New Challenges for Performance Engineers:

Analyzing On-line Services

Varsha Apte Associate Prof., IIT-Bombay. (Part of the work was jointly done with: Mohit Gupta, TCS)

Outline

Motivation

- Product performance engineering techniques are ineffective when designing services
- Performance of on-line services
 - Challenges
 - Existing approaches
 - Ongoing work at IIT-B

Context

- A global shift is happening towards a "service economy", often enabled by the Internet
 - Many technology providers are shifting focus towards services or systems integration
- Pressure towards accelerating time-to-market of services



Elements of Performance Engineering - Product



e.g. Internet routers, switches, Web-servers, Web backend software, application servers, DB servers

Elements of Performance Engineering - Product

Early in Product Cycle				Later in Product Cycle	
Performance Requirements	Performance Budgeting	Performance Modeling	Performance Test and Measurement (feed into the model)	Performance Prediction at future usage volumes, using models+ measurement	
Performation modeling in modeling in details of p	nce nvolves nternal roduct		Measu	rement is is tightly	
 Detailed r can be used choosing p design 	nodels d in product	Tight coupling wi	couple with an and wi	ed and verified nalytical models th developers	
		development teal	m		

Enter: Services

E-commerce web-sites

- Banking
- Shopping
- Web-based e-mail service
- Technical support service

Service Architecture-





Service provider needs to integrate disparate systems for providing a composite, seamless service

Service Performance Engineering – Assuring good user-perceived performance



Challenges in Performance Engineering of Web-services

- Internal details of products may not be known
 - Internals of off-the-shelf products are protected as IP
 - Custom software developers may be
 - " Geographically "far away"
 - "Not very eager to share details
 - Not much may be known about legacy systems

No control over external systems

Elements of Performance Engineering - Service



So What's the Point?

 Advanced queueing models for performance analysis not possible/not useful

Focus has to shift instead to the means available and the needed information

"Means" and "Needs"

Means:

- Measurement analysis of black/"gray"
 boxes
- Simple models for high-level architecture
- Detailed models of well-known technologies (e.g. Web-servers, TCP/IP, SSL))
- Needs
 - Capacity analysis, sizing analysis, bottleneck analysis





Analysis Approaches

- We'll discuss these three "means"
 Models of well-known technologies, in this case, Web-server
 - 2. Measurement-based analysis
 - 3 End-to-end modeling of systems

1. Web-server Models

Various queueing models proposed

- Reeser et al [1] first proposed a detailed model which captured all aspects of a Web server which serves static files
- Mainkar [6] as well as Reeser et al [2]
 extended this model to represent dynamic
 Web-servers

Web Server Queueing Model



Web Server Queueing Model

Original queueing model captures details of system I/O queues and the rate at which they are "drained"

- Shows that web-server throughput depends on whether users access it mainly over dial-up or over a LAN (lower when dial-up)
 - Has deep impact on how results based on performance measurement on a LAN are extrapolated to a dial-up scenario

Dynamic Web Model

Response Time vs Hit Rate - LAN test, 512 thread limit 50 40 An orall 30 Model - Test 20 10 50 100 150 200 250 300 350 Hit rate per hour

•Dynamic server model validated with *tests*

•Validation shows good results

 Two layered model (requests queue at HTTP threads, HTTP threads queue at CPU)
 Solved using iteration

Response Time vs Arrival Rate - dialup test, 512 thread limit



2. Performance Measurement

For web-based services,

Off-the-shelf load generator and performance monitoring products

Performance measurement may have to be of a "black box" (internals not known)



 Commercial load generator tools focus on ease-of-use for "system test group"
 There is a need for better tools targeted towards performance analysts

Testing team focus: Check if service meets requirements

Measurement tools

Load generator



Performance analyst's focus: Take everything into account and produce a performance/capacity analysis, sizing plan, as well as architectural improvements



Using Existing Tools for Capacity Analysis:



Should be automated

Performance

Monitoring

Tools

Ongoing work at IIT-B (nascent stage)

Load generator software

> Tool that intelligently co-ordinates working of load generator and gathering of performance statistics at the server (e.g. rules for detecting steady state, for range of load over which measurement is to be done)

User-oriented results
System performance measures

Tool does intelligent analysis of data collected by performance monitors that were run during the measurement period.

