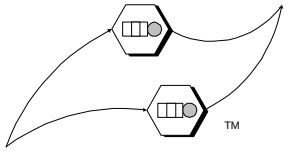
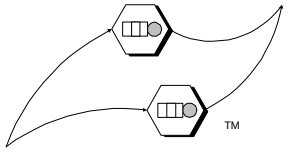


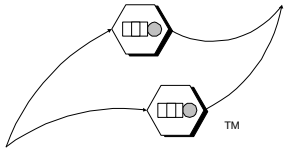
Simalytic Modeling: The Best of Both Worlds

Tim R. Norton
Candidate, Doctor of Computer Science
Colorado Technical University
Colorado Springs, CO
http://www.simalytic.com



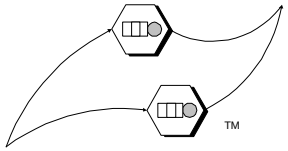
Overview

- Introduction
- Background
- Simalytic Modeling
  - Foundation
  - Response Time Comparison
- Conclusion



Introduction

- Applications
  - Once Batch Systems on a Single Computer
  - Now Multi-Platform
    - On-Line Transaction Processing
    - Client/Server Systems
    - GUI (Graphical User Interface) Front-Ends
    - PWS's (Programmable Work-Stations)
    - Departmental Servers and Mainframe Repositories

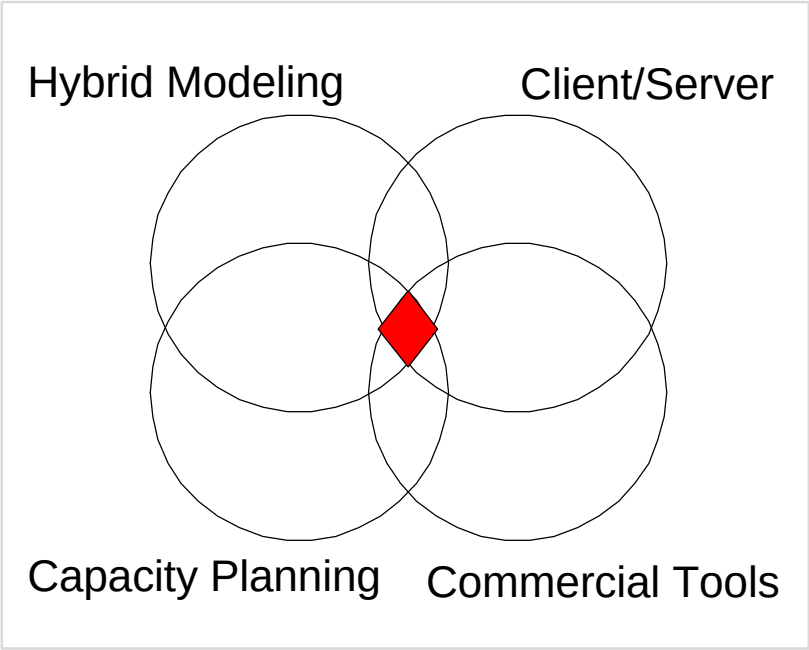


Introduction

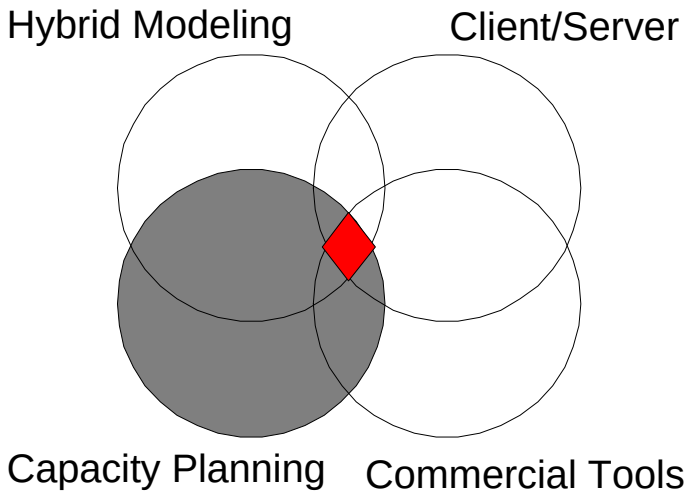
- Applications
  - Require Features and Services
    - Different Types of Computers (Mainframe, Open Systems, Desktop)
    - Different Operating Systems (MVS, Unix, OS/2, Windows, etc.)
    - Variety of Communication Network Techniques (RPC, DCE, NFS, FTP, SNA, APPN, etc.)

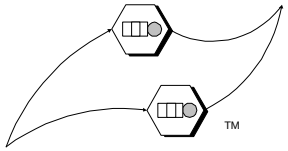
Introduction

- Model Application Requirements at Enterprise Level
- Intersection of:
  - Capacity Planning
  - Client / Server
  - Commercial Tools
  - Hybrid Modeling

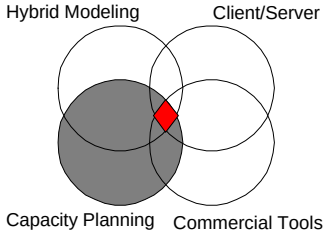


Capacity Planning

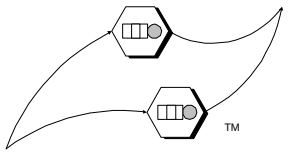




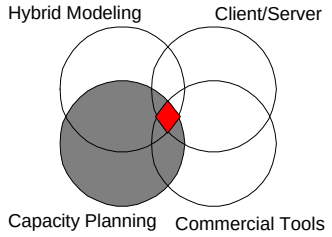
Capacity Planning



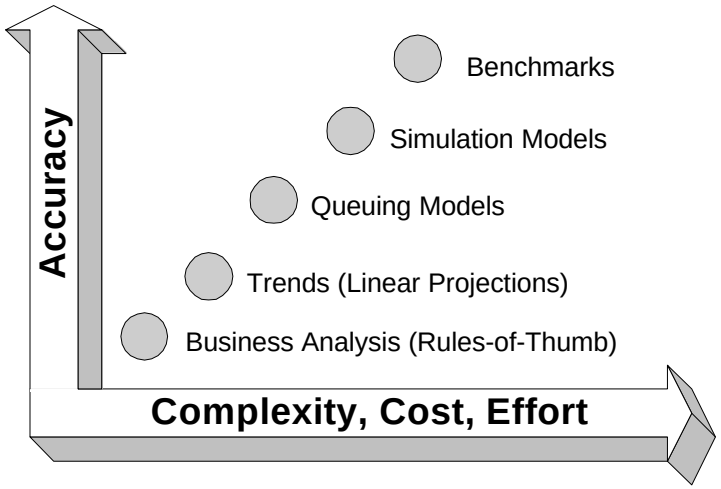
- Capacity Measured by Business Performance Objectives
  - Decisions About Resource Requirements Based Predicting Future Application Performance Using Business Goals and Expectations
  - What Do We Have to Buy and When Do We Have to Buy It to Make Sure That the Business Applications Perform at the Level Required to Insure the Business Succeeds?



