Performance and Scalability

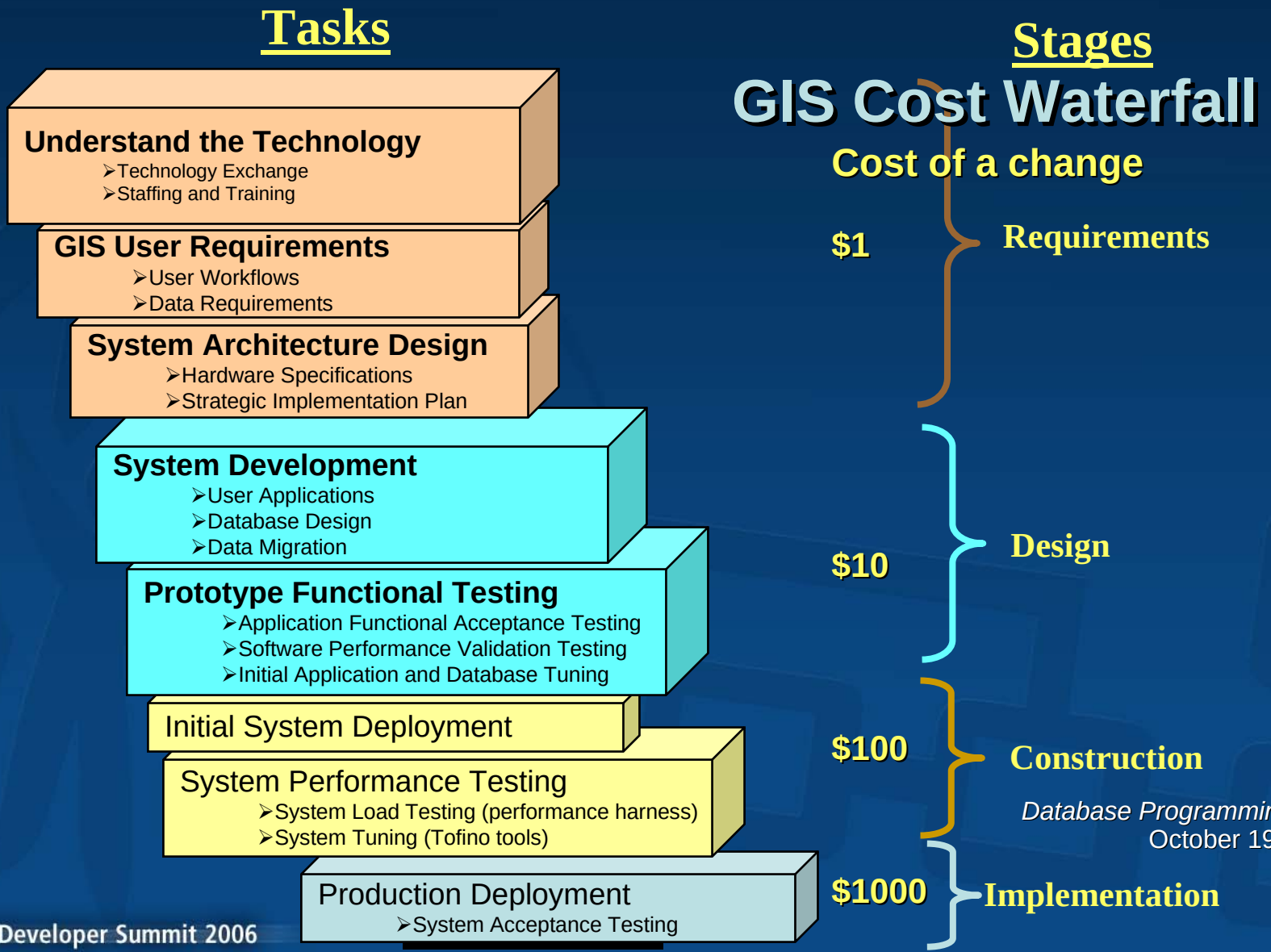
Dave Peters – System Architecture Design
Jeff DeWeese – Performance Validation Testing
John Meza – Performance Testing and Tuning

System Architecture Design

Dave Peters

Manage for Success

How can we manage cost?



(Moody)
Database Programming & Design
October 1996, p57–64

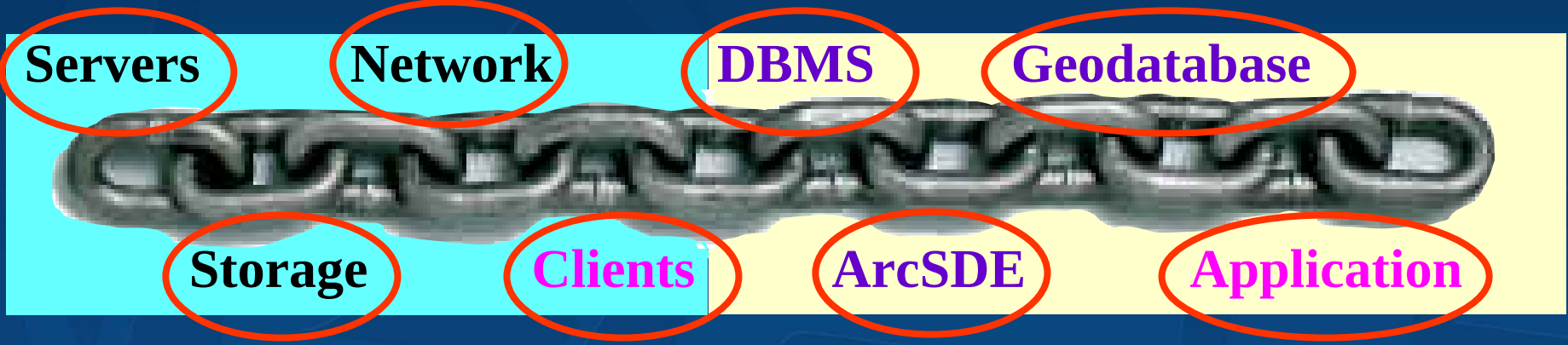
What is System Architecture Design?

Why is System Architecture Design Important?

Reduce Cost

Improve Productivity

Balanced System Design



Hardware Infrastructure

Database Design

User Workflow

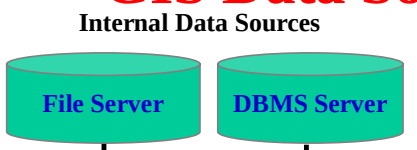
System Architecture Design

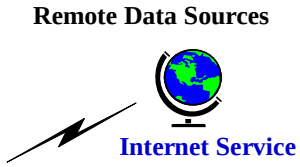
Framework for Productive Operations

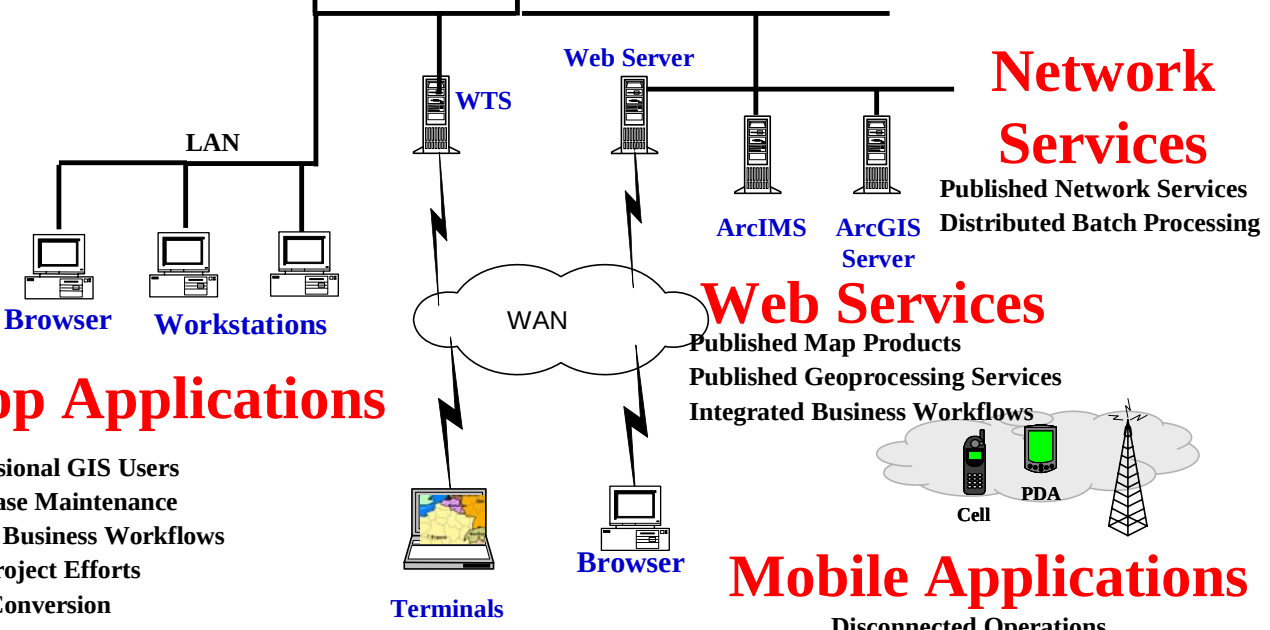
Use the right technology.

