ORB Performance: Gross vs. Net

Brad Balfour
Objective Interface
1892 Preston White Drive
Reston VA 20191-5448

703/295-6500 (voice)
703/295-6501 (fax)
http://www.ois.com/
brad.balfour@ois.com

Topics

- **End-to-End Round Trip Time Measurement**
- **End-to-End Round Trip Time: Where Could the Time be Spent**
- **Comparing Apples to Apples – What to Hold Constant**
- **Ensuring Apples to Apples Comparison**
- **Measuring Gross vs. Measuring Net**

End-to-End Round Trip Time Measurement

- Typical ORB Benchmark
- What *Does* it Reveal
- What *Doesn't* it Reveal

Typical ORB Benchmark

- Measure the average value of a remote two-way call and return sequence
  - No impl processing except timing measures
  - No client processing
  - Small or no data to transmit
- Easy to gather

What *Does* it Reveal

- If a given scenario
  - On a particular platform
    - Hardware
    - Operating System
    - Network
  - For certain data
    - Types
    - Sizes (amounts)

  will or will not meet timing requirements
- Also:
  - Time to move data from point A to B
  - How many messages per second from A to B

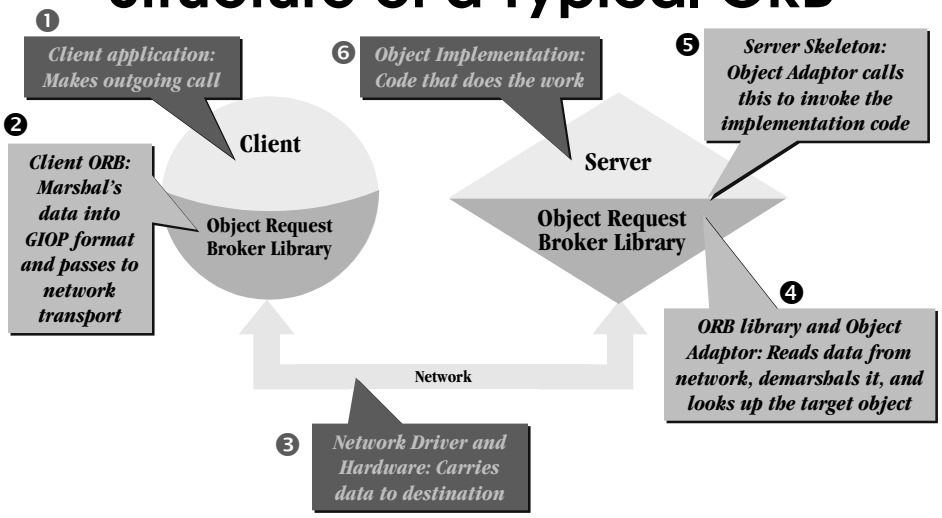
What *Doesn't* it Reveal

- The composition of the total time
  - Amount (%) spent in the TCP/IP stack and on the network transmission
  - Amount (%) spent in the ORB ("Overhead")
    - Time to marshal & unmarshal the data to/from GIOP format ("Variable Overhead")
    - Time spent in demultiplexing, upcall, context switches, system call overhead, etc. ("Fixed overhead")
- Predications: How an individual data value will relate to any non measured values
- Will the application be able to meet requirements for scenarios and platforms that cannot currently be tested?

