

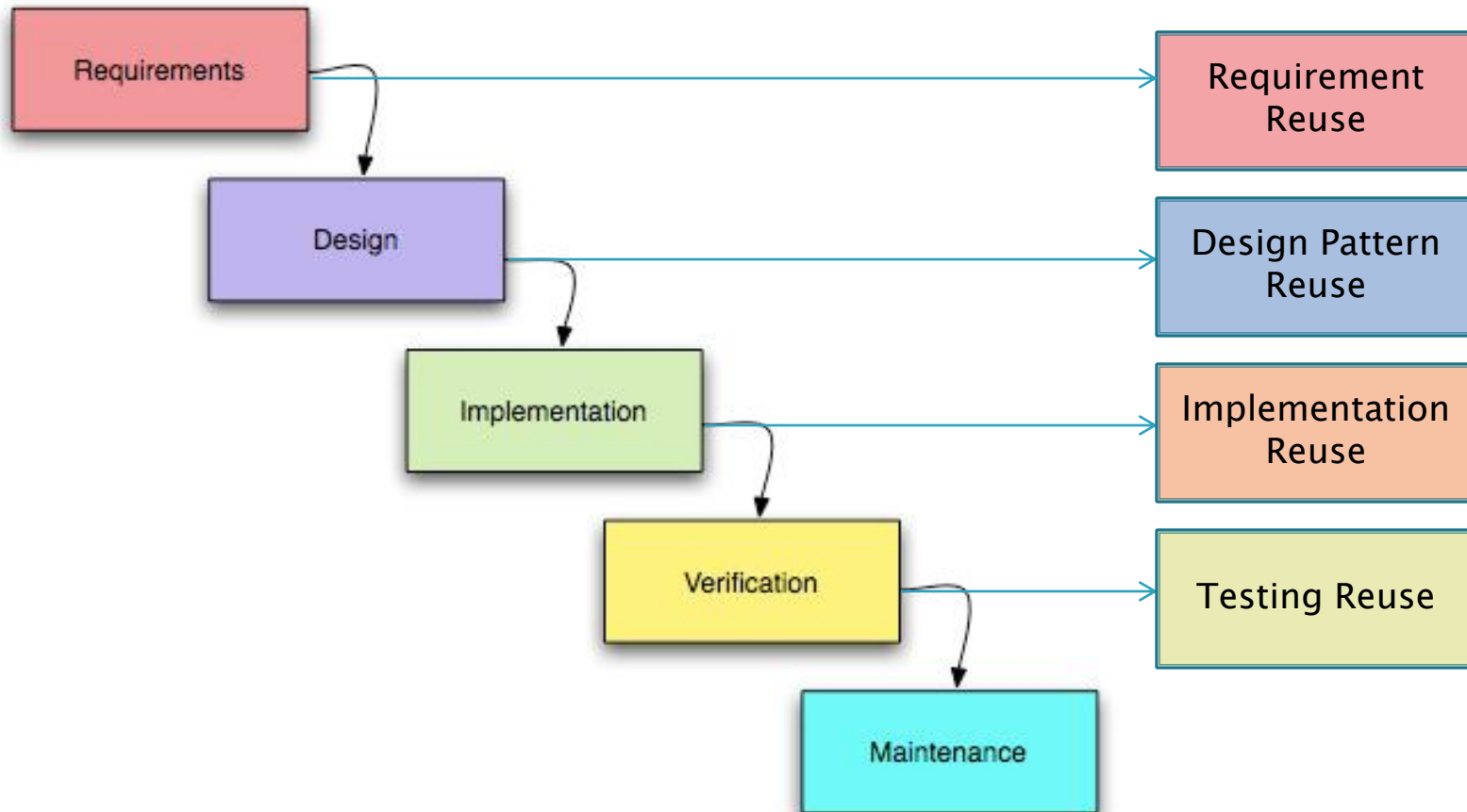


Module 5

Software Testing Reuse (Part 1)

Dr. Shiping Chen

Where are We?



Part of this figure is from: http://en.wikipedia.org/wiki/File:Waterfall_model.png

Outline

- ▶ Overview of Software Testing
- ▶ Reusable testing artifacts
 - Test case reuse
 - Test script reuse
 - Test suits reuse
- ▶ Reusable testing benchmarks TPC-C

What is Software Testing?