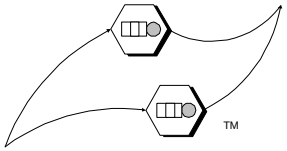
Capacity Planning



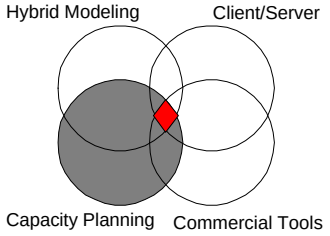
- Past
  - Processor Utilization
  - Overnight Batch Window



- Today
  - Operating Systems
  - The Platforms
  - The Clients
  - The Servers
  - The Networks
  - The Transaction Systems
  - Relationships

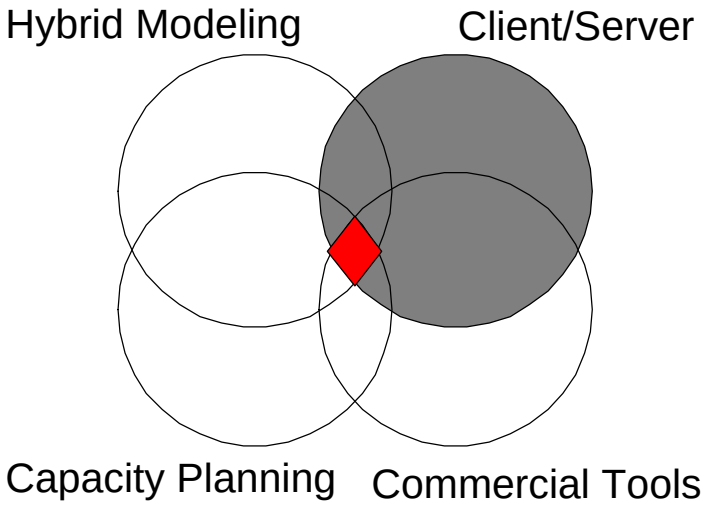


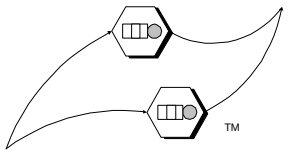
Capacity Planning



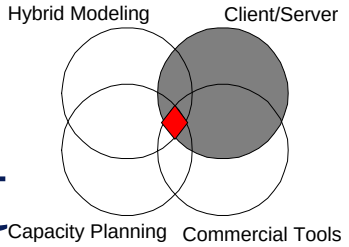
- Objective
  - Successful Middle Ground
    - Cost
    - Performance
  - Application Performance Assessed Against the Business Objectives and Goals.
  - Projected Business Volumes Modeled to Predict the Capacity Required to Meet Those Goals at Future Volumes

Client/Server

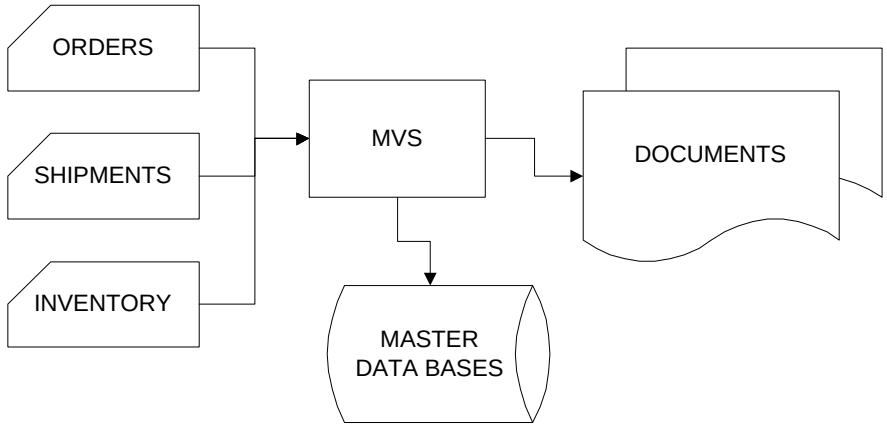


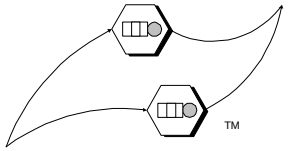


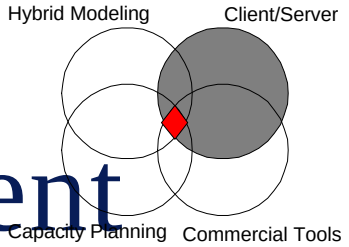
Centralized Environment



- Central processing
- Overnight
- Batch Orientated
- Data Entry
- Printed Reports

