CRIM

From Functionnal to Performance
The easy way…

Presented by
Raymond Rivest
Advisor, Test Specialist
CRIM, Center for Test and Interoperability

Presented at WOPR 16
April 2011

Partenaire financier :
Développement économique, Innovation et Exportation Québec

CRIM

The project

- An Insurance Software Developer needed to have its software evaluated by a neutral organization for a sale
- Proof of concept aimed towards a LoadTest Tool
  - 300 Active connexions (users logged in)
  - 5 years database history
  - **Customer records with ‘2008’ CustId = 9872.0**
  - **Policy Transactions for ‘2008’ CustId = 26464.0**
  - **Average Policy Transactions for these Customers 2.68**
  - **Average Policy Transactions per week 508.92**
  - **Average Policy Transactions per day 101.79**

CRIM

Let's get started

- Underlying protocol is COM/DCOM
- 1500 lines of code for a single login script
- Code is messy and barely replays
- Time needed to fix every transaction will be huge

- Way out of the initial POC

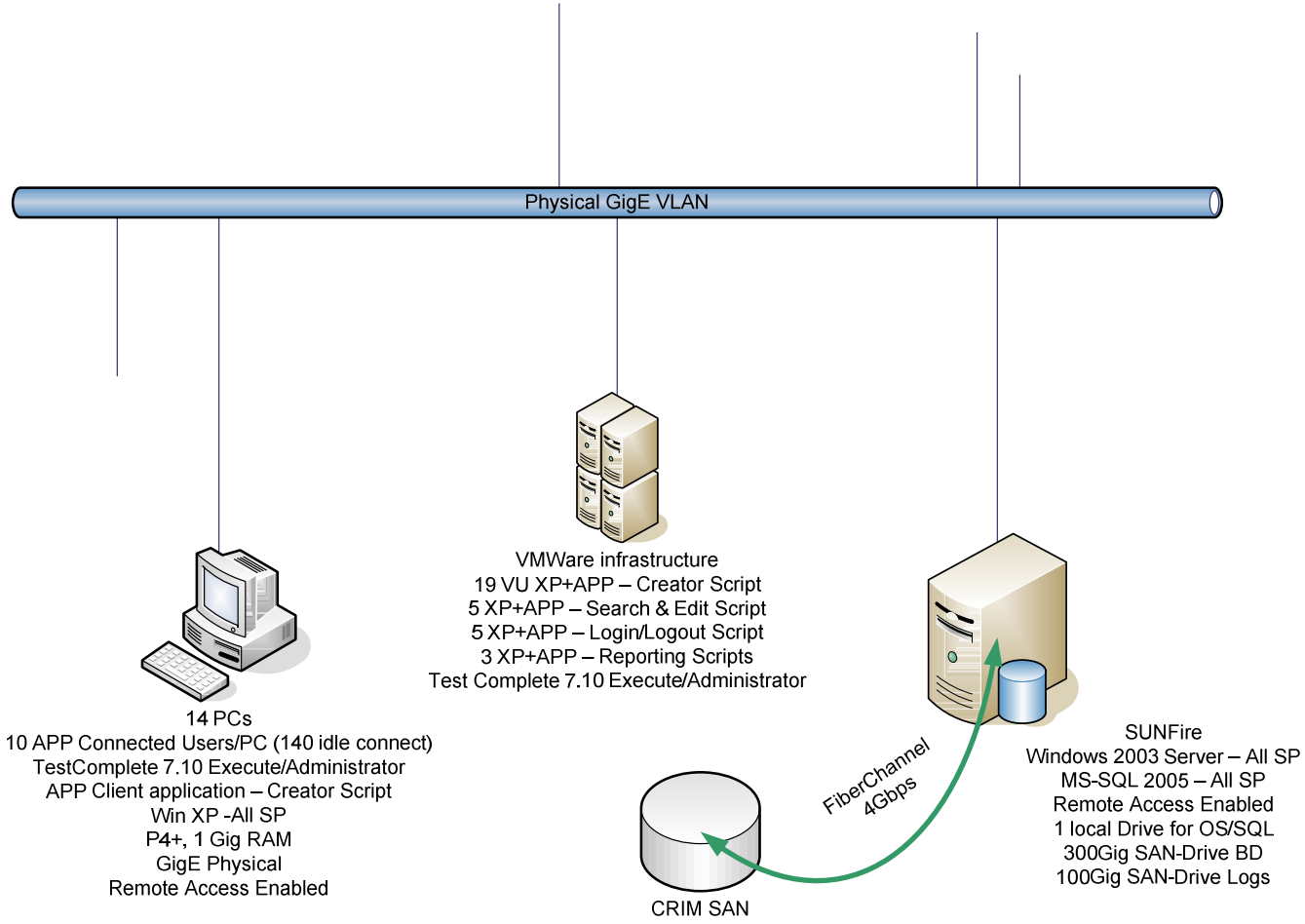
CRIM

Back to the maths