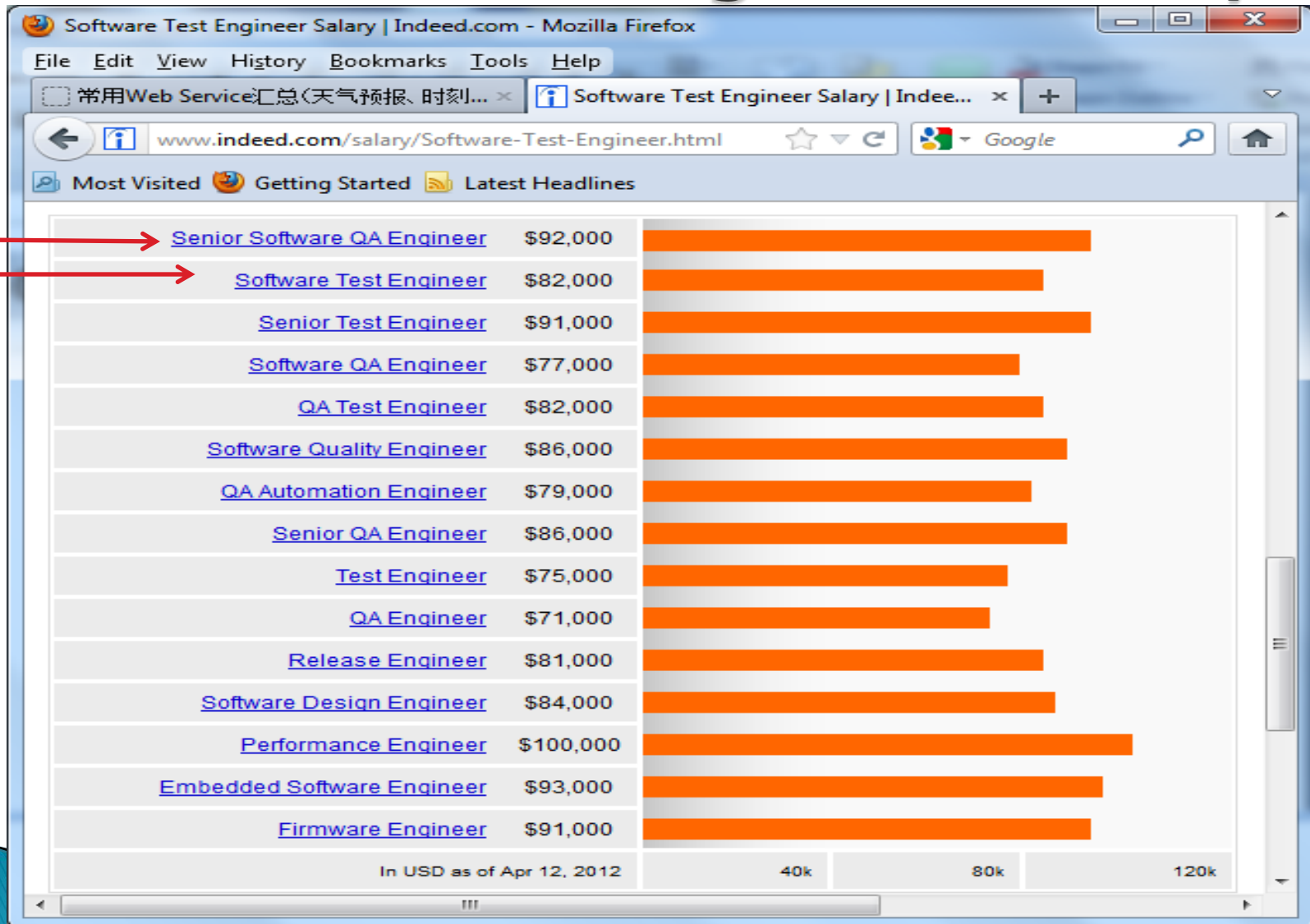
- ▶ Software testing is one of activities during SDLC, aimed at evaluating correctness and/or capabilities of a software system to provide some degree of Quality Assurance of the software.

Revised from: http://www.ece.cmu.edu/~koopman/des_s99/sw_testing/

Evolution of Software Testing

- ▶ The primary function of software testing is to detect bugs in the software;
- ▶ ...also measure the software system capabilities, such as performance and security etc.
- ▶ Its role becomes increasing as software systems get big and complicated;

Software Test Engineer Salary



Testing Methods

White-box Testing	Black-Box Testing	Gray-Box Testing
The tester has access to the internal data structures and algorithms including the code	Has no knowledge of internal implementation	Black-Box Testing with some knowledge of internal implementation
<ul style="list-style-type: none">• API Testing• Code coverage testing• Fault injection testing	Specification-based testing	Integration testing
Used by the developers	Used by the dedicated testing engineers	Used by integration or deployment engineers

Testing Levels – Size/Scope

Unit Testing	(Initial) Integration Testing	System (Integration) Testing
Also known as <i>component testing</i> , refers to tests that verify the functionalities of a specific section of code	Seek to verify the interfaces between components against a software design.	A completely integration test to verify if the system meets its requirements as a whole
At developing time	At initial integration time	At final integration time
By developers	By developers and/or testing engineers	Testing engineers

Testing Levels – Objectives

Regression Testing	Acceptance Testing	Alpha Testing	Beta Testing
Continuous tests to focus on finding defects	Also called 'smoke' testing – check basic functionality	Simulated or actual operational testing	considered a form of external User acceptance testing
After a major code reversion	After internal integration testing	At late stage	After Alpha testing and before formal release
Testing Engineers	With customers	potential users/customers	potential users/customers

Non-Functional Testing

- ▶ Performance testing
- ▶ Security Testing
- ▶ Usability testing
- ▶ Internationalization and localization
- ▶ Destructive testing, also called robustness testing