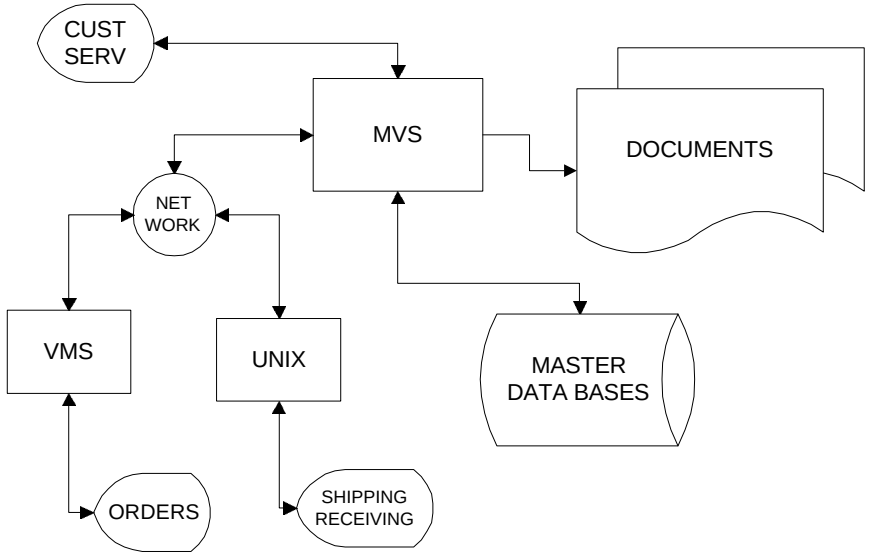


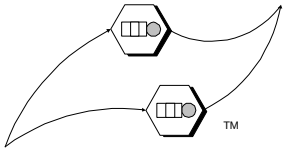




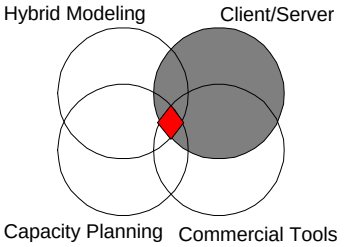
Decentralized Environment

- Overnight
- Data Entry
- Batch Orientated
- Printed Reports
- Downloaded Data

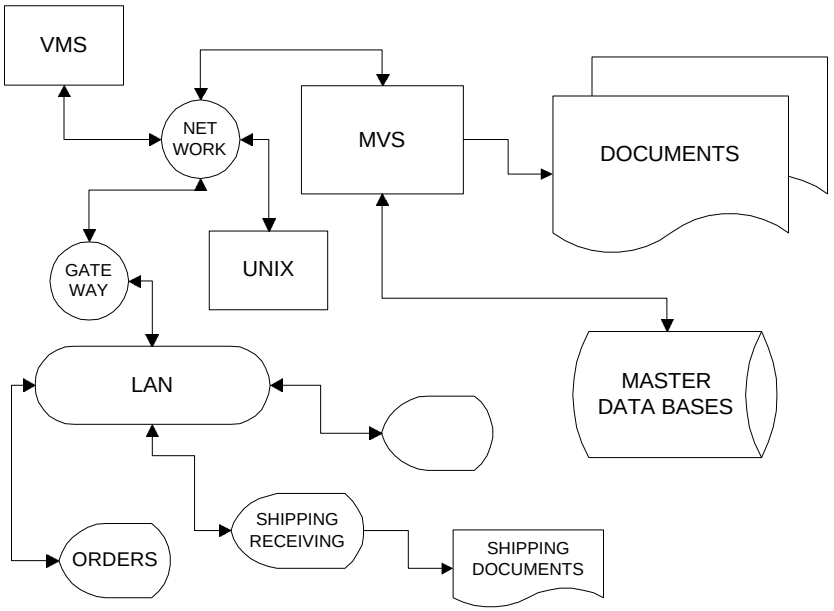


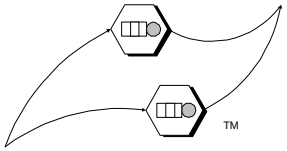


Client/Server

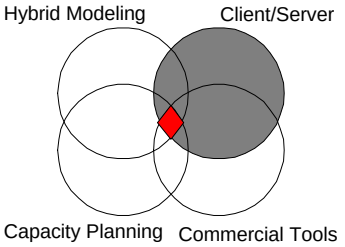


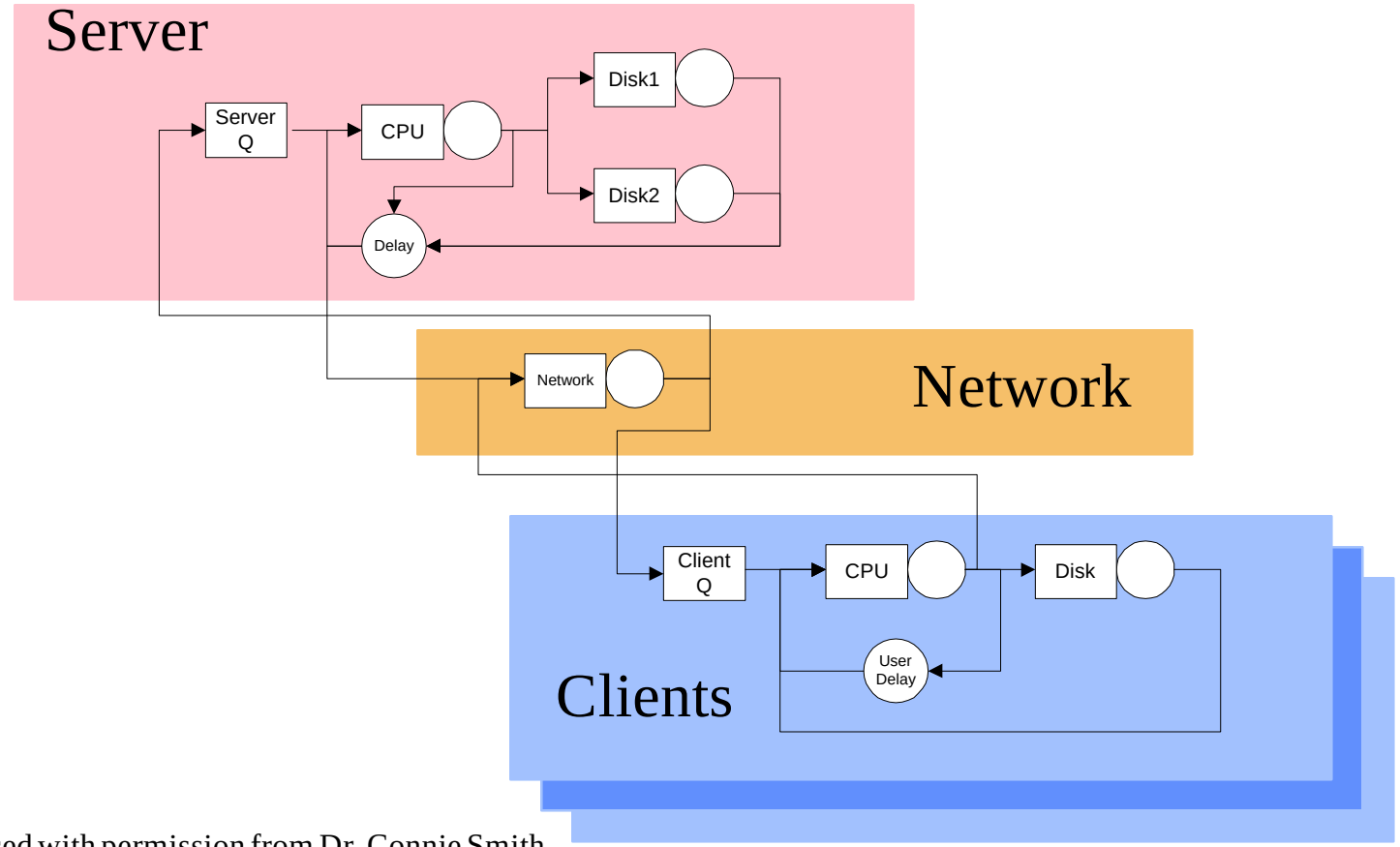
- Mix of Applications
- Different Systems
- Different User Interfaces
- Real-Time Interactions
- Inter-Dependent Workloads



