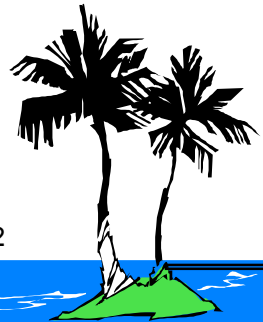


1998 DI USERS GROUP

DI Translator Benchmarking Capacity Planning and Performance Tuning

Paul Fosse
Infinet Solutions



DI1112/2

DI/EDI RIDE THE WAVE!

The Problem...



Benchmarking for Performance Tuning and Capacity Planning

▲ Takes a lot of time because you need:

- ✓ **System Installed**
- ✓ **Environment Setup**
- ✓ **ADF**
- ✓ **Trading Partners**
- ✓ **Maps, etc...**

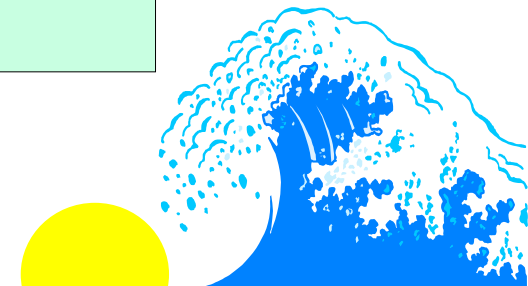


You Need Information to Help You:

Plan if new transactions can meet the business objectives in the current environment

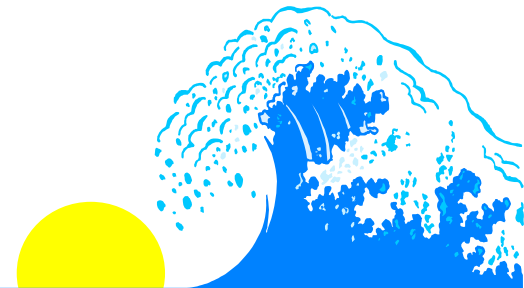
Tune environment to reduce need for CPU charges or upgrades

Plan when to move to a larger box



People Go to Extremes

- ▲ No performance testing (cross your fingers and hope for the best)
- ▲ Run exact situation (lot of work for every transaction)
- ▲ DI doesn't care if it is an 810 or an 850, it only cares about size and complexity



A Solution

▲ Using the DI formula to eliminate Benchmarking for Performance Tuning

✦ How was this developed?

➤ Used a simple 850 outbound mapping

- 3 different transaction sizes (100, 1,000 and 10K)
- 3 different number of transactions in envelope (1,100 and 10K)
- Ran Translate and Envelope, separately and together
- Ran with all the combinations of Management Reporting, Transaction Store and Image
- Tested with up to 2.3 million executions of special literal mappings to determine it's effect on translation



A Solution (cont'd)

- **Timed MVS Batch jobs with different workloads**
- **CICS timed with TMON and doesn't include CPU time of DB2 subtasks (therefore not really accurate)**
- **On each test varied only ONE variable so all of difference must be attributable to that ONE change**
- **Tests run on a development system in the morning so not much interference with other tasks**



And the formula is....

✦ **Formula to predict time for a simple send map doing a PERFORM TRANSLATE and ENVELOPE is:**

➤ **Overhead to start and stop translator**

▸ **In MVS Batch:**

▪ **.4700 CPU seconds and 7.000 Clock seconds**

▸ **In CICS with Persistent Environment:**

▪ **.0450 CPU seconds and 5.000 Clock seconds**



And the formula is.... (cont'd)

➤ Transaction Costs

- ▶ .0070 CPU seconds and .0074 Clock seconds
- ▶ .0040 CPU seconds and .0200 Clock seconds more if Transaction Store is on
- ▶ .0010 CPU seconds and .0230 Clock seconds if image stored in Transaction Store
- ▶ .0070 CPU seconds and .0260 Clock seconds more if Management Reporting is on

Transaction Store, Transaction Image and Management Reporting Clock time can run in parallel if not busy

And the formula is.... (cont'd)

- **Size Costs per Kilobyte of data**
 - **.0040 CPU seconds and .0300 Clock seconds**
 - **.0010 CPU seconds and .0230 Clock seconds if Image Stored in Transaction Store**
- **Special Literal Usage**
 - **.0001 CPU seconds and .0002 Clock seconds for each special literal operation**
- **Error costs**
 - **.0006 CPU seconds and .0200 Clock seconds for each error reported**