Software Technology Selection

GIS Data Source







Network Services

Published Network Services
Distributed Batch Processing

Web Services

Published Map Products
Published Geoprocessing Services
Integrated Business Workflows

Desktop Applications

Professional GIS Users
Database Maintenance
Heavy Business Workflows
GIS Project Efforts
Data Conversion

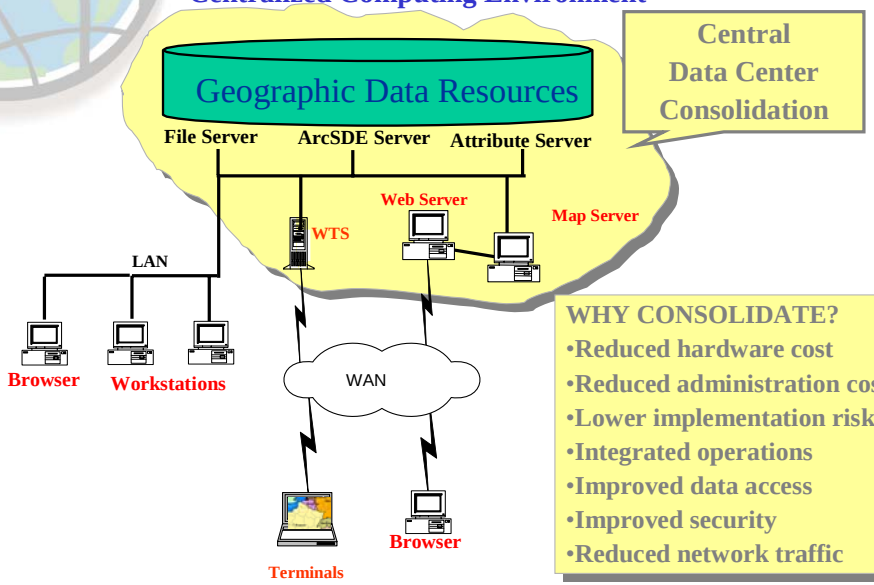
Mobile Applications

Disconnected Operations
Loosely Connected Workflows

Select the right architecture strategy

Database Configuration Options

Centralized Computing Environment



WHY CONSOLIDATE?

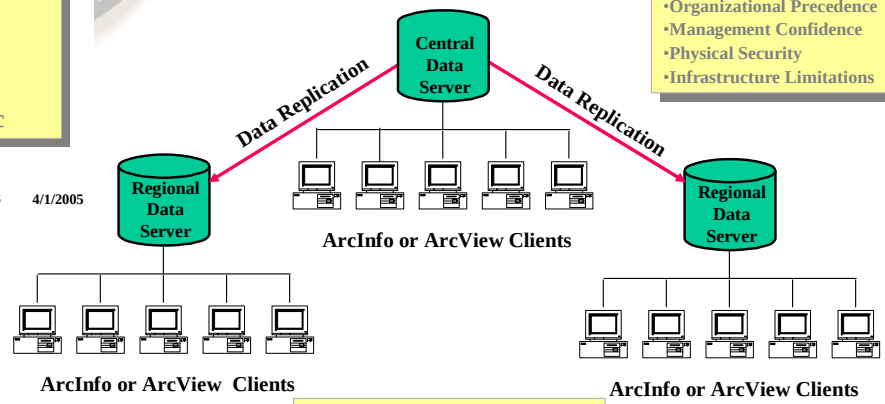
- Reduced hardware cost
- Reduced administration cost
- Lower implementation risk
- Integrated operations
- Improved data access
- Improved security
- Reduced network traffic

Database Configuration Options

Distributed Computing Environment

Why use distributed architecture?

- Organizational Precedence
- Management Confidence
- Physical Security
- Infrastructure Limitations

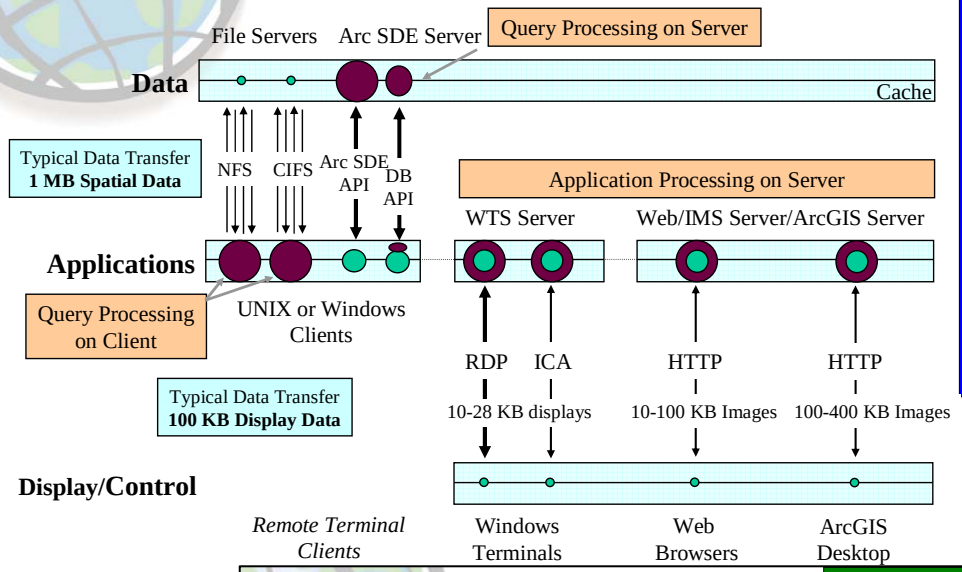


DISTRIBUTED DATA SOLUTIONS

- Increased hardware cost
- Increased administration cost
- Higher implementation risk
- Disconnected operations
- Limited data access
- Reduced security
- Increased network traffic

Understand Network Communications

Client/Server Protocols



Network Design Guidelines

Local Area Networks Bandwidth	Concurrent Client Loads File Servers	ArcSDE	Windows Terminals	Web Products
10 Mbps LAN	2–4	10–20	350–700	150–300
16 Mbps LAN	3–6	16–32	550–1100	250–500
100 Mbps LAN	20–40	100–200	3,500–7,000	1,500–3,000
1 Gbps LAN	200–400	1,000–2,000	35,000–70,000	15,000–30,000
Wide Area Networks Bandwidth	**Concurrent Client Loads File Servers**	**ArcSDE**	**Windows Terminals**	**Web Products**
56 Kbps Modem	NR	NR	2–4	1–2
128 Kbps ISDN	NR	NR	5–10	2–4
256 Kbps DSL	NR	NR	10–20	5–10
512 Kbps	NR	NR	20–40	10–20
1.54 Mbps T-1	NR	1–2	50–100	25–50
2 Mbps E-1	NR	1–3	75–150	40–80
6.16 Mbps T-2	1–2	6–12	200–400	100–200
45 Mbps T-3	10–20	50–100	1,500–3,000	700–1500
155 Mbps ATM	30–60	150–300	5,000–10,000	2,500–5,000

Client/Server Performance

Client/Server Communications	Network Traffic Transport Time 56 Kbps	1.54 Mbps	10 Mbps	100 Mbps	1 Gbps
File Server to Workstation Client (NFS) •1 MB => 10 Mb + 40 Mb = 50 Mb	893 Sec	32 Sec	5 Sec	0.5 Sec	0.05 Sec
ArcSDE Server to Workstation Client •1 MB => 10 Mb >> 5 Mb	89 Sec	3.2 Sec	0.5 Sec	0.05 Sec	0.005 Sec
Windows Terminal Server to Terminal Client (ICA) •100 KB => 1 Mb >> 280 Kb	5 Sec	0.18 Sec	0.03 Sec	0.003 Sec	0.0003 Sec
Web Server to Browser Client (HTTP) •50 KB => 500 Kb	9 Sec	0.3 Sec	0.05 Sec	0.005 Sec	0.0005 Sec
Web Server to ArcGIS Desktop Client (HTTP) •200 KB => 2 Mb	36 Sec	1.2 Sec	0.20 Sec	0.02 Sec	0.002 Sec