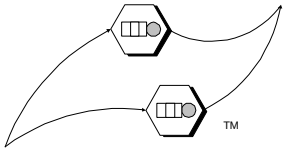


An Enterprise Model

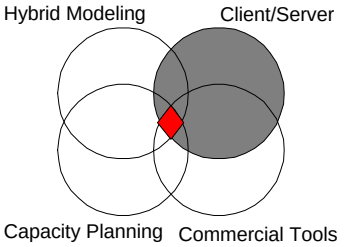




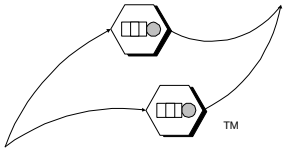
Reproduced with permission from Dr. Connie Smith, Performance Engineering Services.



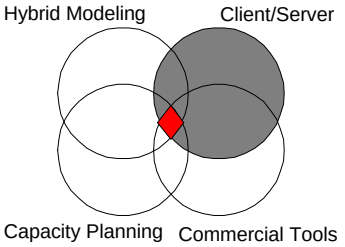
Client/Server



- Sizing Client/Server Applications
  - Right Systems at Each Level
  - Insure Systems Are Right Size
  - Too Small - Application Fails
  - Too Big - Excessive Cost

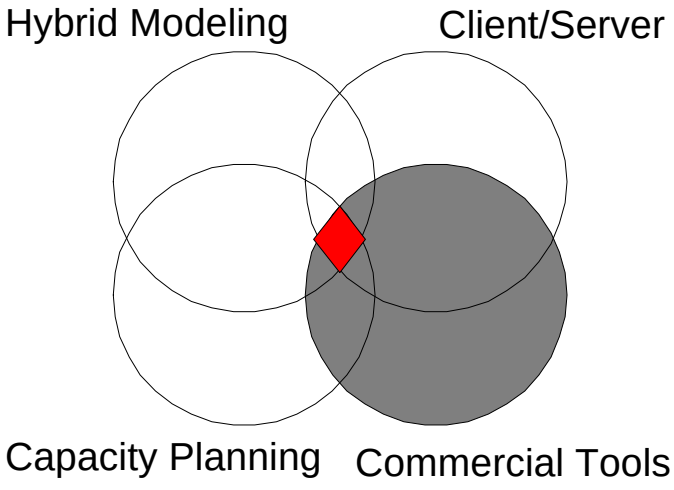


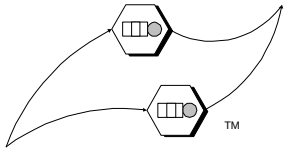
Client/Server



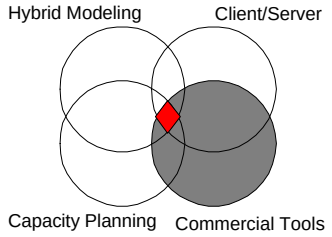
- Modeling
  - Systems Cannot Be Modeled Independently
  - Transaction Arrival Rate for One System May Be Dependent on the Response Times of the Others
  - Issues
    - Transactions to Several Servers
    - Synchronize Parallel Transactions
    - Wait for Response Before Sending the Next

Modeling Tools

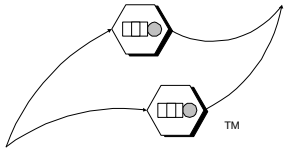




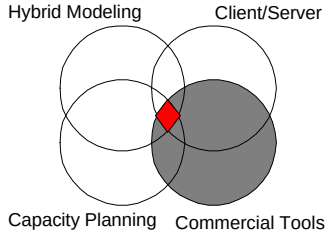
Modeling Tools



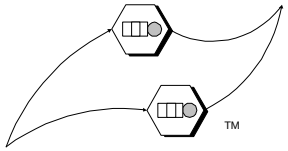
- Platform-Centric Tools
- General Purpose Tools
- Different Problem Sets



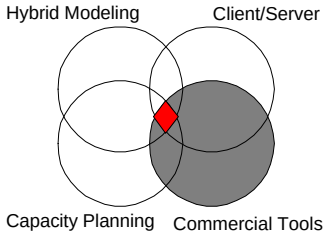
Platform-Centric



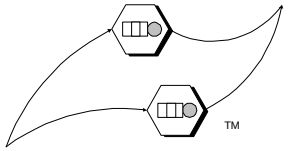
- Single platform at a time
- Detailed information about the platform
  - Number and type of processors
  - Speed and transfer rate of disk devices
  - Level of the operating system.
- Easier to build
  - “Building blocks” defined
  - Relationships fully understood