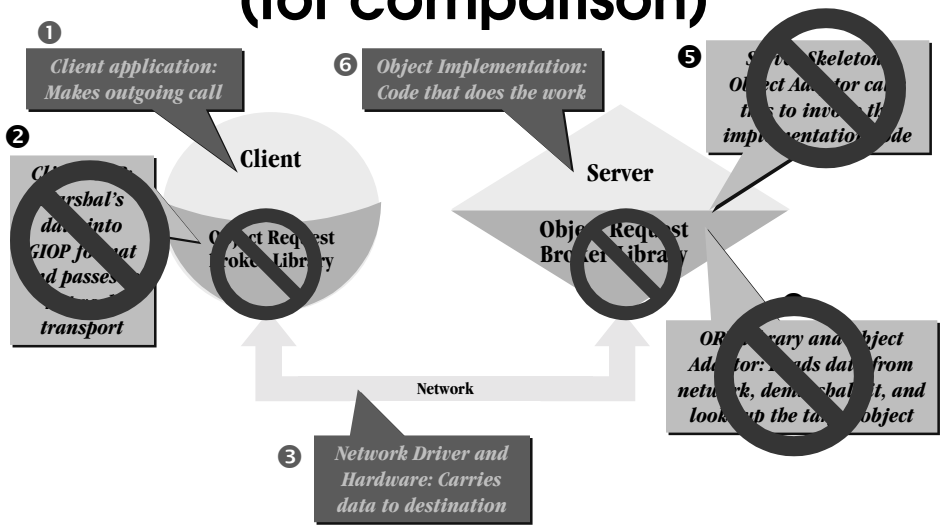
End-to-End Round Trip Time: Where Could the Time be Spent

- Structure of a Typical ORB
- Minimal Socket Program
- Generic ORB Overhead Assertions

Structure of a Typical ORB



Minimal Socket Program (for comparison)



Generic ORB Overhead Assertions

- In an ORB comparison benchmarking application, steps 1 & 6 are "null". Step 3 is held constant
- End-to-end round trip time is measured
- Measurements have shown a wide range of results for different ORBs
  - All numbers are *approximate generalizations* of "null data two-ways" on typical desktop/workstation hardware
  - Desktop/Enterprise: times as long as 2000µs (2 ms)
  - Java/Browser ORB: approximately 1500µs (1.5ms)
  - Optimized Research Desktop ORBs: 800-1200µs
  - Real-Time ORBs: 350µs-1100µs

Gross Time Comparisons

- Comparisons of End-to-End Round Trip time
  - Tells who won the race
    - *That's All*
  - This may or may not generalize to other situations
- To see this, look at:
  - Absolute Comparisons
  - Relative Comparisons
  - Problems with Relative Gross Comparisons
  - What is the Minimal Baseline?

Absolute Comparisons

- Scenario One: No parameters sent over 10BaseT network from Client to Server
  - ORB A End-to-End Round trip: 660µs (0.66ms)
  - ORB B End-to-End Round trip: 1,320µs (1.32ms)
  - ORB A is faster by 660µs (0.66ms)
- Scenario Two: 9KB of parameters sent over 10BaseT network from Client to Server
  - ORB A End-to-End Round trip: 10,110µs (10.11ms)
  - ORB B End-to-End Round trip: 10,950µs (10.95ms)
  - ORB A is faster by 840µs (0.84ms)

Relative Comparisons (same scenarios)

- Scenario One:
  - ORB A End-to-End Round trip: 660µs
  - ORB B End-to-End Round trip: 1320µs
  - ORB B is 100% slower/faster than ORB A
    - ((B-A)/A)*100=((1320-660)/660)*100=A is 100% faster

This percentage is misleading

- Scenario Two:
  - ORB A End-to-End Round trip: 10,110µs
  - ORB B End-to-End Round trip: 10,950µs
  - ORB B is 8% slower/faster than ORB A
    - ((B-A)/A)*100=((10950-10110)/10110)*100=A is 8% faster

This percentage is misleading

ORB B didn't suddenly get faster in scenario two

Problems with Relative Gross Comparisons

- Relative (%) comparisons of End-To-End Round trips are misleading
- They compare the "Gross" measurement
  - This is the reason they don't apply across scenarios
- Example of why:
  - In scenario two, doubling the speed of ORB A does *not* result in a halving of the End-to-End Round Trip result
    - Because the time spent on the network transmission of the data remains constant (and dominates)
    - Halving the End-to-End Round Trip speed requires changing the laws of physics

What is the Minimal Baseline?