Typical Testing Process

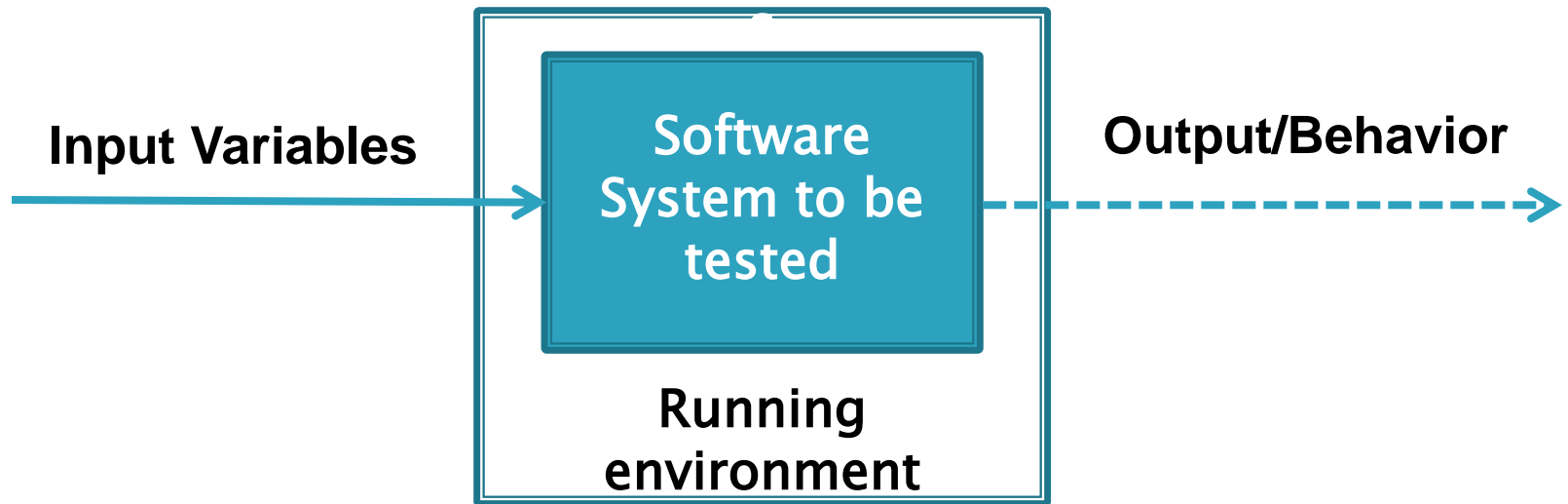
- ▶ **Test planning/preparation:**
 - Test strategy, testbed & resources, including Hardware, Software, HR, etc.
- ▶ **Test development:** Test procedures, test scenarios, test cases, test datasets, test scripts
- ▶ **Test execution:** Testers execute the software based on the plans and test documents
- ▶ **Test reporting:**
 - Report errors and/or performance found during testing to the development team

Testing Artifacts for Reuse

- ▶ Test cases
- ▶ Test scripts
- ▶ Test suites (benchmark)
- ▶ Test harness
 - a collection of software, tools, samples of data input and output, and configurations

Test Cases

- ▶ A set of *conditions or variables* under which a tester will *determine* whether an application or software system is *working correctly or not*



A Typical Test Case Format

- ▶ test case ID
- ▶ test case description
- ▶ test step or order of execution number
- ▶ related requirement(s)
- ▶ Depth/level
- ▶ test category
- ▶ author
- ▶ check boxes for whether the test is automatable and has been automated.
- ▶ Expected Result and Actual Result.
- ▶ Additional fields that may be included and completed when the tests are executed:
 - ▶ pass/fail
 - ▶ remarks

An Test Case Example for Login

Login Test Case Examples.pdf - Adobe Reader

File Edit View Window Help

1 / 5 81.1%

Comment Sign

Test case name	test case desc	test steps			test case status	test status (P/F)	test priority	defect severity
		step	expected	actual				
Validate Login	To verify that Login name on login page must be greater than 3 characters	enter login name less than 3 chars (say a) and password and click Submit button	an error message "Login not less than 3 characters" must be displayed		design		high	
		enter login name less than 3 chars (say ab) and password and click Submit button	an error message "Login not less than 3 characters" must be displayed		design		high	
		enter login name 3 chars (say abc) and password and click Submit button	Login success full or an error message "Invalid Login or Password" must be displayed		design		high	

From: <http://www.vietnamesetestingboard.org/zbxe>

Test Scripts (Test Automation)

- ▶ **A set of instructions that perform the test cases on the system under test.**
- ▶ Usually for automatic testing, e.g.,
 - Regression testing.
- ▶ Usually written in script languages (Unix scripts, Perl, Tcl, PHP etc), programming languages (Java & C# & VB); and
- ▶ Vendor-specific languages for their tools, such as:
 - HP QuickTest Professional
 - Borland SilkTest
 - IBM Rational Robot

Test Suits

- ▶ *A collection of test cases ready to be reused* to test a specific class of software systems, such as performance test suits:
 - Database Benchmark: TPC-C
 - CPU Benchmark: SPEC CPU2006
 - High Performance Computing: SPEC MPI2007
 - Details refer to: <http://www.spec.org>

What is TPC?

- ▶ TPC = Transaction Processing Performance Council
- ▶ Founded in Aug/88 by Omri Serlin and 8 vendors
- ▶ Industry standards body for OLTP performance
- ▶ More information about TPC-C specifications and results are on <http://www.tpc.org>

TPC-C Overview

- ▶ A moderately complex OLTP (Online Transaction Processing) Application
- ▶ TPC-C models a wholesale supplier managing orders
- ▶ Provides a conceptual model for a **typical class of OLTP systems**
- ▶ Workload consists of **5 transaction types**