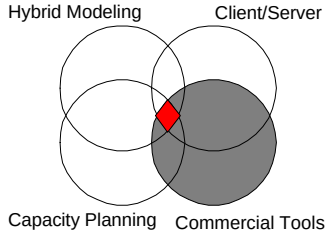
Platform-Centric



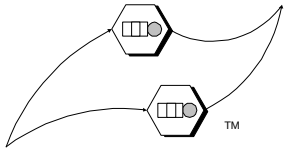
- Only Environments Built Into the Tool
  - Define New Servers
  - New Server Performance Characteristics
  - But Not Dramatically Different from Supported Platform
- Data Collected from Running Systems
- Generally Analytic or Queuing Theory



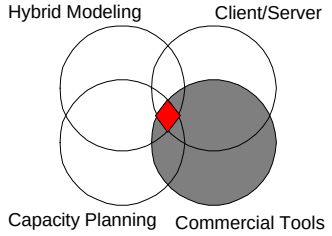
General Purpose



- Features to Model Anything
- No “Built-in” Platform Understanding
- Libraries of Sub-Models
- Model More Than Just Hardware

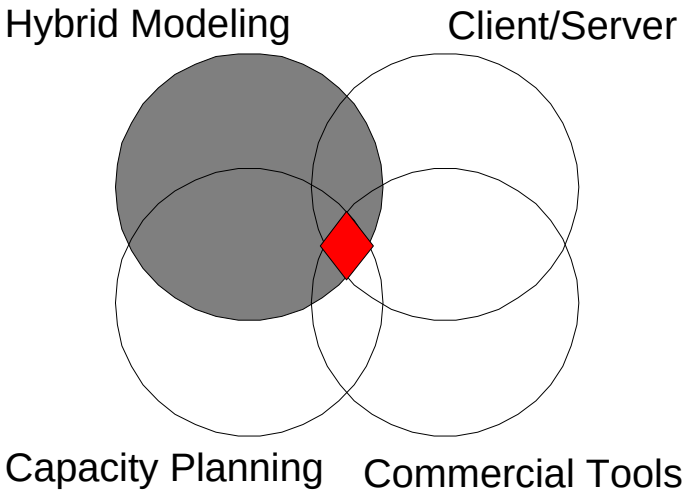


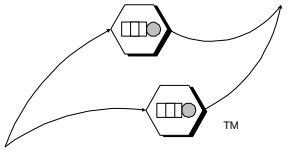
General Purpose



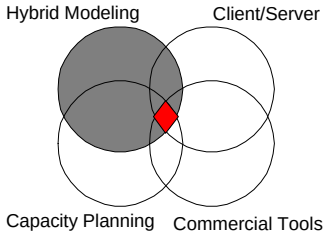
- Requires Understand the Hardware
  - How Processors Communicate
  - Completing a Unit of Work
  - Build a Sub-Model
  - Determine a Delay Value
- Level of Granularity
  - Too General or Too Detailed
- Generally Simulation Techniques

Hybrid Modeling

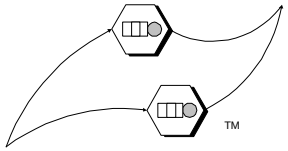




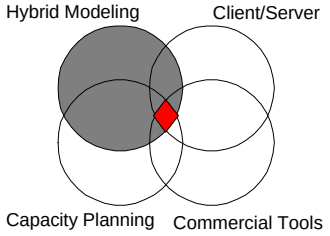
Hybrid Modeling



- Combination of Techniques
  - Simulation
  - Analytic Queuing Theory
- Benefits
  - Complexity Reduction
  - Performance Improvement
  - Analysis Flexibility

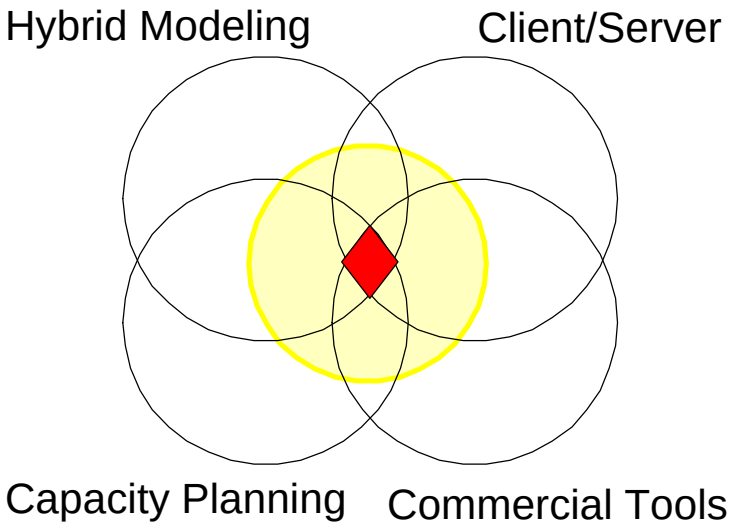


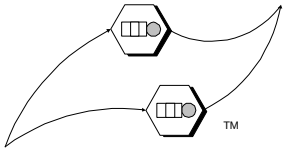
Hybrid Modeling



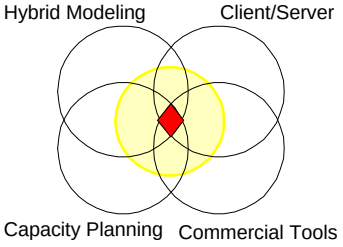
- Need for Enterprise Modeling
  - Bridge Across Existing Techniques
  - End User View of an Enterprise Level Application
- Implemented with Submodels
  - Takes Advantage of Existing Models and Tools

Simalytic Approach

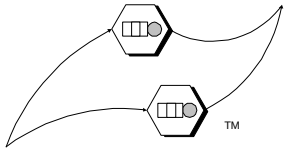




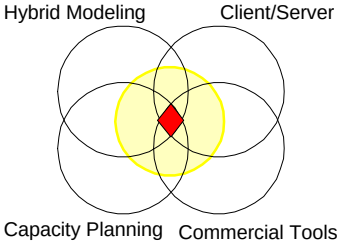
Simalytic Approach



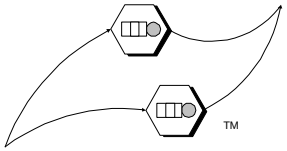
- "Simalytic" (**Sim**ulation/An**alytic**) Enterprise Modeling
  - Hybrid Modeling Technique
  - General Purpose Simulation Tool Framework
  - Analytic Modeling Tool Nodes
  - Existing Tools
  - Predict Capacity Requirements
  - Heterogeneous Computer Systems
  - Enterprise Level Application Model