User Types	Volumes	Raw numbers	Per business days	Per Hours	Per Employee Per Hour	Time per transaction
(Base numbers & units)			255 BD	7 per days		(Minutes)
1. Customer personal lines (125 employees) in 10 modules of 3 person groups handling specific clients	12% of NBS on a total of 100 000 policies in the last year which means 88% of renewal in the daily process.	100000 (88%)	345,0980392	**49,2997199**	0,394397759	152,130682
		100000 (12%)	47,05882353	**6,72268908**	0,053781513	1115,625
2. Customer Commercial Lines(140 employees) in 2 modules (over $5000 and under $5000)	13 873 policies + 1781 bounds , 16% of NBS last year in all the revenue of the department.	13873+1781 (84%)	51,56611765	**7,36658824**	0,052618487	1140,28363
		13873+1781 (16%)	9,822117647	**1,40315966**	0,010022569	5986,48908
3. Customer Accounting (20 employees)						
4. Customer - Intact affiliation (90% of policies are with Intact)						

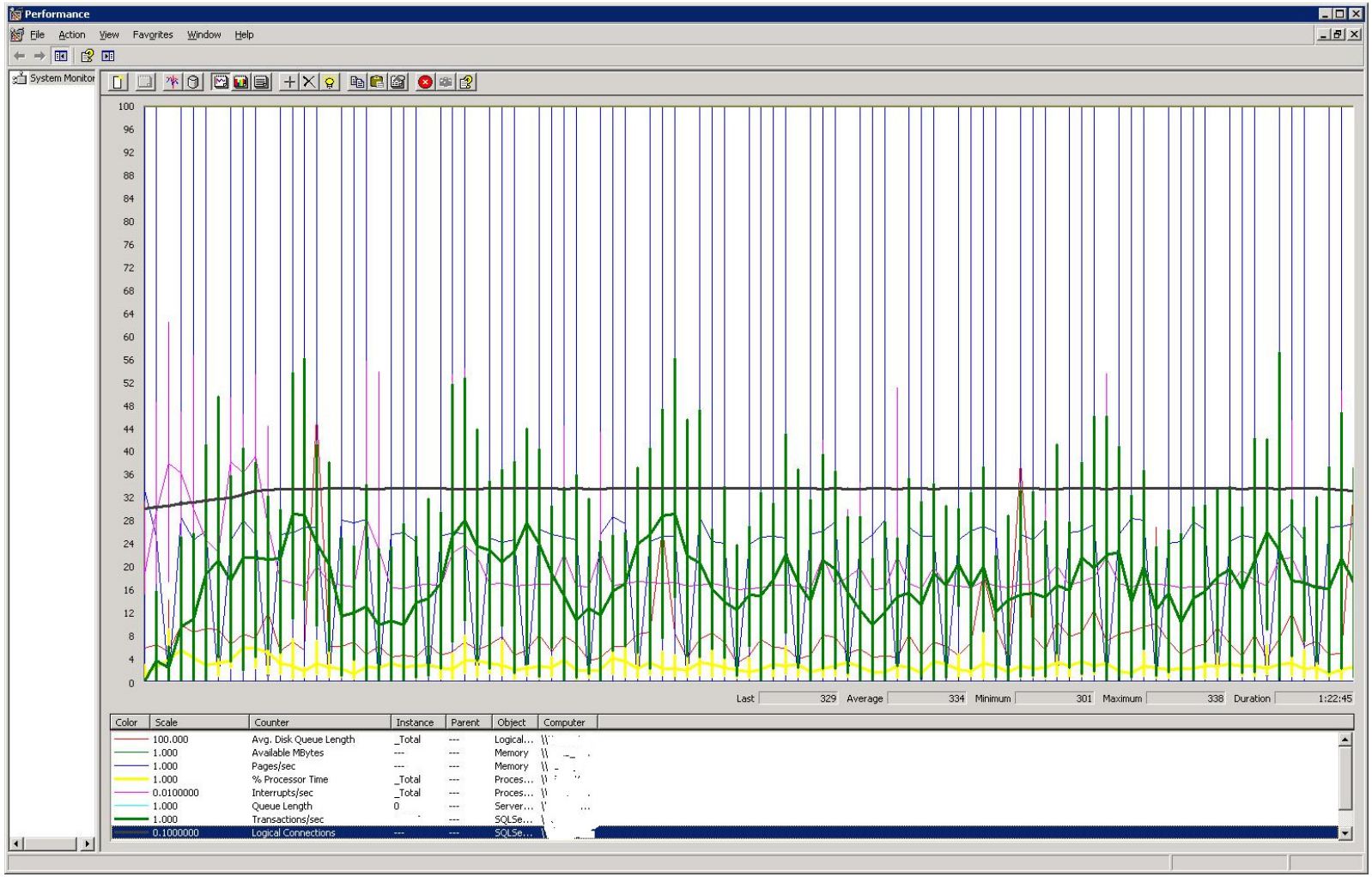
**Time per Transaction = Theoretical time per transaction (according to transactions per hour per employee)