Sending Batch of Invoices Example

- Sending 500 Invoices of 2000 bytes each in one batch with management reporting on and transaction store off

$500 * 2000 = 1,000,000$ bytes or 1000 kilobytes of data

CPU Time

0.47 seconds

$500 * .0070 = 35.00$ seconds

$500 * .0070 = 35.00$ seconds

$1000 * .0040 = 40.00$ seconds

Total 110.47 CPU Time

Wall Clock Time

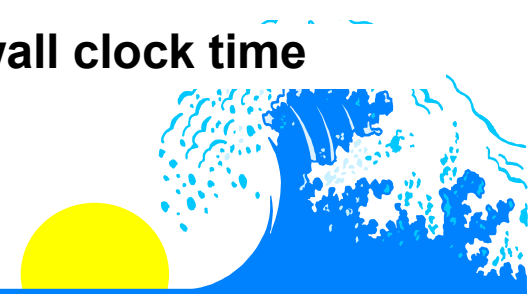
7. seconds - Start and stop time

$500 * .0074 = 37.$ seconds - Transaction time

$500 * .0260 = 130.$ seconds - Management Reporting Time

$1000 * .0300 = 300.$ seconds - Size time

474. seconds wall clock time



Sending Individual Invoices Example

- Sending 500 Invoices of 2000 bytes each in separate batches with management reporting and transaction store off

$500 * 2000 = 1,000,000$ bytes or 1000 kilobytes of data.

CPU Time

Wall Clock Time

$500 * 0.47 = 235.00$ seconds

$500 * 7. = 3500.$ seconds - Start and stop time

$500 * .0070 = 35.00$ seconds

$500 * .0074 = 37.$ seconds - Transaction time

$500 * .0070 = 35.00$ seconds

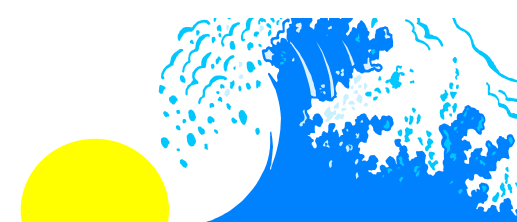
$500 * .0260 = 130.$ seconds - Management Reporting Time