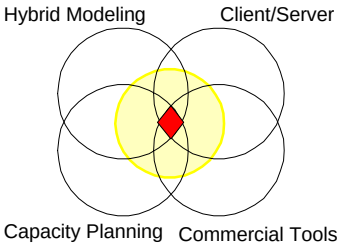
Simalytic Approach



- Key Differences - Simalytic vs Current
  - Applications at Enterprise Level
    - Mixes Different Tools and Techniques
  - Hardware at Node Level
    - Existing Tools and Techniques
    - Calculated Arrival Rate
  - Reduce the Time and Effort
    - Commercial Platform-Centric Tools
    - Existing Detailed Application Models
    - Reuse and Incorporate Existing Processes



Foundation



- Simulation Response Time Formula $T = \frac{\sum_{i=1}^{n_t} T_i}{n_t}$
  - Transform Function Using Queuing Theory Formula $T = \frac{S}{1 - lS}$
  - $f(l_i)$ Replaces T_i Server Time
    - Where T is the time, i is the Iteration and l_i is Based on Interarrival Times

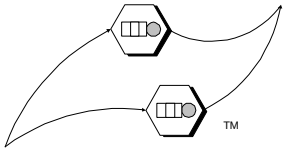
$$T = \frac{\sum_{i=1}^{n_t} f(l_i)}{n_t}$$

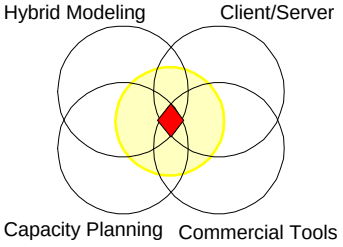
where l = arrivals per second as:

$$l_i = \frac{b}{c_i - c_{i-1}}$$

where c = simulation clock value

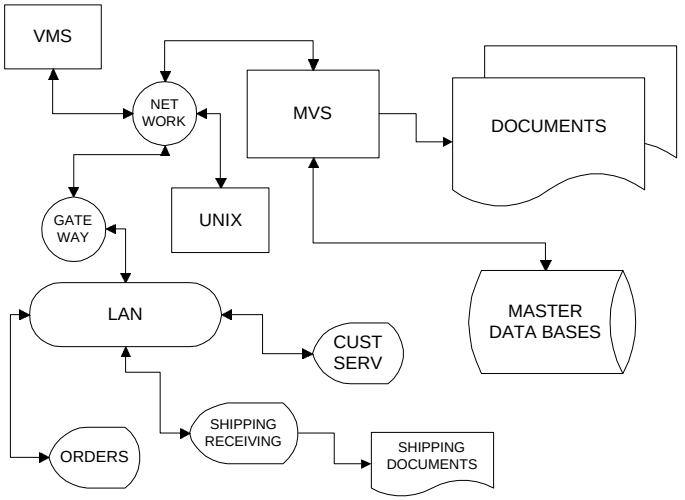
and b = simulation clock ticks per second

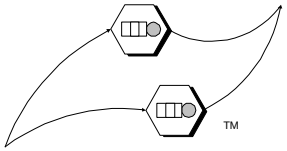


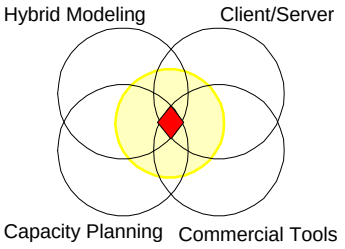


Simple Enterprise Model

- Order Entry
  - Queries Shipping
- Shipping Response Time:
  - 1 Second When Arrivals < 3 / Minute
  - 2 Seconds When Arrivals > 3 Minute

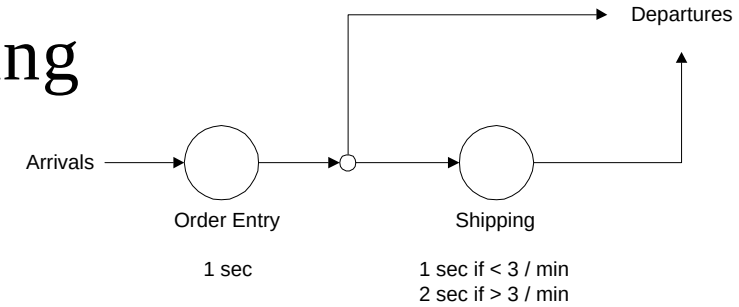


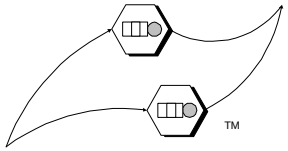




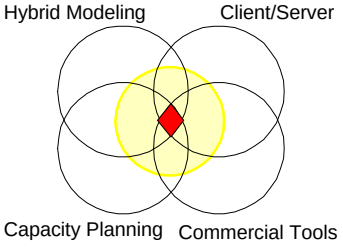
Simple Enterprise Model

- Model of Order Entry System Alone
  - Shipping Load Has No Effect
- Simalytic Model Adjusts Service Time
  - Long (Shipping) Transactions
- Simalytic Model of Order Entry Application
  - Other Loads Can Impact Response Times
  - Also Impacts to Shipping Application