In short, the tool's aim is:



Performance Measurement – New Challenges



 "Box" internals are not known
 Apart from capacity analysis, diagnosis of performance problems may be required
 Analyst can work only with measures collected by operating system

Performance Measurement – New Challenges

Different approach required for such analysis

- Signature-based analysis is one such approach, described in [3]
 - Signatures are characteristic, repeatable behaviors of server software
 - Approach involves deducing the performance problem by observing measurement signatures

Signatures example



Figure 1.1 - Throughout is, Thus for Apolication with Futat Memory Leak



Figure 1.2 - Memory Usage vs. Time for Application with Futul Manury Loak

•Two charts form a "signature" for a fatal memory leak

3. Performance Modeling

Estimation of end-to-end delay requires queueing network models

- Only simple models need be used, because of the unpredictability of service components
- End-to-end delay/capacity analysis requires modeling of hardware and software resources
 - Layered queuing network approach is needed
- Desirable to have "standard" specification methods converted into queuing network models

Existing Approaches

- Various tools and models for distributed system modeling – using a "layered approach"
 - Tool: Spe*ed[7]
 - Queueing network model generation from a software model specification, both hardware/software resources are specified
 - Layered Queueing Networks (M. Woodside et al)[5]
 - " Generated from Use Case Maps, similar
 - Method of Layers (Roila, Sevcik)[4]

Ongoing Work at IIT-B

A tool for performance analysts Should be simple Should have intuitive specification Should do simple models Take away repetitive tasks from performance analyst Leave advanced tasks to performance analyst

CFA- Call Flow Analyzer

Joint work with Mohit Gupta, now with Tata Consultancy Services.



 Specification based on "call flow"
 Currently, simple calculations
 based on approximate
 open queueing models

CFA- Call Flow Analyzer

- ◆Intuitive specification ∠ analytical solution
- Layered model
 - Software servers executing on hardware server
 - Hardware resources can be specified separately (server uses x ms on CPU, y ms on Disk)
 - Simple model of network links also included

CFA- Call Flow Analyzer*

lie Add	Edit Wor	kaniq I I Z I I Z Z		-	Physics	al Server	ธนท1	•		
			16		Name		c:02			
Add Se	rver 👘				Duesne	Dariphne	las	•		
Add Logics	Server	1	-		Dovice	Type	CEU			
Name	CCard	26.00	CCard	<u> </u>		Add	Cput		Dono	
Add Sorve	es		Therety	and and a		d Condro				
Service	c.ecit	Add	arup	 Delete 	C	au service		illi (dilli		
	-100	2001			Sener	1660	EC T			
		Next>>			- Server	n hund		del lare	(h T	
		Next>>	1		Server Server	v Lied	<u>ra •</u> i. А	ld su	th T	
Placinal	unical Marati	N8X(>>			Server Server		i. A Server Mappi	ld su	uh 🔻	
□ Pigesical L _ Jervers	ugical Mapui	NBX(>> 10		Add Resource	Rem Resauce:		i. A Server Mappo AP	dd aw 1g r sun2	(h v	Set Physical
■ Piposical L ■ Jersers 9 © sorvint ■ init	ungical Mappul	NBX(>> 19 1 - (1)) (+950).(19		Aubut Raescourter 6 1 me	Rem Resaucca:		i. A Server Mappi AP Logicol Si	ld au 19 19 19 19 19 19	(h 🔻	Set Physical yolcal Eerver
Pleasical L Servers P C condition init	ungical Mappul cru J	NBX(>> 10 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	Se vice	Add Besinnitin 6 1 me	Renn Rissauccae		i. A Server Mappi AP Logicol Si Act	dd aw 1g 1 sun7 1901	uh 🔻	Set Physical yolcal Corver
Pleasical L Servers P C condit I init I init P C LOAF	ungicial Mappul crum	NBX(>> 19 1 ~ (* 1)) (* 950L (* 3	Sevice	Add Beston (); 6 1 1)@	Renn Resauccae		i. A Server Mappi AP S Logicsi Si Acti P	id au ig sun7 rvor	uh 🔻	Set Physical yolcal Corver
Pleasical L Jervers P C condit C mil C m 1 P C LOAF	ungicial Mappul I com	NBX(>> 19 1 - (1)) Hesource	Se vice	Add Besamin:	Rern Resaucce:		ка 	id su ng sun2 r70	uh 🔻	Set Physical yolcal Eerver
Pipesical L Servers Constrat init I m 1 Second	ungiscal Magagui t	NBX(>> 10 1 • (11) (11) (1950).(13	Se vice	Add Beston ();	Rorn Resaucce:		i. A Server Mappi aµ Logicol Si Act P	ld su ng sun2 nvc	Uh 🔻	Set Physical volcal Corver
Pleasical L Jervers P C condit C mil 0 m 1 9- C LOAF	ungincial Magaqui t	NBXC>> 19 1 ← (* 11) (* 950L, 12)	Se vice	Add Beston ();	Rem Resauce:		i. A Server Mappi aµ Logicol Si Act P	id su ng sun2 nvc	Uh 🔻	Set Physical volcal Eorvor