- To be able to compare *both* absolute and relative values for the End-to-End Round Trip, benchmarks need to factor out the speed of the items "below" the ORB
  - These includes:
    - The TCP/IP socket stack
    - The Network card, drivers and firmware
    - The Network Hardware
- The simplest way to do this is by measuring the data transmission time for a simple C program using socket read/write routines
  - Measure for various data sizes

Ensuring Apples to Apples Comparison

- Measuring What Varies
- Net Performance: Definition and Computation

Measuring What Varies

- Want to Determine the time spent in the ORB
  - Performing marshalling, multiplexing, unmarshalling, demuliplexing, invoking, etc.
- Not easy to directly measure
  - So measure indirectly – by difference
- Measure time spent "below" the ORB in the End-to-End Round Trip
- Subtract this time from all ORB "Gross" End-to-End Round Trip measurements
  - Absolute measurements are still equivalent
  - Relative (%) comparisons are *more* meaningful

Net Performance: Definitions and Computation

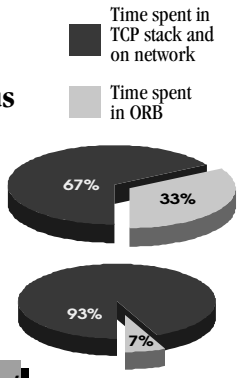
- LowerLayerTime = Round trip transmission time for an equivalent "socket" program to send the same size data
- GrossTime = Raw End-to-End round trip time measurement
- NetTime = GrossTime – LowerLayerTime
- %ORB "Overhead" = NetTime/GrossTime*100
- Assuming ORB A is the smaller GrossTime, then
  - %Faster or Slower = $((Net_B-Net_A)/Net_A)*100$

Measuring Gross vs. Measuring Net

- Absolute Net Speed and the Percentage ORB "Overhead"
- Relative Comparison Update
- The Two Components of Net Speed
- Why Do We Care?
  - How Much More Efficient To Catch Up?
  - Faster Networks: The Impact on Gross and Net
  - OS X vs OS Y: The Impact on Gross and Net

Absolute Net Speed and Percentage ORB "Overhead"

- For a given ORB at different data transmission values:
  - As the data size goes up, the absolute value of the Net *increases*
    - At no data transfer, for ORB A, the net is 220µs
    - At 9KB data transfer, for ORB A, the net is 630µs
      - More work is done to marshal the data
  - As the data size goes up, the Percentage value of the Net *decreases*
    - At no data transfer, for ORB A, the net is 33%
    - At 9KB data transfer, for ORB A, the net is 7%
      - Network transmission time dwarfs ORB "overhead" (for a Real-Time ORB)



Fixed and variable components of the Net speed exist

Relative Comparison Update

- **Scenario One:**
  - **ORB A End-to-End Round trip: 660µs**
  - **ORB B End-to-End Round trip: 1320µs**
  - **Socket End-to-End Round trip: 440µs**
    - **ORB A Net = 660µs - 440µs = 220µs**
    - **ORB B Net = 1320µs - 440µs = 880µs**
  - **ORB B is 300% slower than ORB A**
    - ((880-220)/220)*100=A is 300% faster than B (a 4:1 ratio)
- **Scenario Two:**
  - **ORB A End-to-End Round trip: 10,110µs**
  - **ORB B End-to-End Round trip: 10,950µs**
  - **Socket End-to-End Round trip: 9,480µs**
    - **ORB A Net = 10110µs - 9480µs = 630µs**
    - **ORB B Net = 10950µs - 9480µs = 1470µs**
  - **ORB B is 130% slower than ORB A**
    - ((1470-630)/630)*100=A is 130% faster (a 2.3:1 ratio)

This percentage is comparable

This percentage is comparable

Why aren't the two percentages the same?

The Two Components of Net Speed

- **Time to marshal the data parameters into GIOP/IIOP compliant format**
  - **This is proportional to the size of the parameters (in bytes)**
- **The remainder of the ORB's work on each remote call. This includes:**
  - **Overhead of kernel calls,**
  - **Marshaling and propagation of the 40-80 byte GIOP header,**
  - **Context switches, and**
  - **Demultiplexing and upcall**