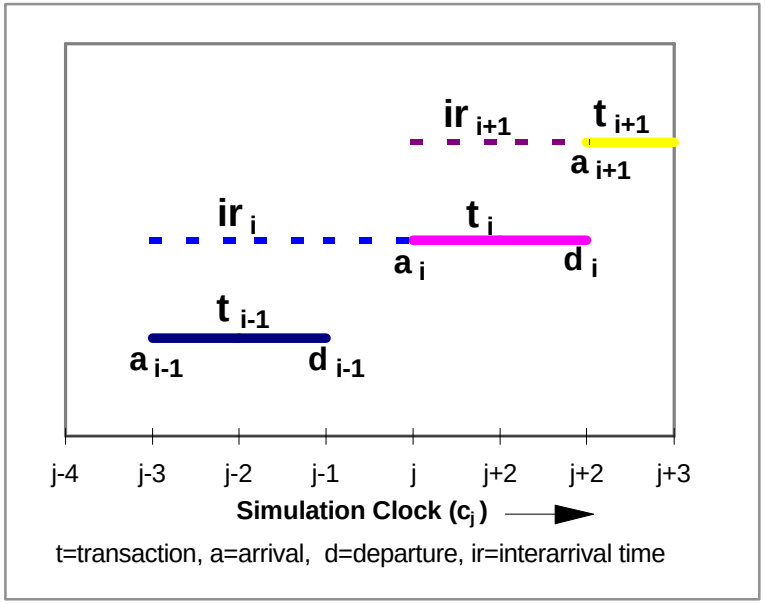


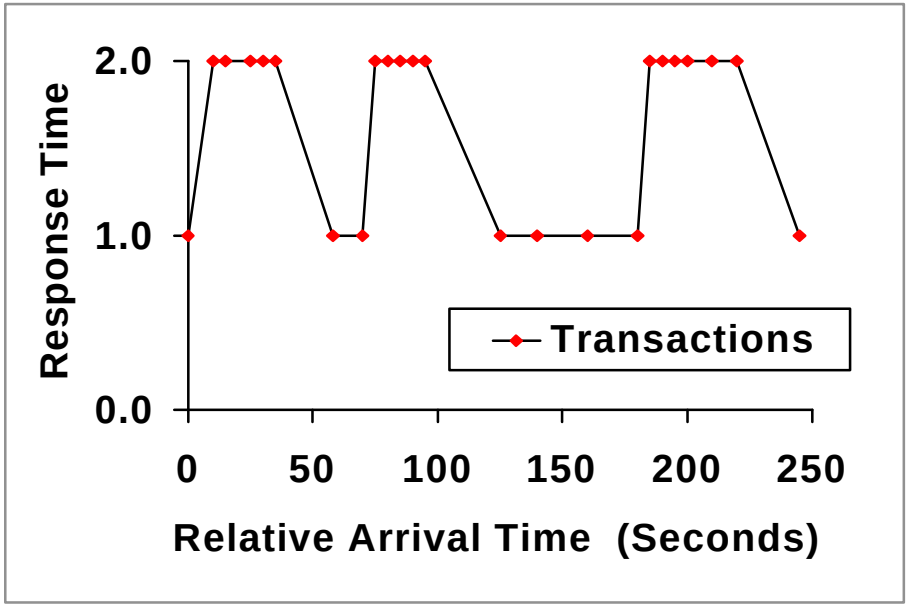


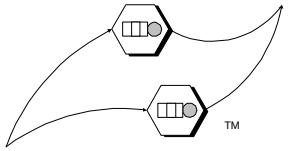
Load Dependent



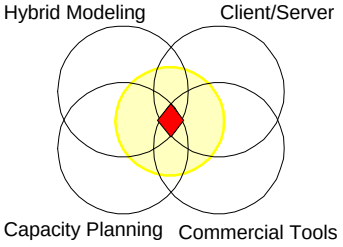
- Response Time
  - Calculated Based on Interarrival Time Between Each Pair of Transactions



