TPC-W Version 2

Benchmark Overview

Outline

- TPC-W V2
  - Need for TPC-W Version 2
  - Goals
  - Business Model
  - Architecture
  - Implementation Requirements
- Summary
  - V1 vs. V2
  - Next Steps

The need for TPC-W V2

- Input from members' marketing, press, analyst indicate strong:
  - Need for industry standard vendor neutral application server benchmark
  - Need for industry standard web services benchmark
  - Need to showcase commercially available application server products.
- Stress transactional features that go beyond traditional DBMS transactions
- Drive industry improvements in distributed transaction environments.

TPC-W V2 Goals

- Provide industry standard transactional application server benchmark.
- Limit scale of benchmark
  - Reduce cost of benchmarking
  - Focus on application server
- Realistic and up-to-date business process
  - Application runs in Managed Environment
  - 2 Phase Commit
  - Durable Message Queues
  - Web Service Interface
    - Adhering to Web Service Interoperability (WS-I) – BP1.0
- Incorporate key standard and emerging technologies
  - Managed runtime environment (Java and .NET)
  - SSL
  - RAID
  - Web Service technologies (e.g., SOAP/XML)
  - Distributed transaction management (2 phase commit)

TPCW-2 Resource Requirements

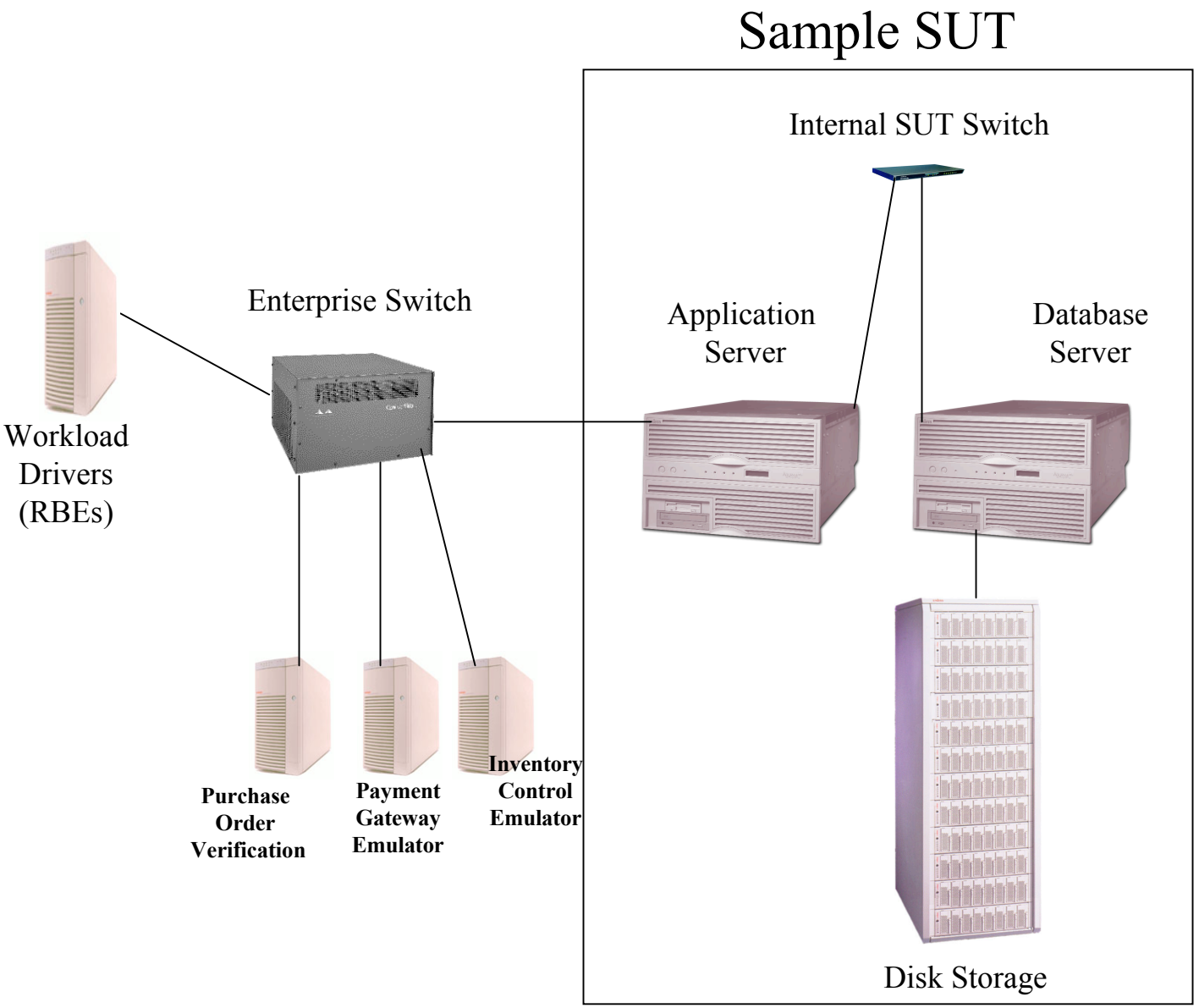
-
- At *most* 2 SUT systems required.
  - Much CHEAPER to benchmark
  - Fewer employee resources required
- Prototype data indicates low I/O requirements
- Application code is ‘managed code’ executing in a managed runtime environment
  - Easier to implement and execute
  - Much fewer L.O.C. to develop application