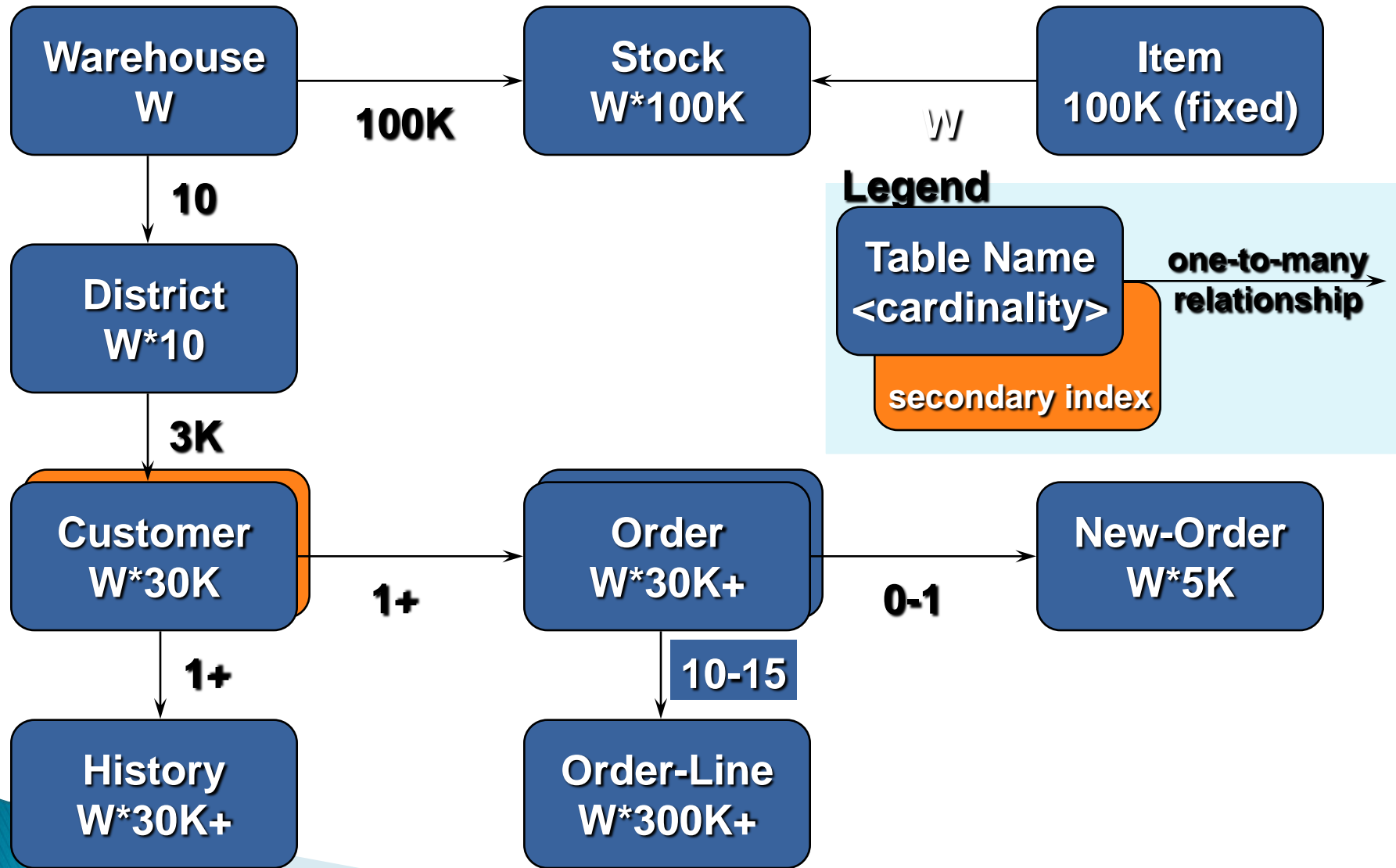
TPC-C's Five Transactions

- ▶ **Stock-level**: monitor warehouse inventory
- ▶ **New-order**: enter a new order from a customer
- ▶ **Order-status**: retrieve status of customer's most recent order
- ▶ **Payment**: update customer balance to reflect a payment
- ▶ **Delivery**: deliver orders (done as a batch transaction)