Functionnal to Performance



** First run had another 14*10 VM connected to achieve the 300 active connexions.. As it did not made any change, focus was set on the transaction rate.

Some porn

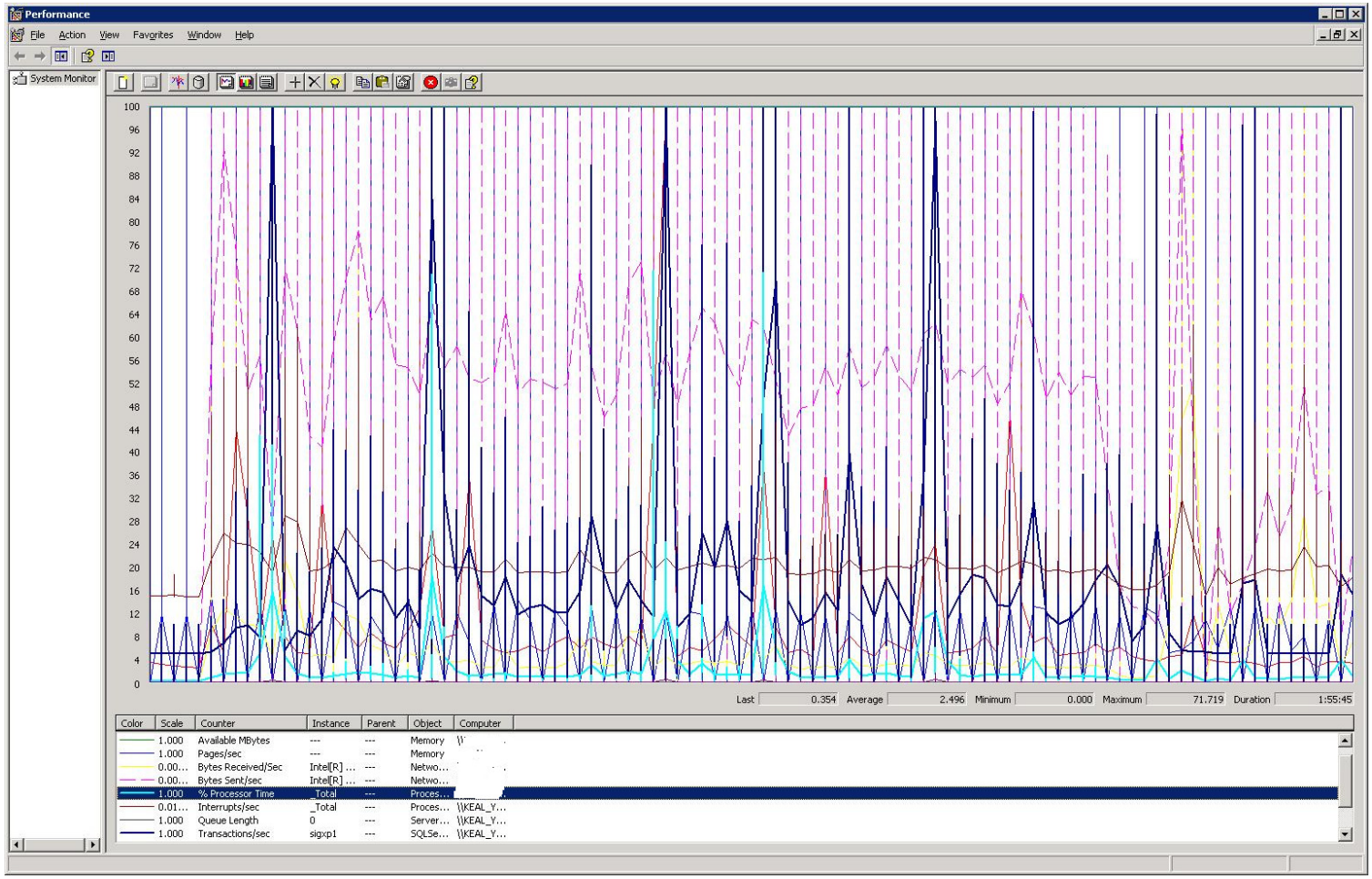

Performance
File Action View Favorites Window Help
System Monitor
100 96 92 88 84 80 76 72 68 64 60 56 52 48 44 40 36 32 28 24 20 16 12 8 4 0
Last 329 Average 334 Minimum 301 Maximum 338 Duration 1:22:45
Color Scale Counter Instance Parent Object Computer
100.000 Avg. Disk Queue Length _Total --- Logical...
1.000 Available MBytes --- --- Memory
1.000 Pages/sec --- --- Memory
1.000 % Processor Time _Total --- Proces...
0.0100000 Interrupts/sec _Total --- Proces...
1.000 Queue Length 0 --- Server...
1.000 Transactions/sec --- SQLSe...
0.1000000 Logical Connections --- --- SQLSe...

CRIM

How it goes…

- In essence, over 300 Virtual PCs, on which a Functional Scripting Tool (Test Complete 7.10) was performing User Actions at a specific rate.
- For each PC, the scripts would perform actions (field change, pane change, listbox selection) within 1 second, key rate was at 0.5 seconds and New Window wait time was maxed at 25 seconds, leaving little margin for response time degradation.
- In order to achieve the required load with the hardware limitation of the test bed, transactions duration were reduced to the shortest possible time. By compressing transaction time, less active users were required to achieve the full load. A session was performed with the following:
  - 14x10 Idle logged in users (140 connected but inactive)