Best Practices

Web Services Network Performance

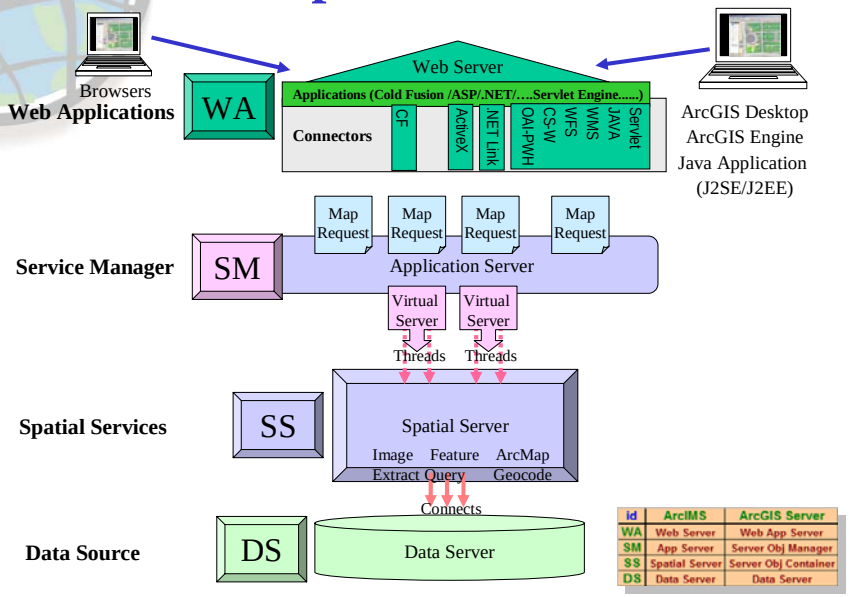
Wide Area Network Bandwidth	Peak Map Requests/Hour (based on Average Service traffic) 10 KB	30 KB	50 KB	75 KB	100 KB	400 KB
56 Kbps Modem	2,016	672	403	269	202	50
128 Kbps ISDN	4,608	1,536	922	614	461	115
256 Kbps DSL	9,216	3,072	1,843	1,229	922	230
512 Kbps	18,432	6,144	3,686	2,458	1,843	461
1.54 Mbps T-1	55,440	18,480	11,088	7,392	5,544	1,386
2 Mbps E-1	72,000	24,000	14,400	9,600	7,200	1,800
6.16 Mbps T-2	221,760	73,920	44,352	29,568	22,176	5,544
45 Mbps T-3	1,620,000	540,000	324,000	216,000	162,000	40,500
155 Mbps ATM	5,580,000	1,860,000	1,116,000	744,000	558,000	139,500

Note: 1 KB data = 10 Kb traffic

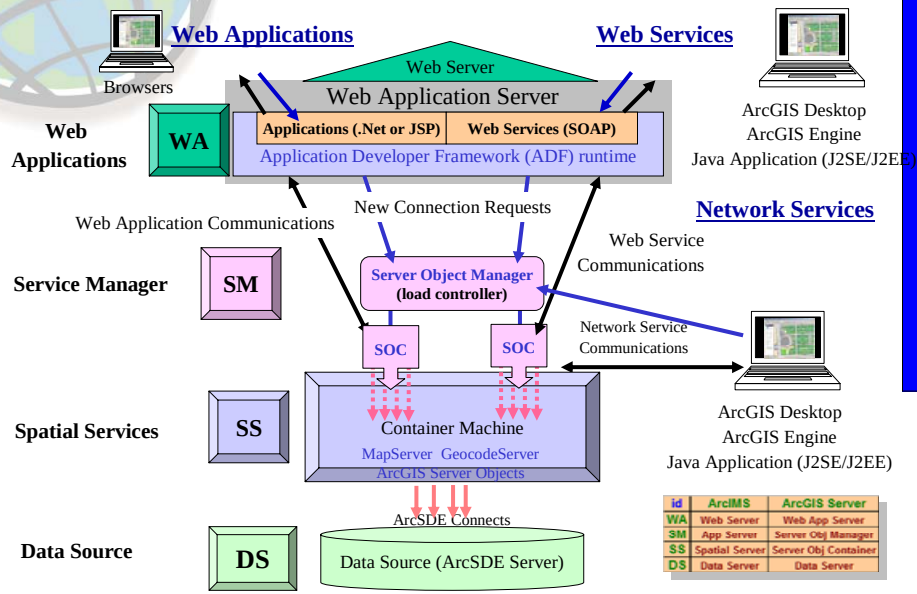
Wide Area Networks Bandwidth	File Transfer Time (sec) based on Average File Size 10 KB	30 KB	50 KB	75 KB	100 KB	400 KB
19 Kbps Modem	5	16	26	39	53	211
28 Kbps Modem	4	11	18	27	36	143
56 Kbps Modem	2	5	9	13	18	71
128 Kbps ISDN	1	2	4	6	8	31
256 Kbps DSL	0	1	2	3	4	16
512 Kbps	0	1	1	1	2	8
1.54 Mbps T-1	0	0	0	0	1	3
2 Mbps E-1	0	0	0	0	1	2
6.16 Mbps T-2	0	0	0	0	0	1
45 Mbps T-3	0	0	0	0	0	0
155 Mbps ATM	0	0	0	0	0	0