CFA- Call Flow Analyzer

I Ink Perti	nomance	Lotai	Hubzations	Call Pertormanc	e	
inpuns	Wariek	and	Service	Performance	Resource Perform	ance
	Metho		1	Value	L nit	1
Log n	5.0		0.0028		10. 24 [.]	1
Malan Thio	aug hiput		42.0		Jobs/Sec	
Rasporse	Time		0.079213	851	Seconds	
Extremu 🕾	geeune			10.0	025A230M3600	
MaxThrou;	hPut		54,7945	2	Jocs/Sec	
LISTUSG				~		
M=an Thru	ushinu.		82.0		ປມເສຽຍ:	
Resporse	e Time		0.04339	8	Secunds	1
Esterun	2002				920020MM03	
MaxThrou.	shPul		124.323	56	JucaSec	
DeleteAcc	8		1		10. av	
Maan Thru	iushpu.		10.0		Juca/Sec	
Respo [®] se	ə Timə		0.03/10:	235	Secunds	
Esterna	greenes		10.50 40	2988	920000000	1
MaxThrou.	shPul		13.6956	3	🛄 Sulve Fur Maxi	murn Arrivat
					CallFlow	Lagm
					Rasponse Time:	0.22
						1
					More addresses	200 E

Analysis results in End-to-end response times of each user request Maximum possible throughputs for each "call-flow" Also computes maximum supportable arrival rate under te 💠 🕅 average response time constraint

Max Arrivals Allowed 42.75002

Solve

Summary

Service performance engineering has significantly different challenges than those of product performance engineering

- Many are not traditional queuing theory problems
- Focus should be on available means and relevant analyses – this shifts focus to measurement tools, and tools that translate intuitive specifications to simple models
- More work necessary on understanding how to analyze a gray box based on operating system measurements (some patent-pending work done in AT&T labs)

References

"	P.K. Reeser, R. D. van der Meri, R. Hariharan, "An Analytic Model of a Web Server",
22	ITC-16, 1999.
	R. Hariharan, W.K. Ehrlich, D. Cura, P.K. Reeser, "End-to-end modeling of Web Server Architectures", ACM Conference on Performance Analysis of Web Servers, 2000.
"	A. Avritzer, R. Farel, K. Futamura, M. Hosseini-Nasab, A. Karasaridis, K. Meier-
	Hellstern, P. Reeser, P. Wirth, F. Hubner, D. Lucantoni, "
	Internet Application Performance: A Signature-Based Empirical Approach", in ITC-18, 2001.
>>	J.A. Rolia and K.C. Sevcik, "The Method of Layers", IEEE-TSE, SE-21, 8 (August
	1995), 689-700.
??	Dorin Petriu, Murray Woodside, "Software Performance Models from System Scenarios in Use Case Maps" Proceedings of Performance TOOLS 2002 London April 2002
>>	V. Mainkar, "A Model of a Web Server with Dynamic Content", INFORMS Fall 1999
	Meeting, Philadelphia, PA.
>>	"SPE*ED – The Software Performance Engineering Tool", http://www.perfeng.com/

