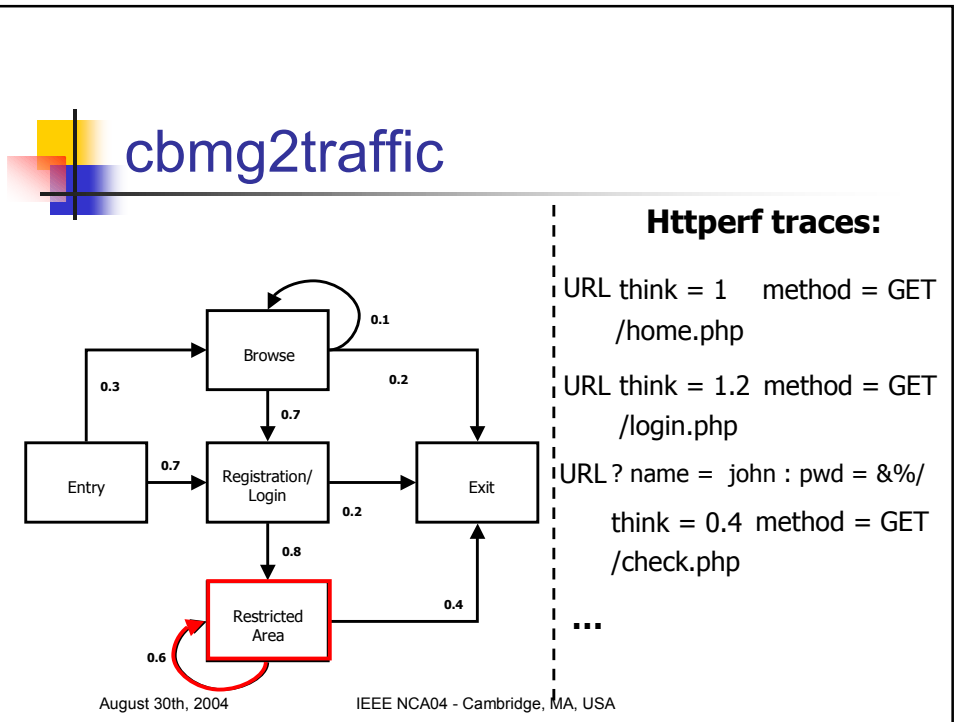
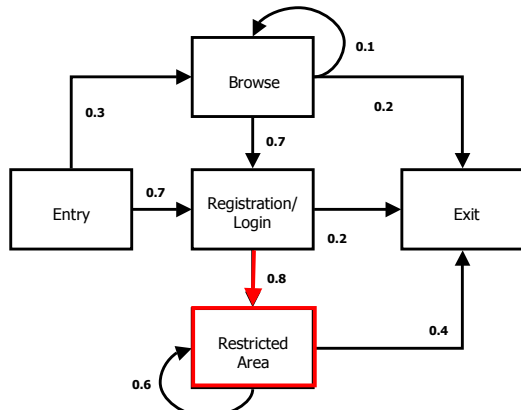
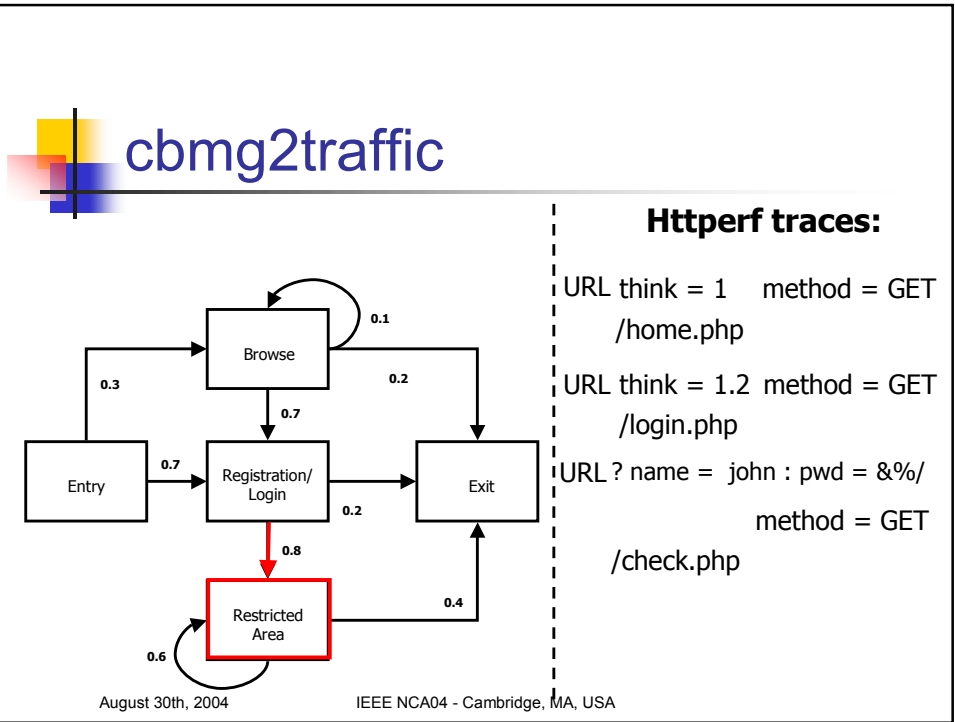
- Idea: a CBMG (reflecting an user profile) can be used to generate an emulated session.



August 30th, 2004

IEEE NCA04 - Cambridge, MA, USA







WALTy

- A Java tool released under General Purpose Licence.
- It implements both modules (CBMGBuilder and CBMG2Traffic)
- Scalable: the analyst can decide the number of virtual users, # sessions, representativeness of each session...

August 30th, 2004

IEEE NCA04 - Cambridge, MA, USA



Conclusions

- CBMGs useful for characterizing user profiles as well as for generating request streams to a Web System;
- CBMG can be used to characterize and test different kinds of Web Services (not only E-commerce sites);
- WALTy can be asked by e-mail to Rossano Schifanella (schifane@di.unito.it)

August 30th, 2004

IEEE NCA04 - Cambridge, MA, USA