Mission Statement

The TPC-W Version 2 benchmark provides for the measurement of the performance and throughput of an application server providing web services to clients. An industry standard database product will be used to provide data storage and integrity of business information. The business model addressed by this benchmark is primarily focused on the business to business scenario.

The SUT will be driven by web service requests made through the industry standard and vendor neutral protocols of SOAP and XML. The web service business logic must execute in the context of a managed environment. Additionally, the benchmark incorporates external business processes involving durable message queues and two phase commit operations.

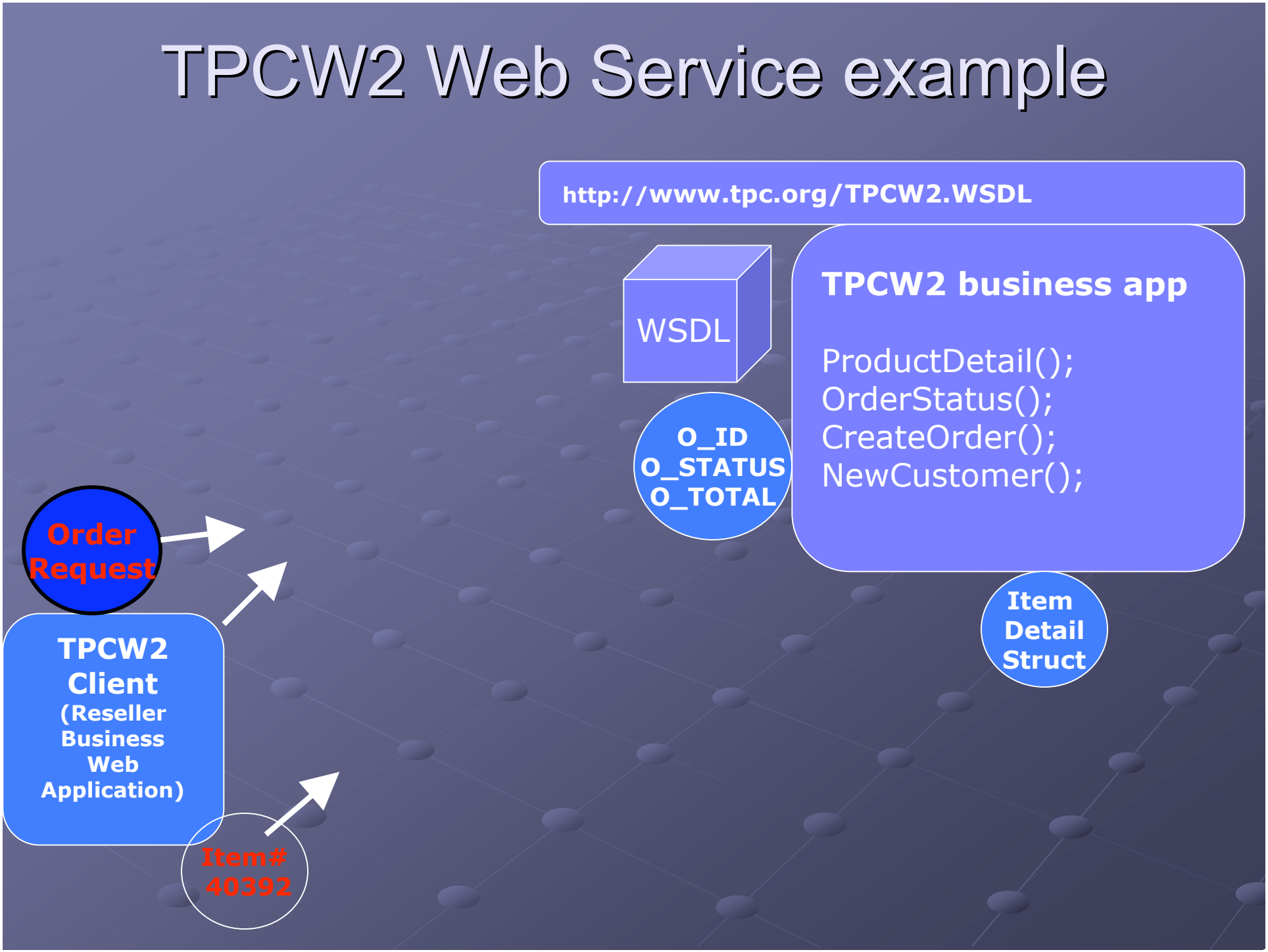
Architecture



Business Model

- Business to business operations of book sellers and distributors.
- Secure web services interface
  - retrieval of product information
  - product list updates
  - purchasing requests
  - customer account functions
  - administrative functions
- Interactions participate in external web service interactions
  - customer credit verification
  - stock fulfillment systems.
- Background processing
  - shipping processes
  - stock management
  - Both require reliable queues and distributed transaction management.