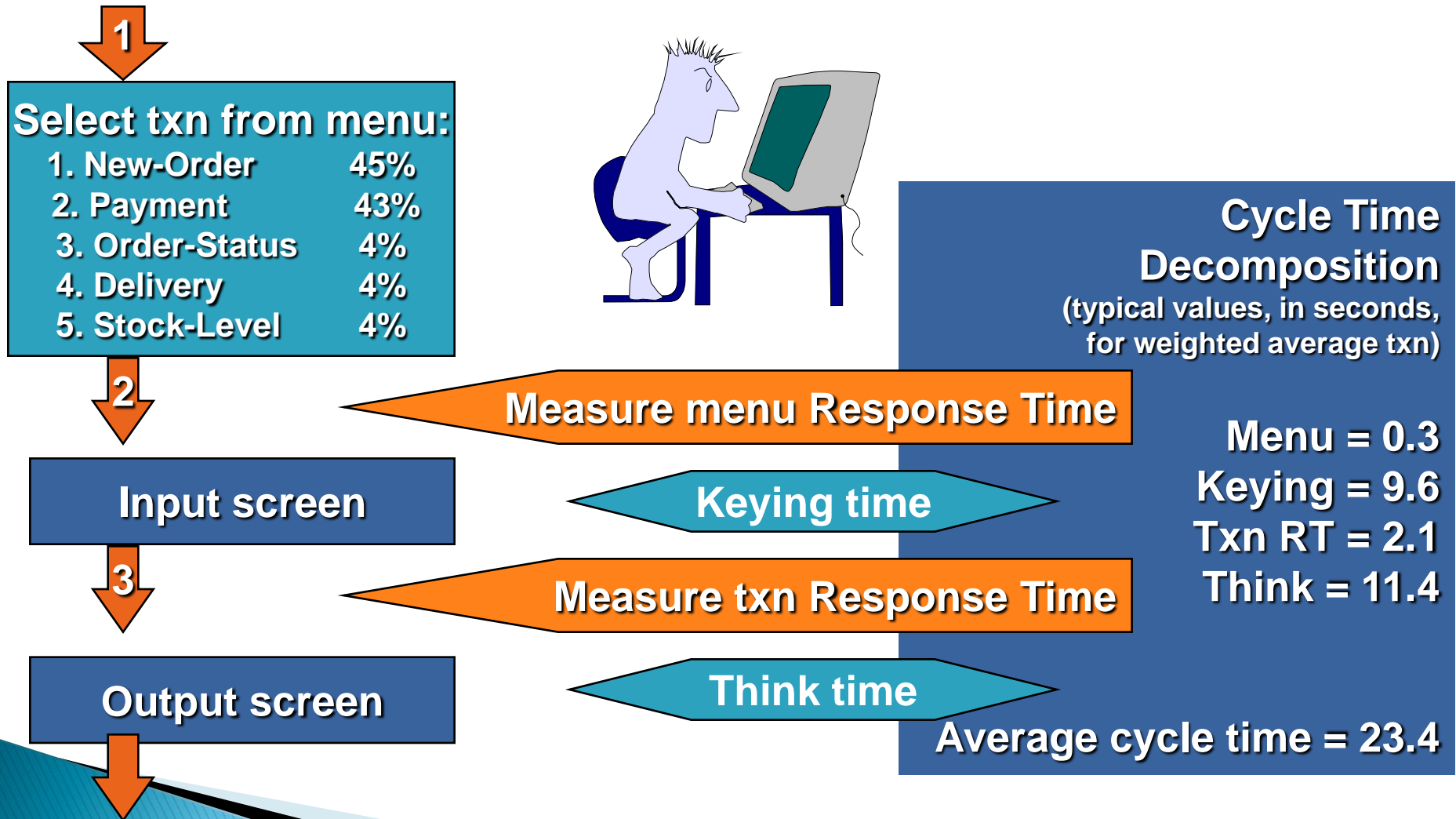
TPC-C Measurement Rules

- ▶ Transactions operate against a database of 9 tables.
- ▶ Transactions do *update, insert, delete, and abort*;
- ▶ Standard DB design Primary & secondary keys
- ▶ Response time requirement: 90% of each type of transaction must have a response time ≤ 5 seconds, except stock-level which is ≤ 20 seconds.
- ▶ Metrics:
 - Order txn rate (tpmC)
 - Price/performance (\$/tpmC)