$1000 * .0040 = 40.00$ seconds

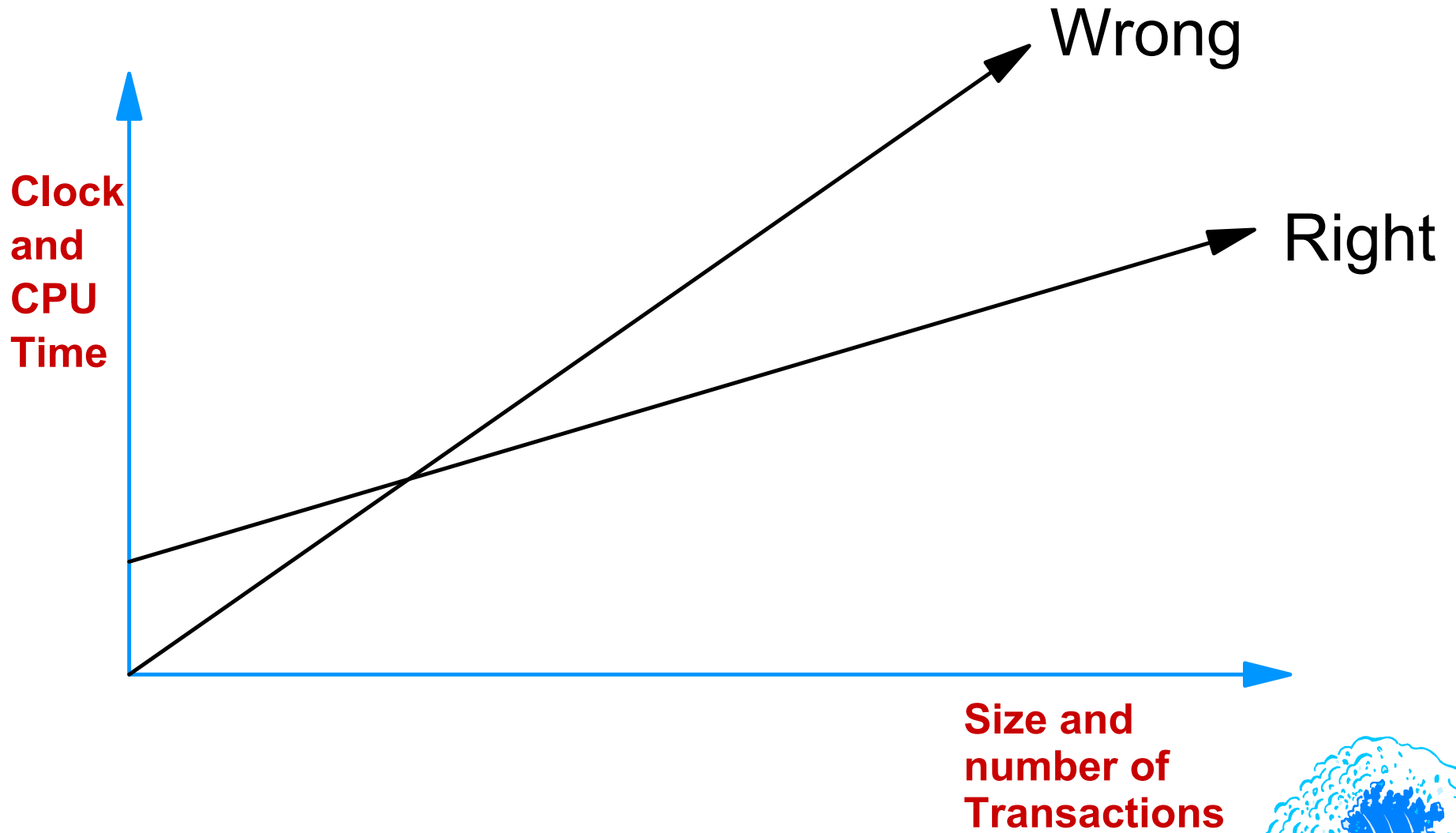
$1000 * .0300 = 300.$ seconds - Size time

Total **345.00 CPU Time**

3967. seconds wall clock time



Graph of Formula



Should You Scale My Formula or Make Your Own?

- △ **Make your own if you have the time and absolute numbers are important.**

"Isn't this same as benchmarking every transaction?"



"No. You are making a formula once, not for each transaction."

- △ **Scale my formula if you are in a hurry or are looking at relative numbers (need to improve something by 50%), just scale my formula.**



Adjusting the Formula for Your Environment

▲ This test run on a 9672-R75

- ✦ You can use A CPU (NOT a box) scalars **OR**
- ✦ You can run a simple test:
 - If it takes twice as long, multiply all my factors by 2
 - Important to use different factors for CPU intensive tasks vs. I/O intensive tasks



Lessons Learned

▲ Is Hot DI Right for Your Environment?

- ✦ Do you have a lot of small transactions that can't be batched?
 - Then it will save a lot of time
- ✦ Are you using the API to get the benefit already?
 - Then you don't need it
- ✦ Are you using Continuous Receive?
 - Can't use with Hot DI



Lessons Learned (cont'd)

△ **Can you increase your throughput by batching transactions?**

✦ **Yes, significantly if the transactions are small**

△ **Should I turn off the saving images or the whole transaction store to increase performance?**

✦ **Maybe. Plug in your numbers to the formula and decide if it is worth it for your company**

△ **Should I turn off management reporting to increase performance?**

✦ **Maybe. Plug in your number to the formula and decide if it is worth it for your company**

△ **Should I Translate and Envelope at the same time or do them separately?**

✦ **Yes, if you do them at the same time to save store search time or make the Envelope request as specific as possible so indexes can reduce search time**



How This Can Work For You

▲ Using the DI formula for Capacity Planning

- ✦ Plug in your numbers to the formula

▲ Developing your own formulas

- ✦ Remember to use sizes that vary by orders of magnitude, not by powers of 2
- ✦ This will allow you to interpolate, not extrapolate



Summary

▲ Benchmark Smart

- ✦ Use formulas to avoid the need to benchmark each transaction

▲ Reuse effort

- ✦ Trade tips with other DI users at the conference