Use a Scalable Configuration Strategy

ArcIMS Component Architecture

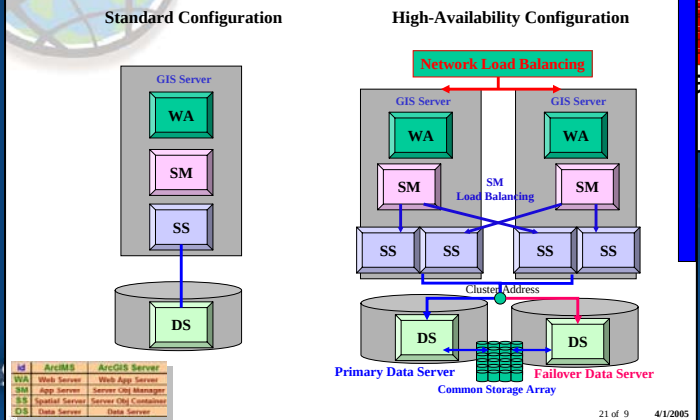


ArcGIS Server Component Architecture

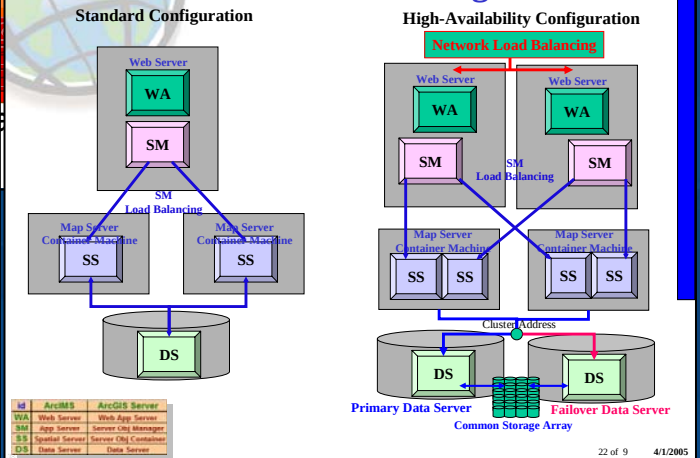


Two-tier Platform Configuration

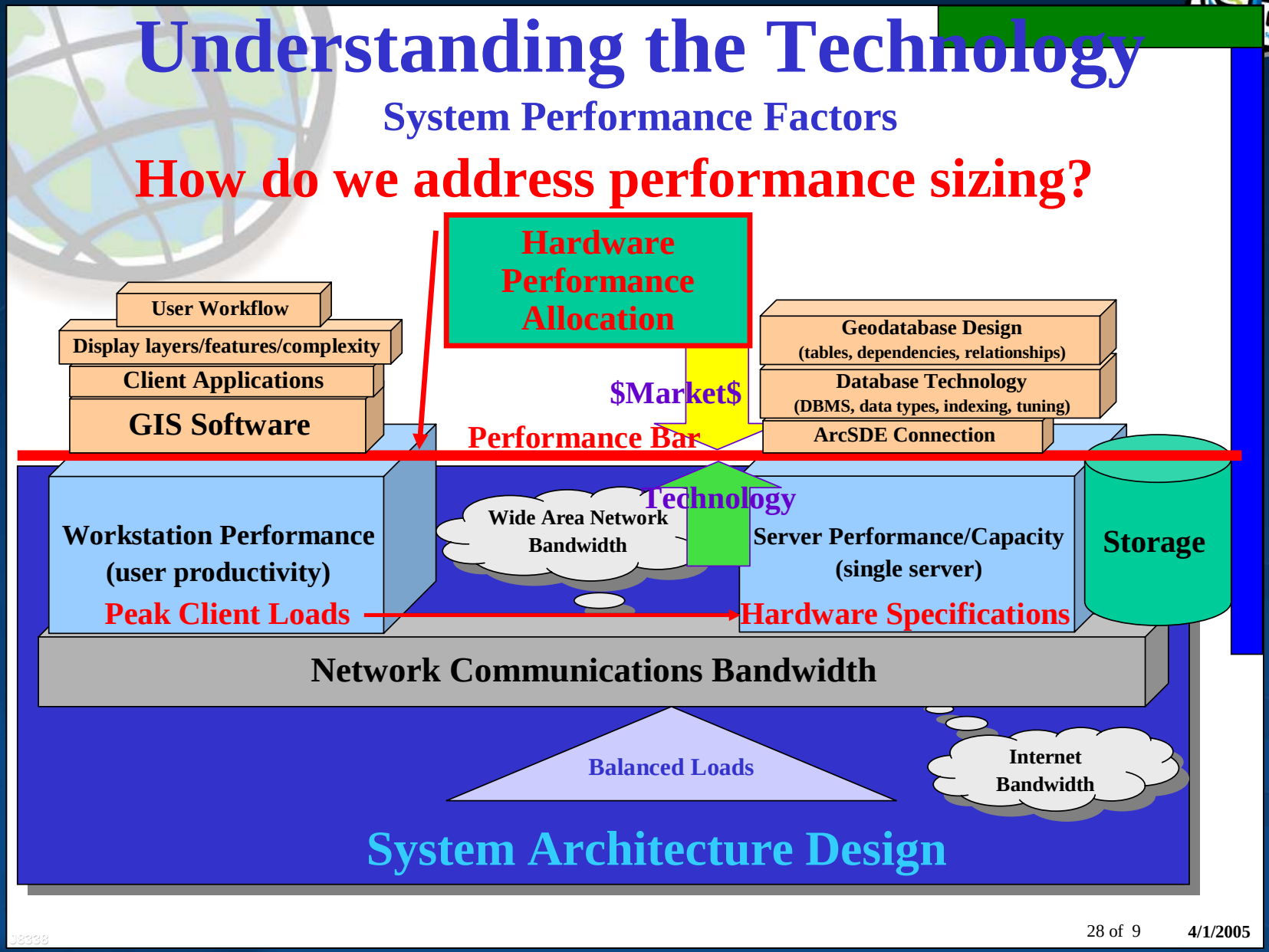
(separate data servers)



Three-tier Platform Configurations



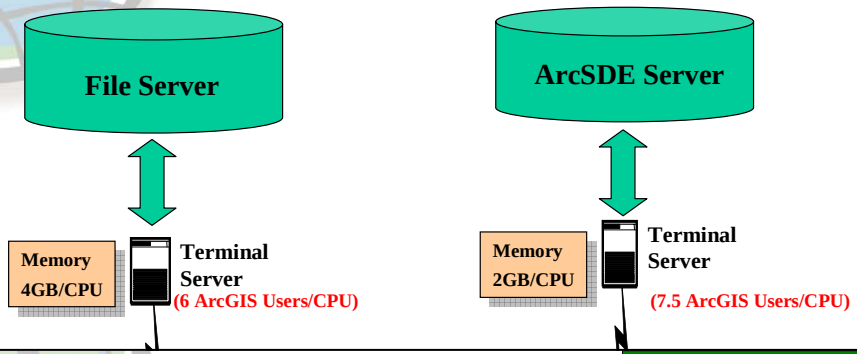
Understand Technology Needs



Web Performance Parameters

Terminal Server Performance

Terminal Server Performance Model

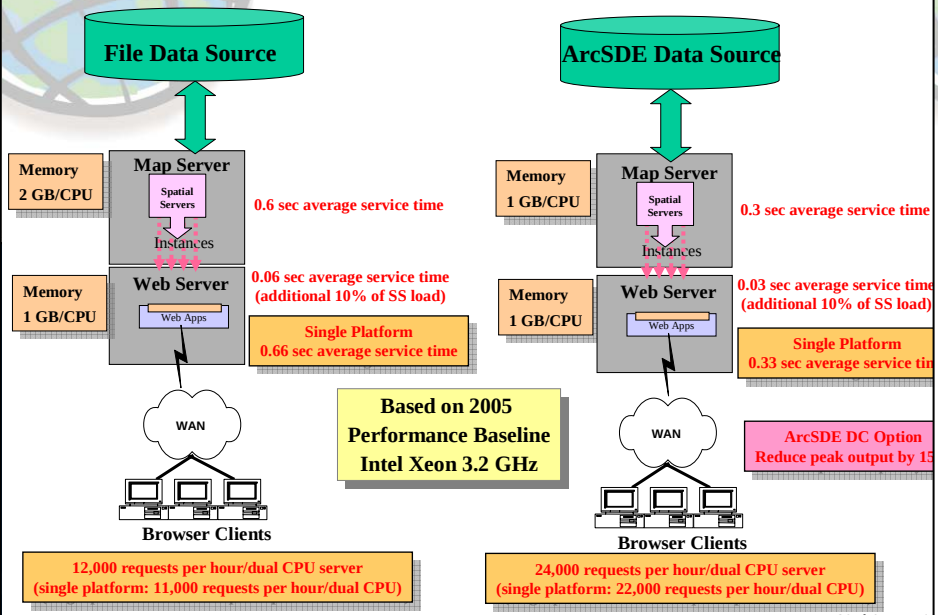


Web Services Performance

Performance Model Summary

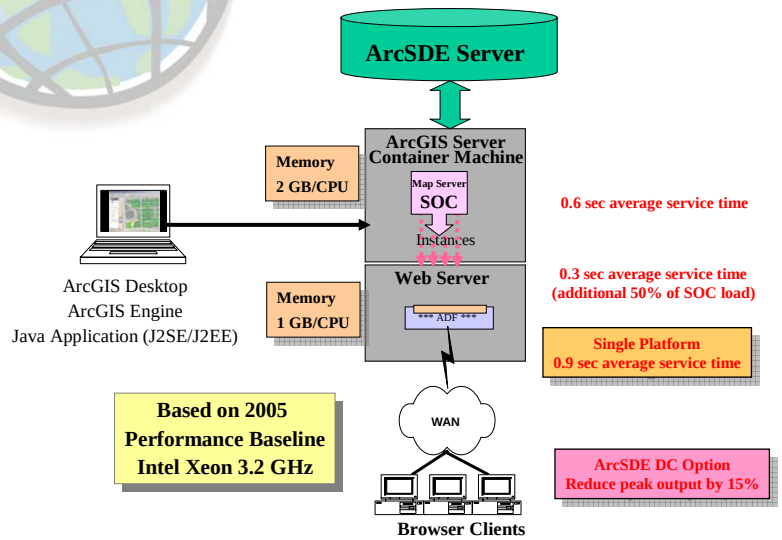
- **Calculate Maximum Configuration Capacity**
  - Measure Average Service Time (existing Web service site)
    - Measure map server CPU utilization during average service transaction
  - (#CPU × 3,600) ÷ Average Service Time on Map Server
    - Standard map service time for ArcIMS 2005 performance baseline
      - 0.6 seconds for file data source
      - 0.3 seconds for ArcSDE data source
    - Standard map service time for ArcGIS Server map server 2005 performance baseline
      - 1.2 seconds for file data source
      - 0.6 seconds for ArcSDE data source
- **Standard Peak Transaction Rate**
  - 2005 ArcIMS Baseline (per dual CPU map server)
    - 12,000 requests per hour for local data or remote file data source
    - 24,000 requests per hour for separate ArcSDE data source
  - 2005 ArcGIS Server Baseline (per dual CPU map server)
    - 6,000 requests per hour for local data or remote file data source
    - 12,000 requests per hour for separate ArcSDE data source

ArcIMS Server Performance



Web Application Transactions

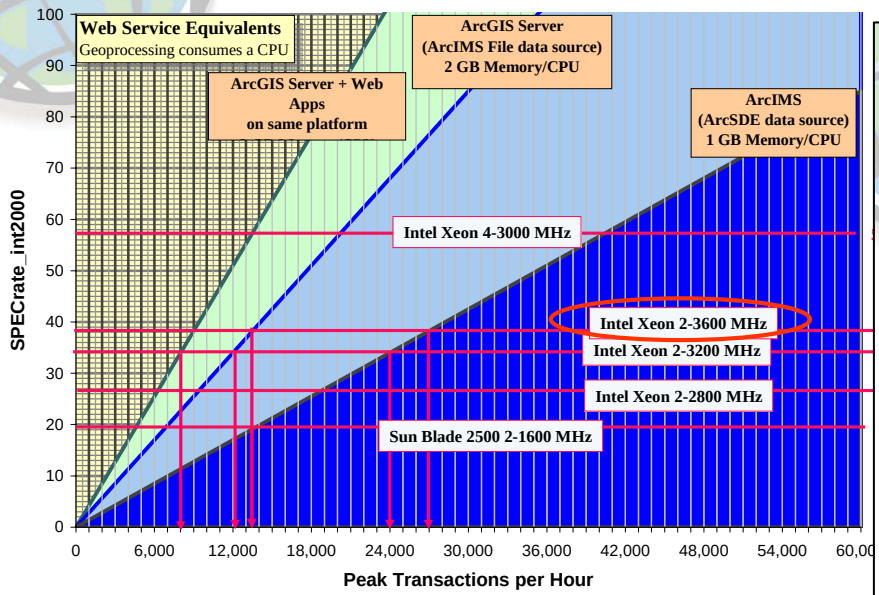
(Simple 9.0 Map Server Application)