TPC-C Database Schema



TPC-C Workflow



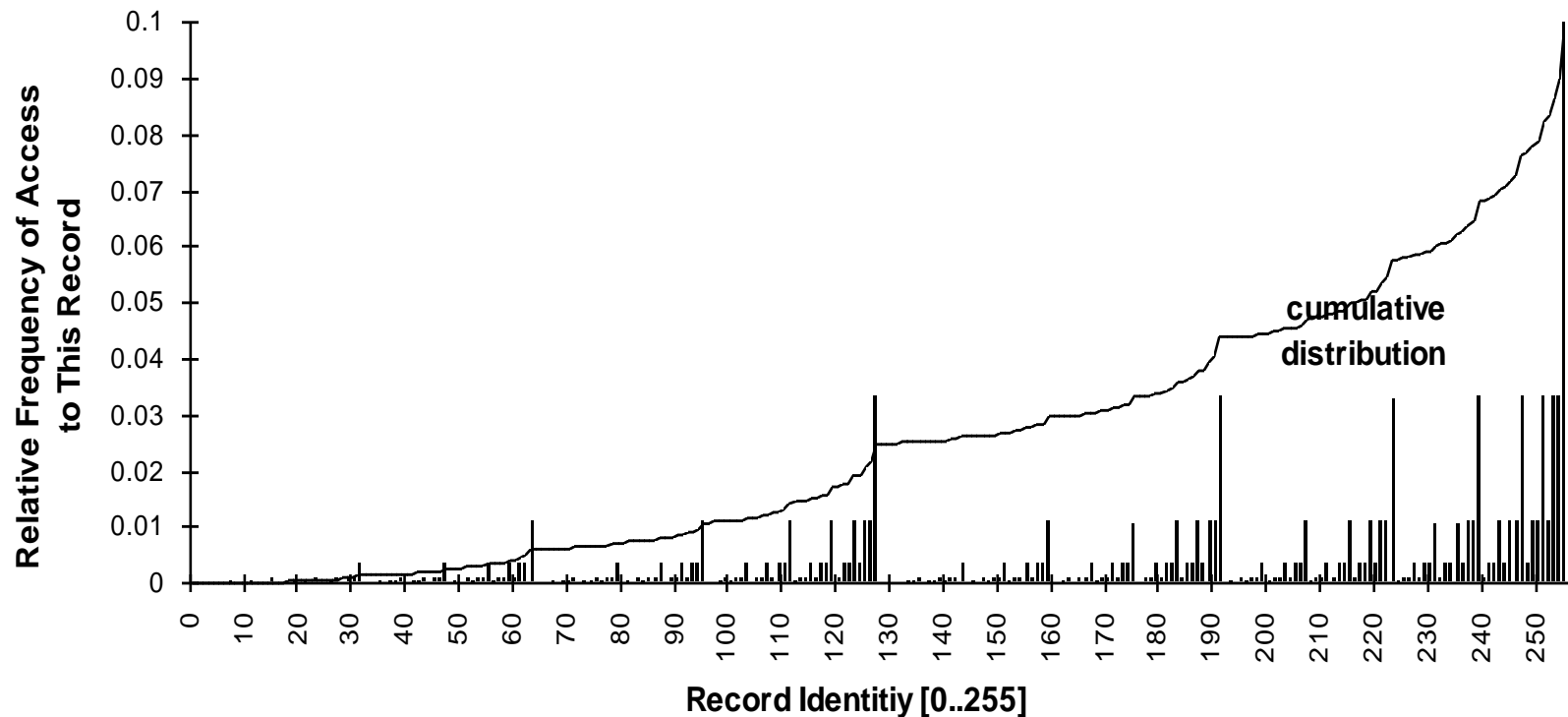
Testing Data Generation

► NURand – Non Uniform Random

- $\text{NURand}(A, x, y) = (((\text{random}(0, A) \mid \text{random}(x, y)) + C) \% (y - x + 1)) + x$
 - Customer Last Name: $\text{NURand}(255, 0, 999)$
 - Customer ID: $\text{NURand}(1023, 1, 3000)$
 - Item ID: $\text{NURand}(8191, 1, 100000)$
- bitwise OR of two random values
- skews distribution toward values with more bits on
 - 75% chance that a given bit is one ($1 - \frac{1}{2} * \frac{1}{2}$)
- skewed data pattern repeats with period of smaller random number

NURand Distribution

TPC-C NURand function: frequency vs 0...255

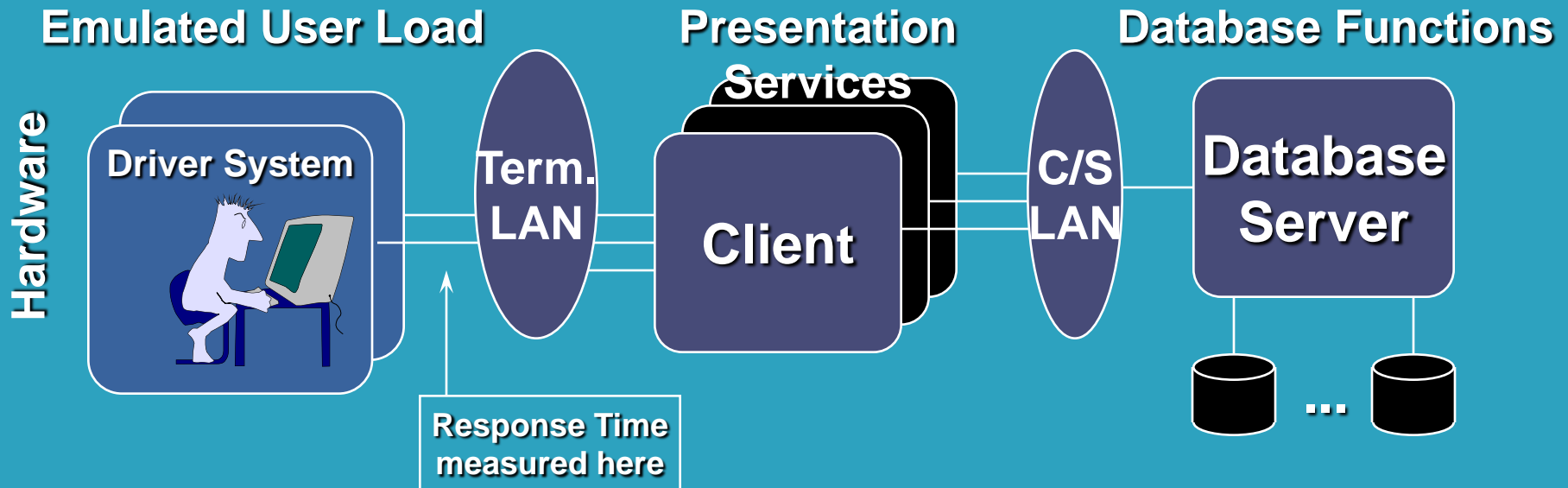


ACID Testing

- ▶ **Atomicity:** Verify that all changes within a transaction commit or abort
- ▶ **Consistency:** from one state to another, nothing between the two states
- ▶ **Isolation**
 - ANSI Repeatable reads for all but Stock-Level transactions.
 - Committed reads for Stock-Level.
- ▶ **Durability**
 - Must demonstrate recovery from
 - Loss of power
 - Loss of memory
 - Loss of media (e.g., disk crash)

Typical TPC-C Configuration

(Conceptual)



Software

RTE, e.g.:
LoadRunner

TPC-C application +
Txn Monitor and/or
database RPC library
e.g., JDBC & ODBC

TPC-C application
(stored procedures) +
Database engine +
Txn Monitor
e.g., SQL Server

TPC-C Rules of Thumb

- ▶ 1.2 tpmC per User/terminal (maximum)
- ▶ 10 terminals per warehouse (fixed)
- ▶ 65–70 MB/tpmC priced disk capacity (minimum)
- ▶ ~ 0.5 physical IOs/sec/tpmC (typical)
- ▶ 250–700 KB main memory/tpmC (how much \$ do you have?)
- ▶ So use rules of thumb to size 10,000 tpmC system:
 - How many terminals?
 - How many warehouses?
 - How much memory?
 - How much disk capacity?