TPCW2 Web Service example



Web Service Interaction Characteristics

- Create Order (50%)
  - SSL/HTTP/SOAP/Database/Durable Queue
  - 50% require 2 second payment gateway
  - 100% require insert into ship queue
  - 2 phase commit
- Order Status (20%)
  - SSL/HTTP/SOAP/Database
- Product Detail (12%), New Products List (10%)
  - SSL/HTTP/SOAP/Database
  - Database interaction (possibly cacheable by Application Server)
- Change Payment Method (5%), New Customer (1%)
  - SSL/HTTP/SOAP/Database
  - 50% require 2 second purchase order gateway
- Change Item (2%)
  - SSL/HTTP/SOAP/Database
  - Administration task used to affect product detail/new products results

Background Processes

- Shipping Process
  - Asynchronous from interactions
  - Read from durable ship queue
  - Update order to shipped (database)
  - 10% write to backorder queue
  - 2 phase commit (with DBMS)
- Stock Management Process
  - Asynchronous from interactions
  - Read from durable backorder queue
  - 2 second interaction w/ external supplier (ICE)
  - 2 phase commit (no database)

Implementation Requirements

- Application Servers
  - Managed Environment (i.e., JRE or .NET)
  - Adherence to WS-I (BP 1.0) for Web Service functions
  - Security (SSL)
- Message Queuing Product (not DBMS)
  - Transactional (two phase commit)
  - Durable
- DBMS
  - Used for data storage and integrity
  - Participates in two phase commit

Definition: Managed Environment

- A software abstraction layer that sits between application code and the operating system. It provides a logical runtime environment that insulates the application from the native operating system.
- Included Features (Non-Exhaustive):
  - Automatic memory management and garbage collection
  - Code verification
  - Code access security
  - Translation of an intermediate language generated by a compiler into native machine code
  - Program loading
  - Thread creation and scheduling. Threads may be created through API's provided by the runtime libraries that are part of the managed environment.
  - Data type checking to ensure that data types are used correctly and consistently.
- Execution Environment
  - Must be compliant with one of the following specifications
    - JRE 1.3 or greater as published by Sun Microsystems
    - ECMA-335 as published by the European Computer Manufacturers Association
  - May be a superset of the specification and contain vendor specific extensions and enhancements.

Definition: Application Server

- A software layer that provides application infrastructure functions and services. It is positioned between the user making a request for information and the underlying business process that satisfies the request.

- Included Features (Non-Exhaustive):
  - Managed environment for application execution
  - Web Services (WS-I BP 1.0 Adherent)
  - Network transparency
  - Transaction management
  - Security
  - Multi-threading
  - Database connectivity
  - Resource pooling
  - Application state management

- The application server may be an integrated part of the operating system or a separate product that is procured from another source or sources. It is permissible to run more than one application server or multiple instances of a given application server. Application servers and application server components from different sources may be used to perform selected functions within the SUT.

- However, the application server(s) must execute on the same SYSTEM. The application server may execute multiple instances of its managed environment. Within each instance of a managed environment, one or more instances of the application program may execute.

Outline

- TPC-W V2
  - Need for TPC-W Version 2
  - Goals
  - Business Model
  - Architecture
  - Implementation Requirements
- Summary
  - V1 vs. V2
  - Next Steps

V1 vs. V2

- Fundamental shift from TPC-W v1
  - WAS E-Commerce Web Storefront
  - IS B2B Web Services Workload
  - WAS Infinite scale out to clusters
  - IS Limited to two nodes (App Server and Database)
  - WAS Optimized hand written static code
  - IS 100% managed (Java and .NET).
  - WAS Relaxed on caching and offloading opportunities (hand opt)
  - IS Tight on caching and offloading opportunities (Application server)
  - WAS ~30% SSL
  - IS 100% SSL
  - WAS Showcase for image caches, web caches, web servers
  - IS Showcase for commercial application server environment

Next Steps

- Specification is in company and public review phase until Feb 2004.
  - Please review specification and submit comments.
    - Comments can be submitted using online forms at the TPC website: www.tpc.org
- Specification is being actively prototyped
  - 2 java prototypes in progress
  - 2 .NET prototypes in progress.

END

Of Presentation

Backup

- Scope

  http://www.tpc.org/information/about/abouttpc.asp