Capacity Sizing Tools

ArcIMS/ArcGIS Server Sizing

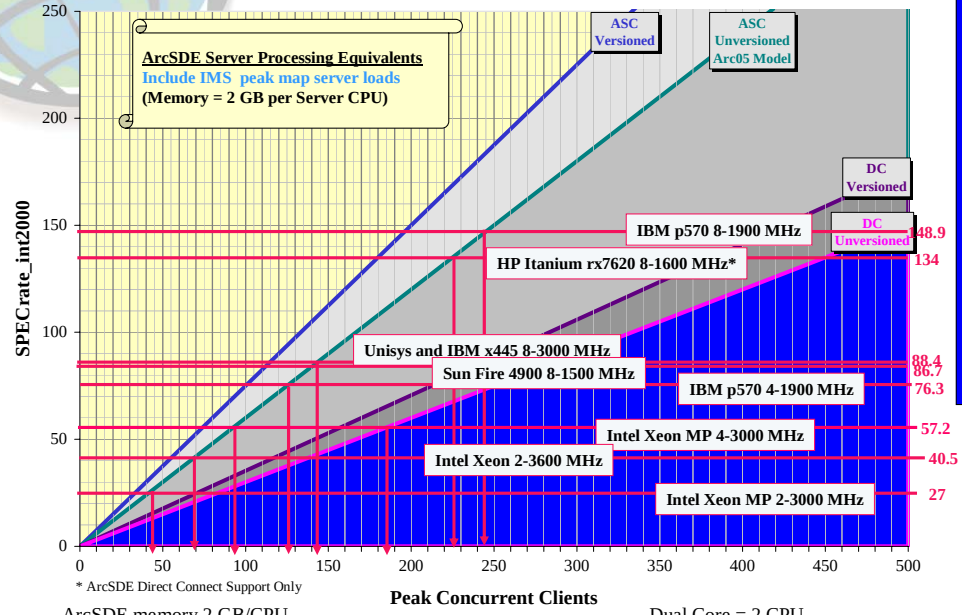
ArcIMS and ArcGIS Server



40 of 9

Workgroup GIS Data Server

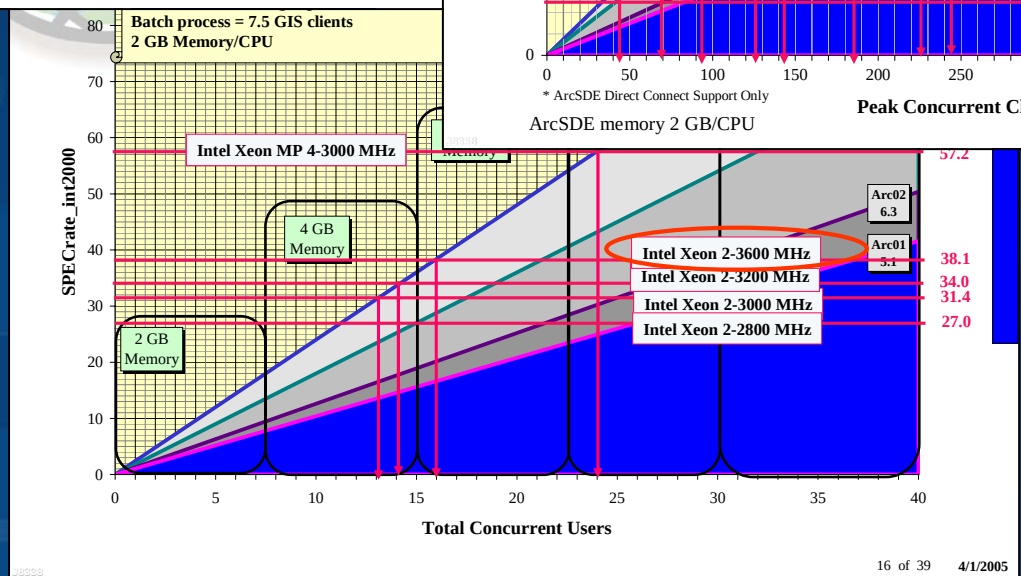
(Arc04-Arc05 ArcSDE Geodatabase Server)



ArcSDE memory 2 GB/CPU

Dual Core = 2 CPU

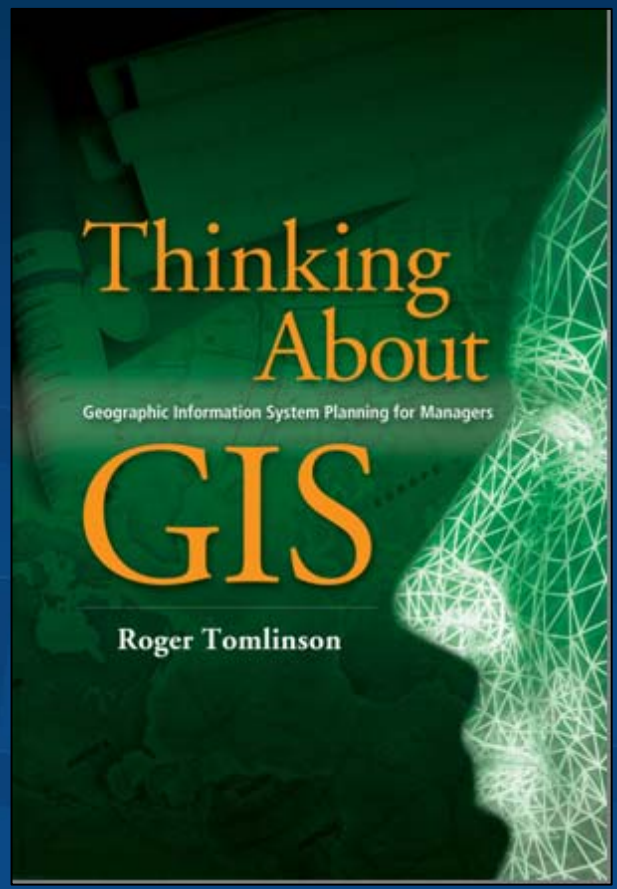
41 of 9 4/1/2005



16 of 39 4/1/2005

Planning Resources

www.esri.com/systemdesign



Dynamic Performance Sizing Models

Workload Summary	Total Users	Peak Users	DPM/client	Peak (DPH)	Web Traffic Analysis	
Equivalent Client Loads	Unlimited	610		66,600	Mbpq	Peak Mbps
ArcIMS Clients	5,000	500	3	90,000	1.000	25.000
ArcGIS Server	300	100	3	18,000	1.000	5.000
ArcGIS Desktop	20	10	6	3,600	5.000	5.000

ArcIMS	Production				Test	
	Sun Fire V440 4 CPU 1600 MHz				Intel Xeon 2 CPU 3200 MHz	
	SPECrate_int2000 = 38.7				SRint2000 = 37.1	
	Service Times Seconds	Capacity Display/Hour	Multiple Platforms Nodes	Capacity	Service Times Seconds	Capacity Display/Hour
Web Server	0.058	250,350	1	250,350	0.030	240,000
Web Applications	0.038		Peak DPH	90,000	0.020	
Application Server	0.019		Capacity	36%	0.010	
Map Server	1.246	11,555	6	69,328	0.650	11,077
GSRVR	0.096		Peak DPH	90,000	0.050	
Map Servers	1.150		Capacity	130%	0.600	

ArcGIS Server	Production				Test	
	Intel Xeon 4 CPU (2 dual core) 2800 MHz				Intel Xeon 2 CPU 3200 MHz	
	SPECrate_int2000 = 59.2				SRint2000 = 37.1	
	Service Times Seconds	Capacity Display/Hour	Multiple Platforms Nodes	Capacity	Service Times Seconds	Capacity Display/Hour
Web Server	0.389	37,061	1	37,061	0.310	23,226
Web Applications	0.376		Peak DPH	18,000	0.300	
Server Object Manager	0.013		Capacity	49%	0.010	
Container Machine	0.877	16,413	2	32,826	0.700	10,286
GSRVR	0.125		Peak DPH	18,000	0.100	
Server Object Containers	0.752		Capacity	55%	0.600	

DBMS Server	Production				Test	
	Sun Fire V490 16 CPU (8 dual core) 1350 MHz				Intel Xeon 2 CPU 3200 MHz	
	SPECrate_int2000 = 131.0				SRint2000 = 37.1	
	Service Times Seconds	Capacity Display/Hour	Multiple Platforms Nodes	Capacity	Service Times Seconds	Capacity Display/Hour
Geodatabase Server	0.521	110,536	1	110,536	0.330	21,818
			Peak DPH	66,600		
GSRVR			Capacity	60%	0.100	
DBMS	0.521		Peak Users	307	0.230	