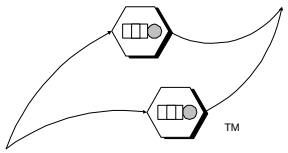




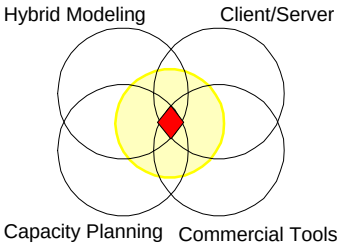
Model Analysis

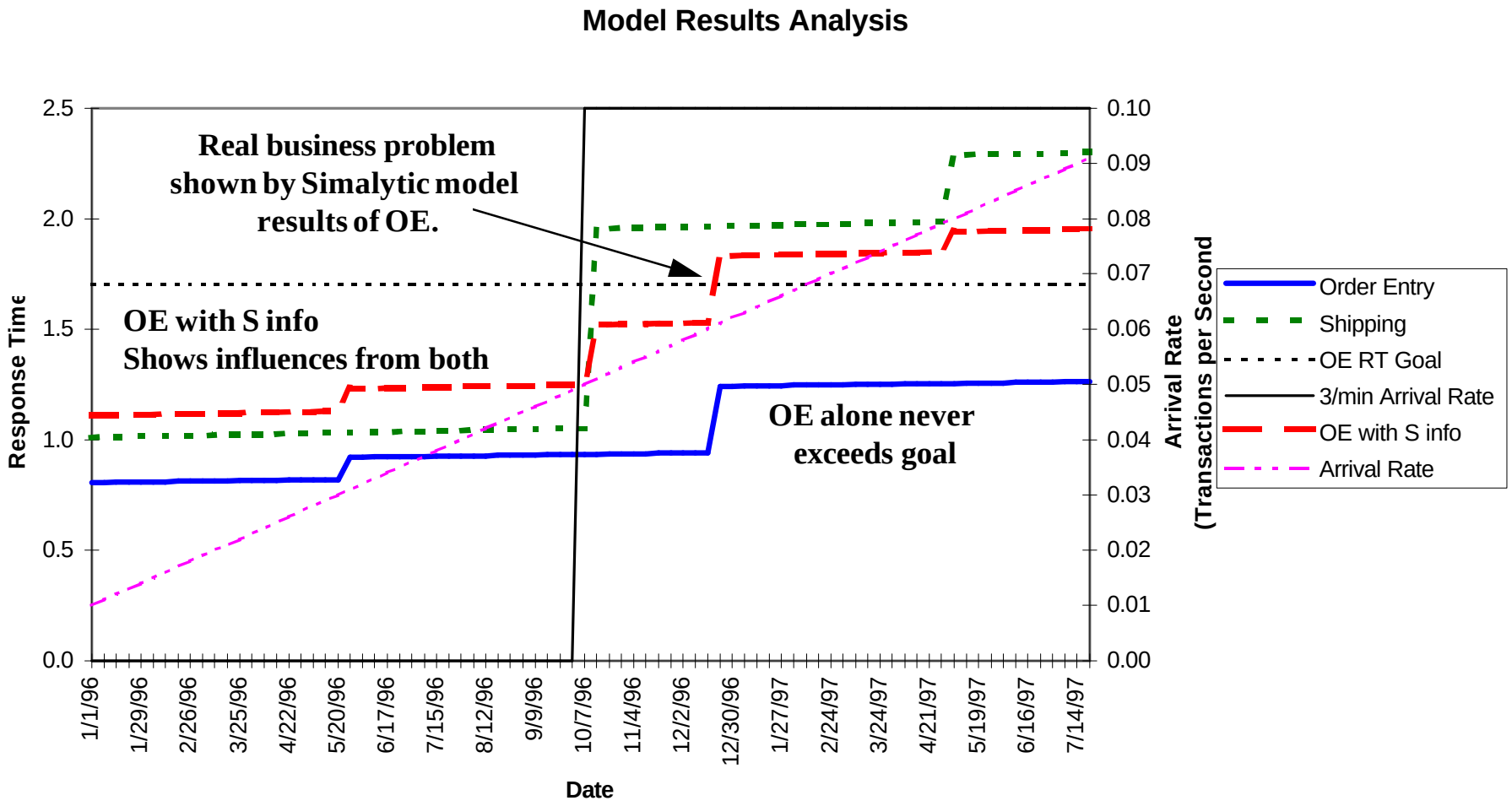


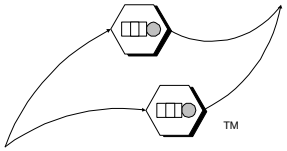
- Assumptions
  - Constant Arrival Rate Increase
  - 30% of OE Transactions Access Shipping
  - Order Entry Response Time Goal < 1.7 Seconds
  - Shipping Response Time Goal < 10 Seconds
  - Analysis Scope of 18 Months
- Analysis Objectives
  - When Does OE System Fail to Meet Goal?
  - What Will Fix the Problem?



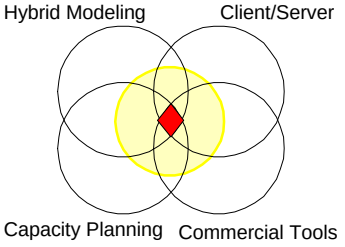
Model Analysis



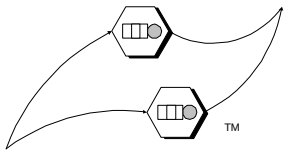




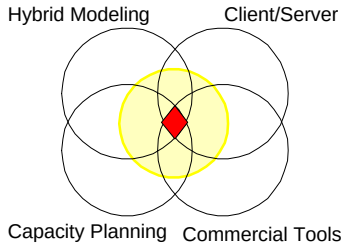
Model Analysis



- Independent Server Analysis
  - Shipping Problem Known and Acceptable
  - Order Entry Never Has a Problem
- Business Problem at Year End
  - Look for Seasonality in Arrival Rate
  - Shipping System Upgrade to Fix Order Entry Problem

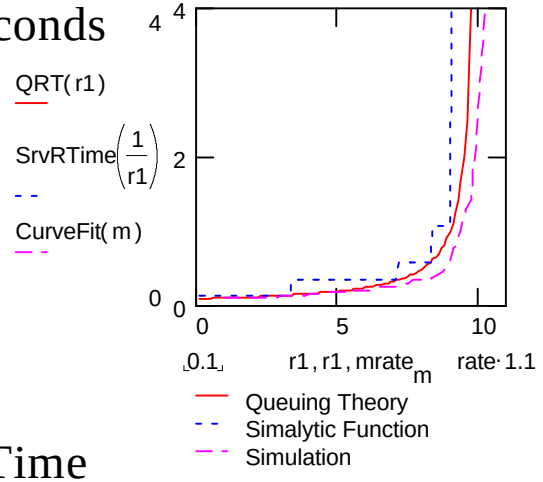


Response Time Comparison

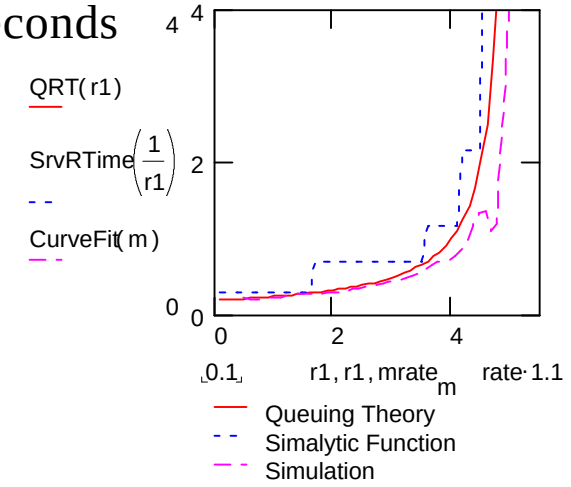


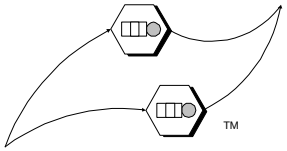
- Range of Arrival Rates for each Service Time
- Three Techniques
  - Simulation
  - Analytic Queuing Theory
  - Simalytic
- Similar Results

Service Time of .1 Seconds

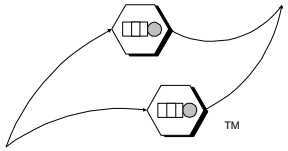


Service Time of .2 Seconds



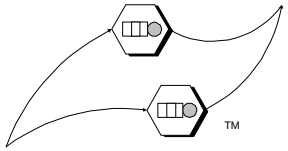


Conclusion



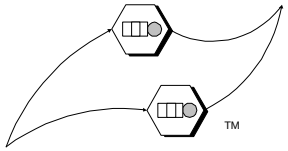
Conclusion

- Enterprise View of Applications
- Applications Relationships
  - Transaction Routing
  - Synchronization of Server Responses
  - Understanding and Documentation



Conclusion

- Combination of Techniques
  - Simulation and Analytic
  - Platform-Centric and General Purpose
- Predict Future Performance
- Client/Server Applications
- Reuse Existing Models
- Reduce Time and Effort



Questions

?