  - 19 Customer+Policy+Billing Creators (1 per 10 minutes each)
  - 5 Search&Edit (1 per 5 minutes each)
  - 3 Reporters
    - - Bill printing (1 per 10 minutes)
    - - Production (1 per 15 minutes)
    - - Balance report (1 per 15 minutes)
  - 5 Login/logout (1 per minutes each)

More porn


Performance
File Action View Favorites Window Help
System Monitor
100 96 92 88 84 80 76 72 68 64 60 56 52 48 44 40 36 32 28 24 20 16 12 8 4 0
Last 0.354 Average 2.496 Minimum 0.000 Maximum 71.719 Duration 1:55:45
Color Scale Counter Instance Parent Object Computer
1.000 Available MBytes --- --- Memory
1.000 Pages/sec --- --- Memory
0.00... Bytes Received/Sec Intel[R] ... --- Netwo...
0.00... Bytes Sent/sec Intel[R] ... --- Netwo...
1.000 % Processor Time _Total --- Proces...
0.01... Interrupts/sec _Total --- Proces... \\KEAL_Y...
1.000 Queue Length 0 --- Server... \\KEAL_Y...
1.000 Transactions/sec sigxp1 --- SQLSe... \\KEAL_Y...

CRIM

End result

- Target load achieved (and a live view of response time)
- A happy customer (Made the sale and end customer has been running for 2 years without performance glitch)
- Within planned budget (Happy management)
- Reusable scripts for Customer's Acceptance Tests (Repeat business)
- Customer is coming back for a new project…

CRIM

Drawbacks

- Scalable up to a certain limit
  - Hardware needed per VM (How many..)
  - Runtime licenses (Not all tools have it, $$)
  - Need rock-solid scripts
    - Not like WEB
    - Rely on the GUI
  - Man Power to watch the test bed