Performance Validation Testing

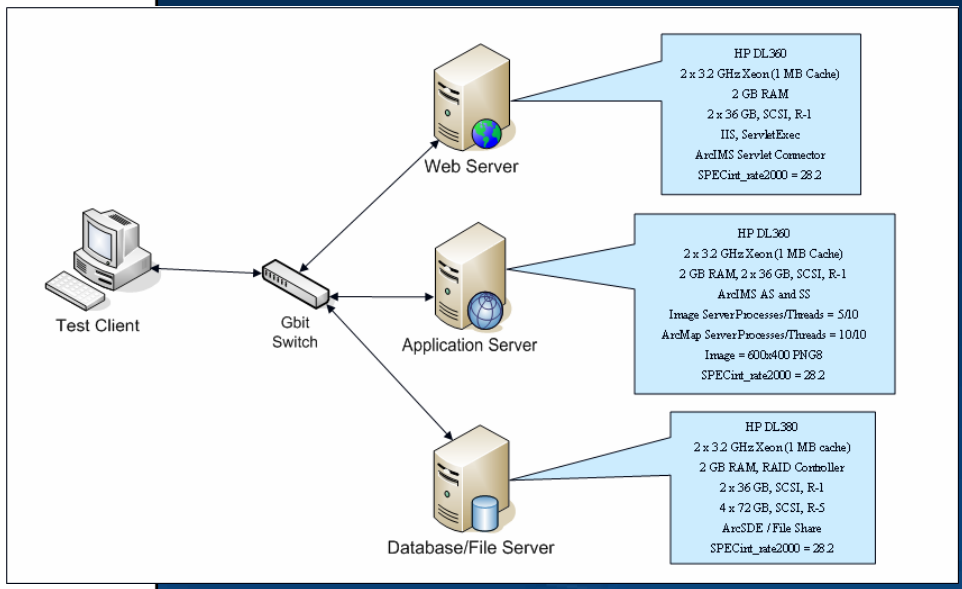
Jeff DeWeese

Performance Validation Testing

Web Application Stress Test Methodology

ESRI Systems Integration Technical Brief

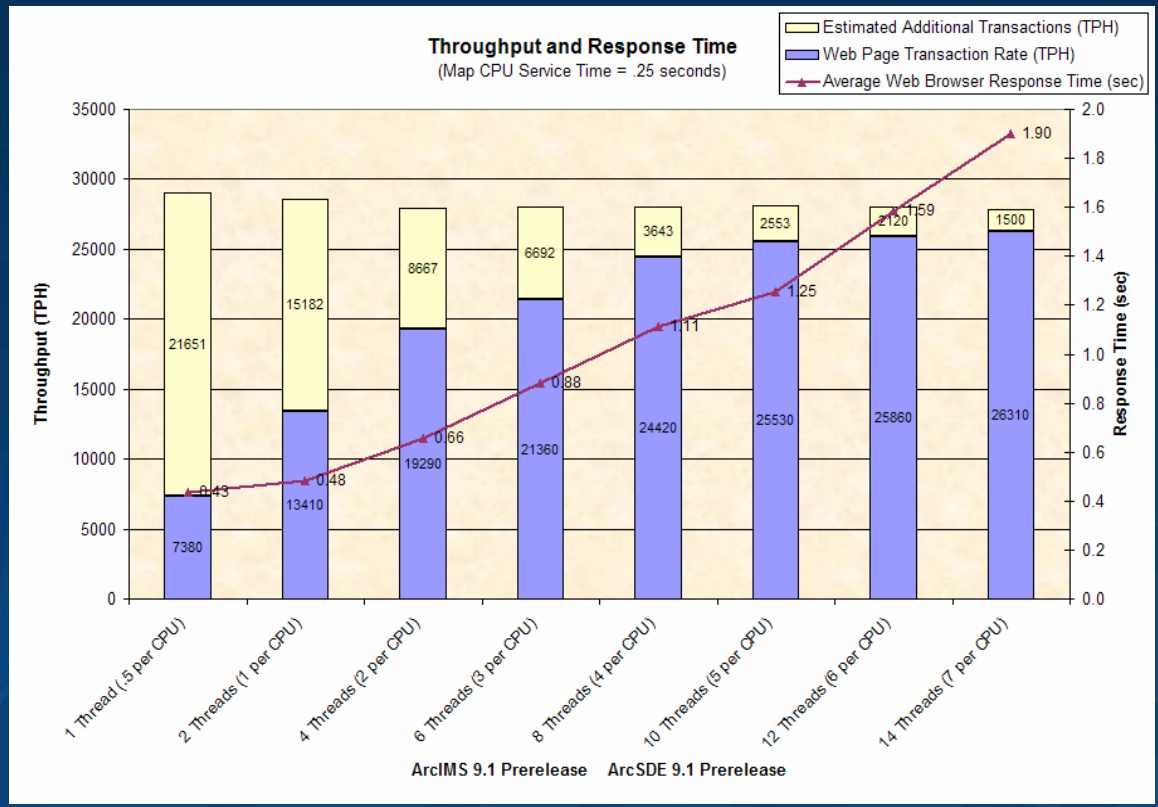
June 2005



http://www.esri.com/systemsint/kbase/docs/stress-test-methodology.pdf

Web Stress Testing (What are we measuring?)

- Transaction
- Throughput
- Utilization
- Service Time
- Queue Time
- Response Time
- Scalability
- Performance



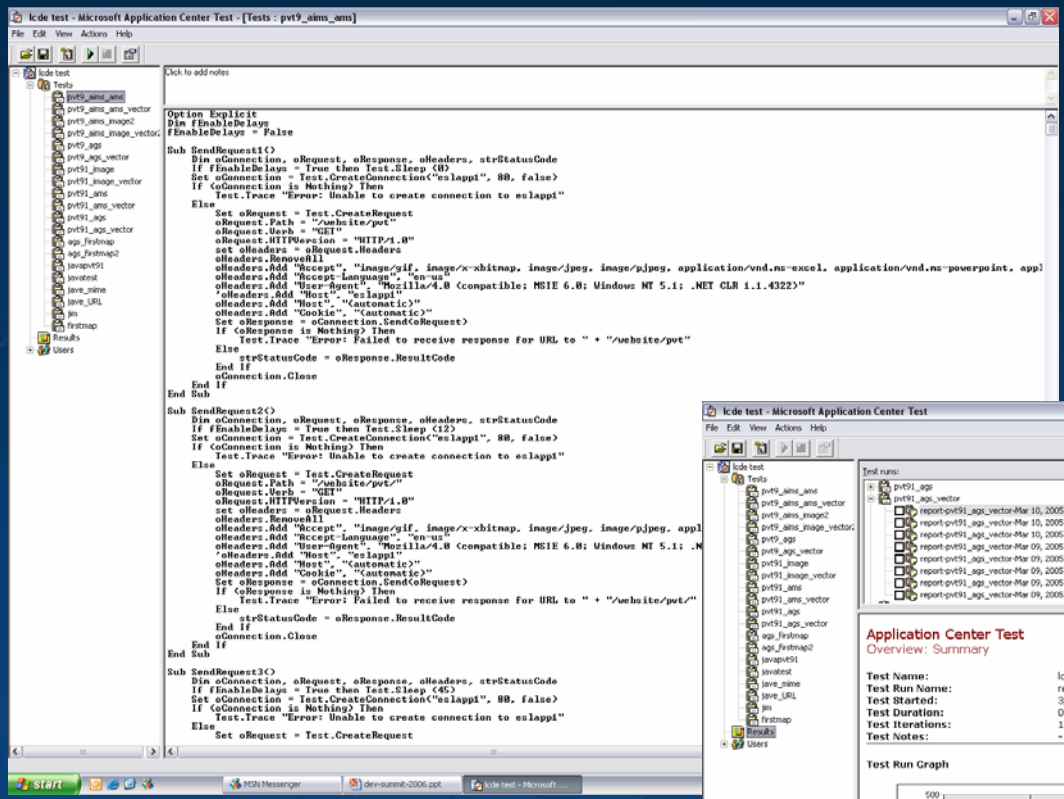
Test Readiness
(Doing our homework?)