  This is fixed and independent of the parameter size

Calculating Both Net Ratios

- Fixed Overhead (the Zero byte data parameter case gives this value):
  - ORB A is 300% faster in the fixed overhead component than ORB B
- Variable Marshalling Overhead
  - Calculate the rate of increase of Gross time as the data size increases
    - Socket rate = 9480µs - 440µs/9216 bytes = 0.981 µs/byte
    - ORB A rate = 10110µs - 660µs/9216 bytes = 1.025µs/byte
    - ORB B rate = 10950µs - 1320µs/9216 bytes = 1.045 µs/byte
  - Subtract socket rate increase from ORB's rate increase to get the marshalling efficiency
    - ORB A net rate = 1.025 – 0.981 = 0.044 µs/byte
    - ORB B net rate = 1.045 – 0.981 = 0.064 µs/byte
  - ORB A is 45% more efficient in marshalling data
    - ((0.064-0.044)/0.044)*100=A is 45% faster than B at marshalling

Why Do We Care?

- How Much More Efficient To Catch Up?
- Faster Networks: The Impact on Gross and Net
- TCP vs Shared Memory: The Impact on Gross and Net
- OS X vs OS Y: The Impact on Gross and Net

How Much More Efficient To Catch Up?

- Suppose that the 2nd place ORB wants to catch up
- Using the Gross numbers, it is not possible to correctly determine how much more efficient the 2nd place ORB needs to become
- Using the Net values, it is easy:
  - To catch up to ORB A's lower fixed overhead, ORB B must get 4x faster, i.e., it must cut it's time by 75% to 25% of the original value
  - To catch up with ORB A's lower marshalling rate, ORB B must get 31% faster and cut it's rate to 69% of the original value

Faster Networks: The Impact on Gross and Net

- The faster the network transmission time – at a given data size – then the more the time spent in the ORB dominates the End-to-End round trip speed
  - This matters when moving from 10Mb to 100Mb to GigaBit Ethernet or when switching to Shared Memory
- Using the Marshalling and Fixed Net numbers, it is possible to calculate how fast End-to-End Round trip will be on a faster network
  - Very useful for determining whether or not to upgrade the infrastructure to meeting performance requirements

Faster Networks: The Impact on Gross and Net (continued)

- Speed of ORB A on 100 Mb Ethernet (10x faster)
  - Hint: it won't be 1/10th of the Gross
- Project the socket Gross:
  - Zero byte transfer won't be significantly faster: still 440µs
  - 9K transfer time: ((9480µs-440µs)/10)+440µs= 1344µs
  - Therefore, all Gross ORB times speed up by 9480-1344= 8136µs
- Projected new End-to-End Round Trip (9K transfer)
  - ORB A = 10110-8136 = 1974µs
  - ORB B = 10950-8136 = 2814µs (still 840µs slower)

Faster Networks: The Impact on Gross and Net (continued)

- Variable Marshalling Overhead Recalculated
  - Calculate the rate of increase of Gross time as the data size increases
    - Socket rate = 1344µs - 440µs/9216 bytes = 0.098 µs/byte
    - ORB A rate = 1974µs - 660µs/9216 bytes = 0.143µs/byte
    - ORB B rate = 2814µs - 1320µs/9216 bytes = 0.162µs/byte
  - Subtract socket rate increase from ORB's rate increase to get the marshalling efficiency
    - ORB A net rate = 0.142– 0.098 = 0.044 µs/byte
    - ORB B net rate = 0.162– 0.098 = 0.064 µs/byte
  - *These numbers have stayed exactly the same*
  - ORB A is still 45% more efficient in marshalling data
    - ((0.064-0.044)/0.044)*100=A is 45% faster than B at marshalling

OS X vs OS Y: The Impact on Gross and Net

- Changing OS will change the TCP/IP stack and possibly the Network Interface's drivers
  - But the network media is unchanged
  - E.g., Windows NT replaced by Linux on x86
- Keep the ORB implementation, hardware and network all the same
- Comparing *Gross* numbers for version N of ORB A with version L of ORB A is not a good method
  - Will make it seem as though the ORB has gotten faster or slower
  - In actuality, the Net numbers for both versions should be similar (if not identical)

Summary

- When Benchmarking
  - Things to Do
  - Things *Not* To Do
- Net Performance Reveals Much About the ORB
  - True Relative Comparison Between ORBs
  - Amount of Change Needed To Catch Up
- Net Performance Is Critical to Changing Network, or OS and to estimating Replacement Protocols