Back-up slides

🗖 Enill losy A	nalyse	er i Se po r	n				R
Link Perform	ance	Total U	lizations	Call Performance	1		
Inputs V	Varkio	ad	Sprvice	Performance	Resource	enformance	8 .
Metro:	le-	oume	Lierve	r I=rxice	Value,	t.ntt	10
Ihm ghp 1	:pu11		i=rv et	1 TH	1 11 11	dinhaf led	1
Unitration	:pi11		(=rv et	11 th	2 - 00 002	l'ement	111
u leue i ength	:pu11		I=rv et	1 T	11.4501566017	dinhs.	
P=sid=nc=	3pi11		l=rx et	17 A	0.0007736	Necond=	
Thm ghp 1	nsF1	,	1=rx et	int.	1 1111	Inha()=c	-
Unitration	13F1	20	1= rv et	1°1	2-10.102	l'ement	- 11
u ene i engih	nisF1	1	I=rx et	in the	11.42622967	d-hs	- 11
Pesidence	nisF1	10	l=rv et	t≓nt	ILTIEFS (DR. 21	lier ond=	
lhm ghp t	:pu11		i=rv et	m dd	2II	Inhaf lec	
Utilization	pp111		(=rv et	m dd	H HHH H	l'ement	- 11
u eue i ength	:pu11		I=rv et	m dd	11 150-4041	dinhs.	- 11
P=sid=nc=	3pi(11		l=rx et	m dd	0.0007736	Decond=	
Thm ghp 1	nsF1*		1=rx et	m dd	4 11	inhar isc	
Unitration	nisH11		I=rv et	m dd	12 101 1014	l'ement	
u ene i engih	nisF1*		l=rx et	m dd	0.47050027	dinhs.	÷

ColFlow Analyser Repo									
1.	CalFlow Analyser Report								
Independent of All Indexed	Service Performance	Resource Performance							
t∕ ebi:	Value:	Linit	ſ						
Servlet(nit)		114 24	120						
M≘an Throughput	130.2	Lobs/Sec	31						
R≊sporse Time	2.0070522744	Seconds							
Extremu **	insurcesses.	CONTRACT.							
WaxThrou <u>c</u> hPut	276.09573	Lobs/Sec							
oson in sectors	50-2 640/03km	and a second sec							
Servlet(** idd)	100.00	114 - 24							
Maan Throuphpu.	10.0	Lobs/Sec	1						
R≘spo [_] se Time	2.015538292	Seconds							
Esternu	Possa facistico e	100 m Mm 0							
MaxThrouphPul	55.126384	Lobs/Sec							
Servlet(Tina)									
Mean Throushpu.	130.2	Lubs/Sec							
R∋spur se Time	1.0035261372	Seconds							
Esternu	Naposocial Marco	Constant of the second s							
MaxThroughPut	378.59573	Lobs/Sec							
24000 E14 C24 C31 A C34 C	NOW OCCUPATION	1450502013	1						

CFA- Call Flow Analyzer



User-oriented results

System performance

measures

Ongoing work at IIT-B



software

Examples of co-ordination work: Consider a load generator which is running in a mode in which it increases the load level every 10 minutes. The tool can do two types of tasks:

•*Routine:* e.g. automatically mark data collected on the server side so that the corresponding load level can be identified

•Intelligent: e.g. figure out how long a duration of test is necessary to get "steady-state" results

Tools Ongoing work at IIT-B



Load generator software

- Examples of capacity analysis work:
- *Routine* calculations:
 - Load level (number of users, request rate, resource utilizations...) at which some performance requirement is met.
 - •Generating graphs of throughput vs number of users, response time vs throughput, etc.

Intelligent calculations: "knee" of response time curve, where does throughput curve flatten out...

Queueing Model : CPU

Flow of typical servlet that generates dynamic content :

Request for CPU : t1 secs

Wait for I/O with back end system : w1 secs

Request for CPU : t2

Wait for I/O with back end system : w2

Request for CPU : t3

CPU modeled as a processor sharing queue Arrival rate of requests to this queue = Web transaction throughput rate X number of CPU request segments in the servlet

Hierarchical Queueing Model

- Then, holding time of servlet is = w1 + w2 + + $R_{cpu}(t1) + R_{cpu}(t2) + R_{cpu}(t3) + ...$
- where R_{cpu}(t) is the response time of a request in the CPU queue

Model variables are interdependent, so iterate until convergence is achieved.