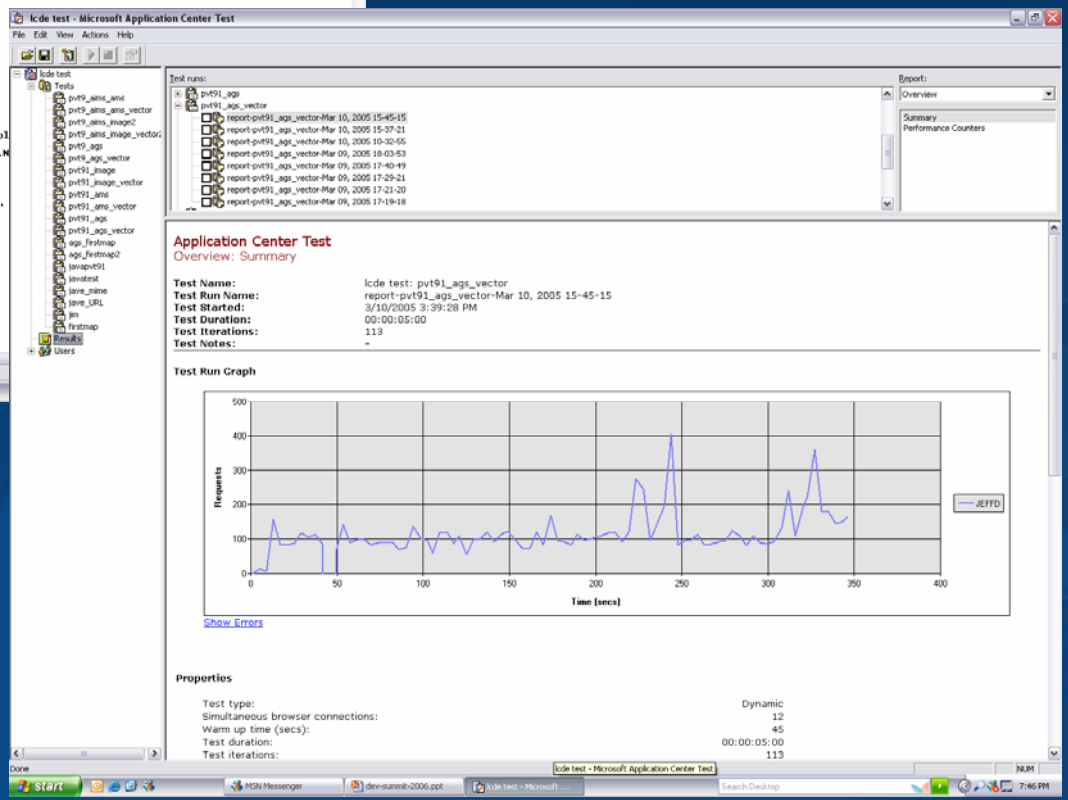
- Application Center Test
- Develop a Hypothesis
- Analyze the Application Design
- Web Configuration Tuning
- System Analysis
- Application Workflow
- System Monitoring

Web Stress Testing

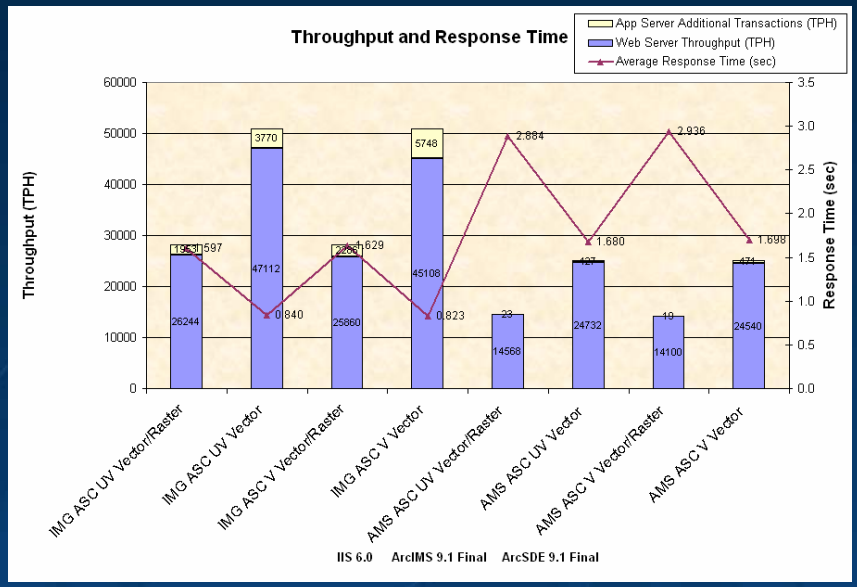
- Application Creation and Configuration
- Recording the Script
- Script Validation
- Calculating Transaction Throughput
- Calculating Map Service CPU Time
- Calculating Web Browser Response Time
- Script Execution and Configuration Tuning

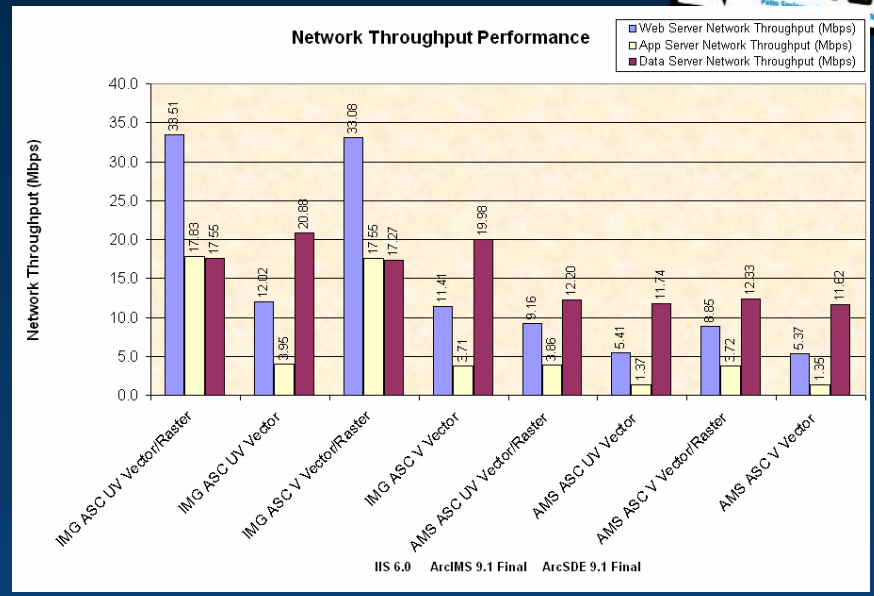
Application Center Test (VS .NET 2003)