$$» 8340 = 10000 / 1.2$$

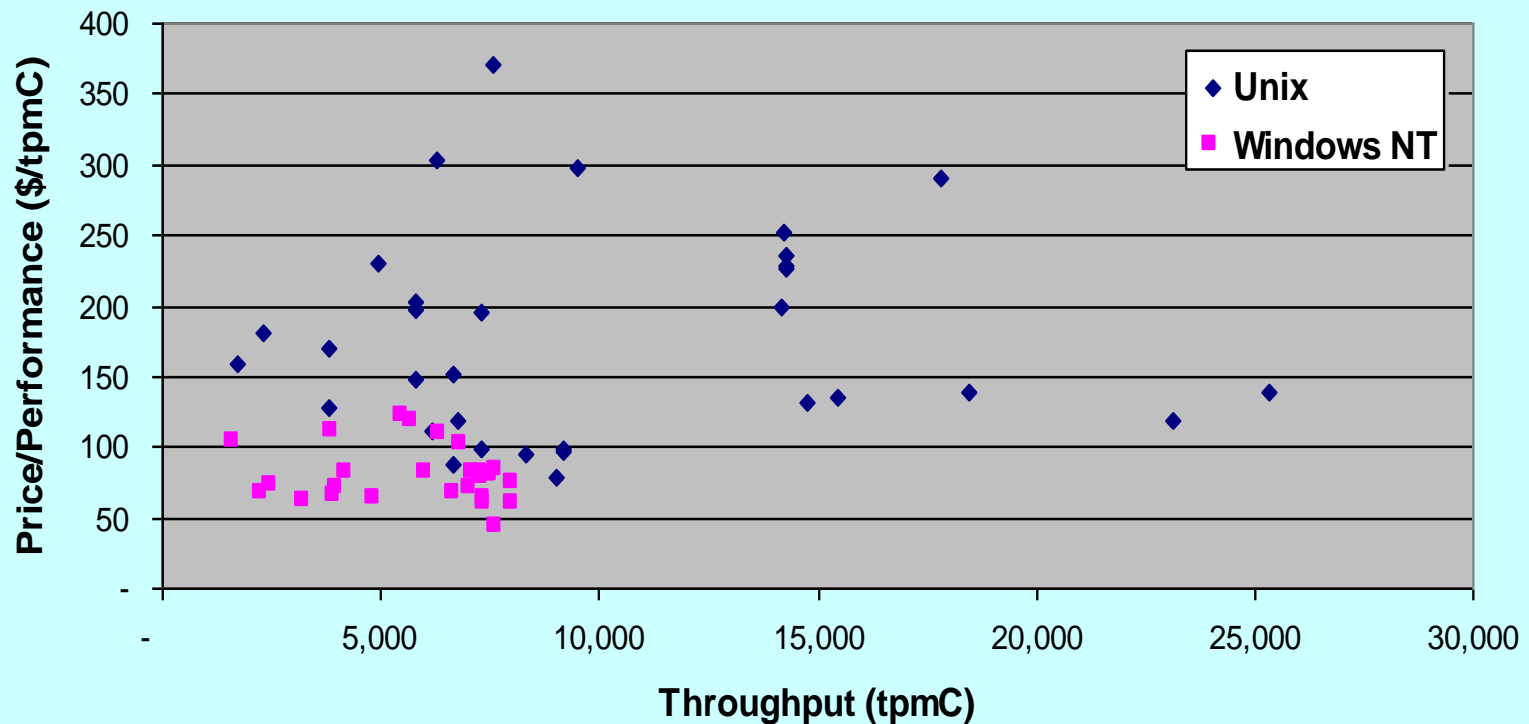
$$» 834 = 8340 / 10$$

$$» 2.5 - 7 \text{ GB}$$

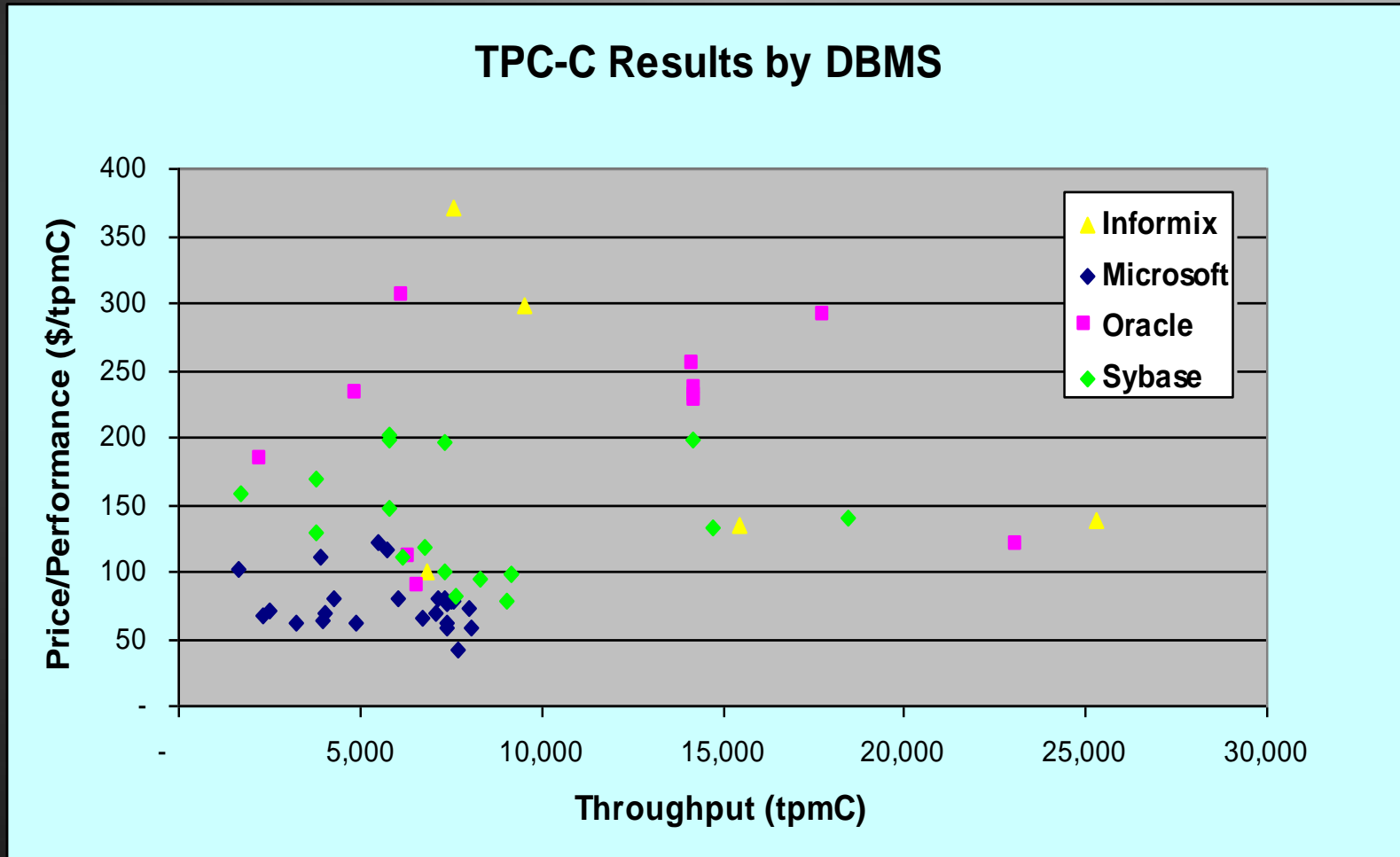
$$» 650 \text{ GB} = 10000 * 65$$

TPC-C Results (by OS)

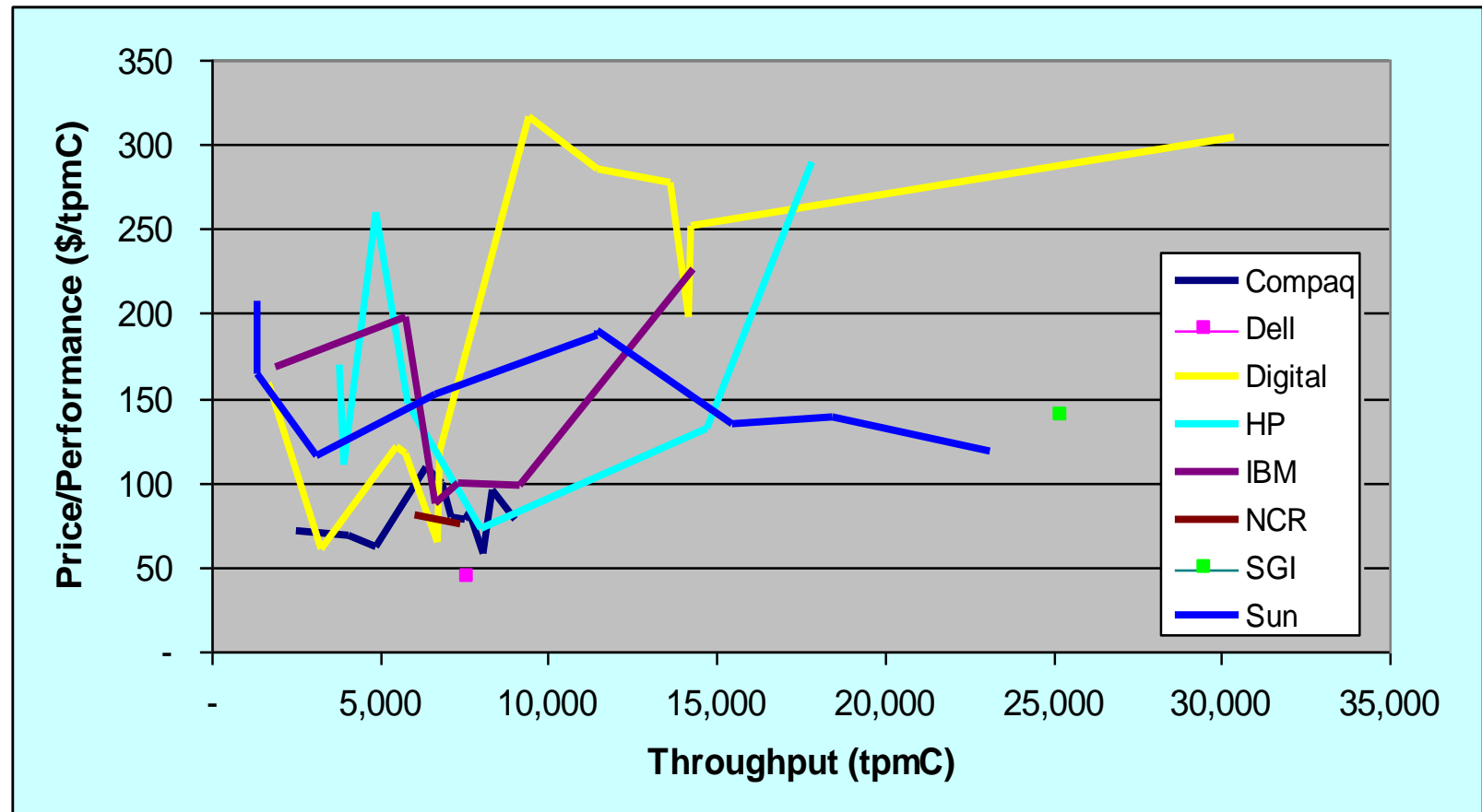
TPC-C Results by OS



TPC-C Results (by DBMS)



TPC-C for Computer Systems



TPC-C Summary

- ▶ Balanced, representative OLTP mix
 - Five transaction types
 - Database intensive; substantial IO and cache load
 - Scalable workload
 - Complex data: data attributes, size, skew
- ▶ Requires Transparency and ACID
- ▶ Full screen presentation services
- ▶ De facto standard for OLTP performance
- ▶ A very good candidate for performance testing reuse