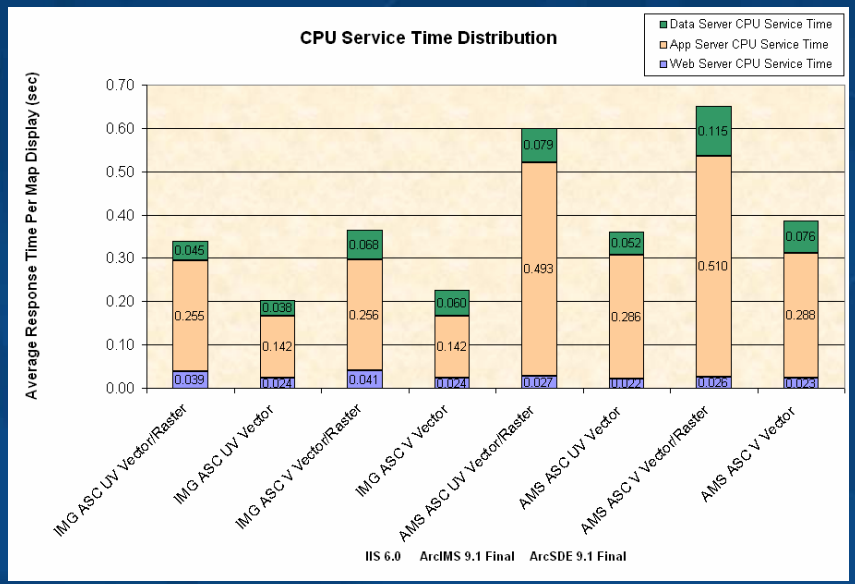


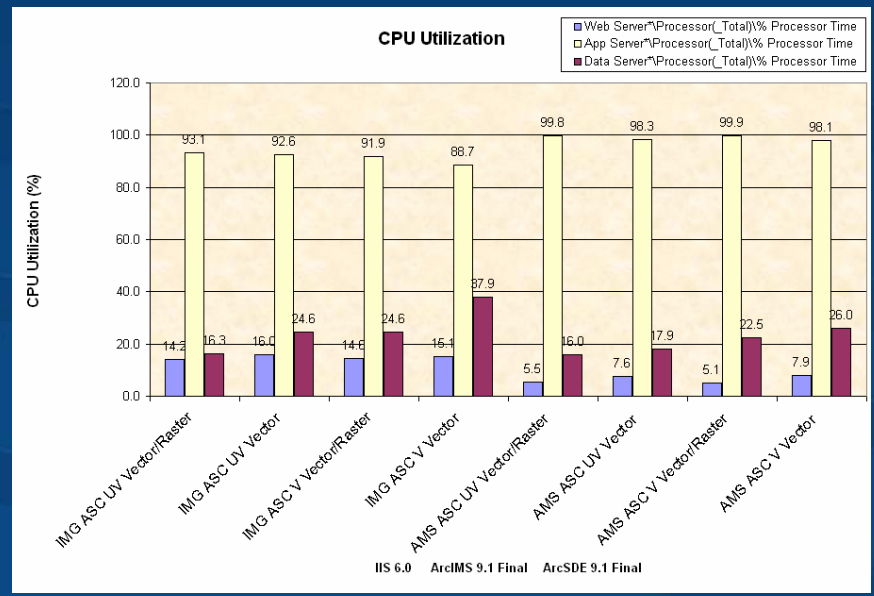


Test Results









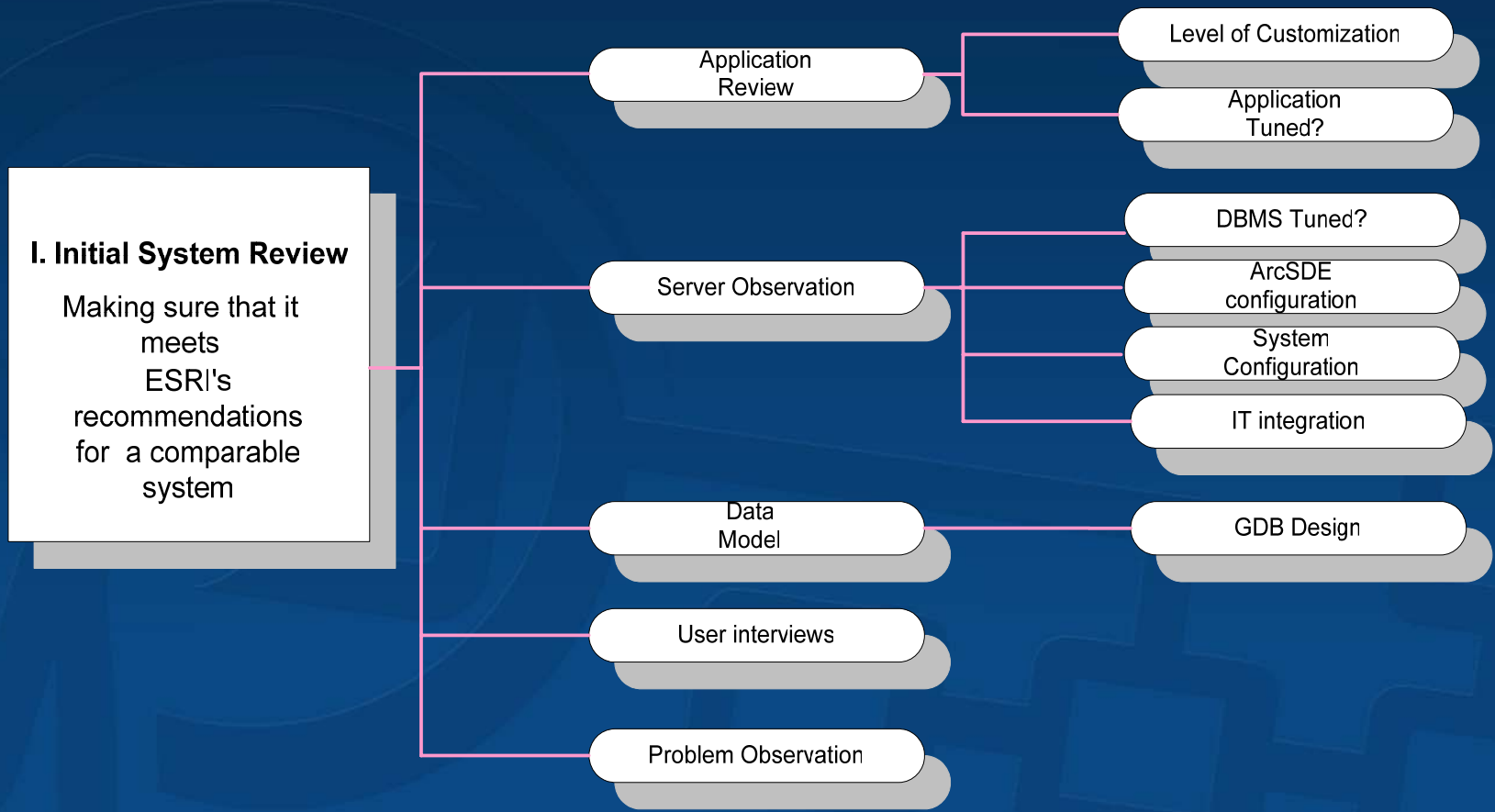
Performance Testing and Tuning

John Mesa

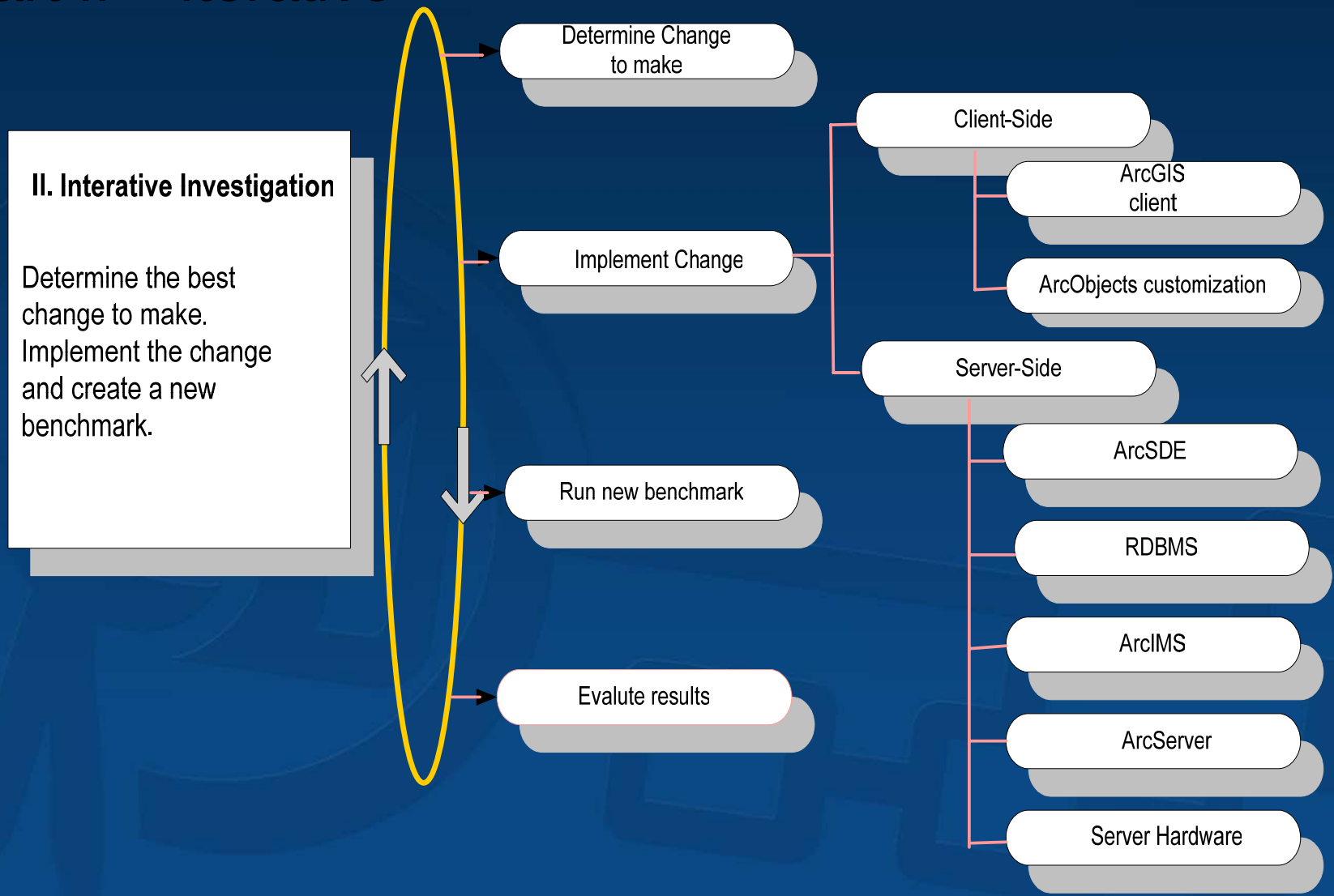
Performance and Scalability
5 Common Causes

- Lack of Geodatabase / RDBMS knowledge
  - RDBMS – inefficient SQL
  - State lineage – custom editing tools add unneeded states per edit to lineage
- Workflow
- Data Models
- Application
  - Too chatty
  - Unneeded/excessive interaction with RDBMS/GDB
- Inadequate testing
  - Real data
  - Scalability testing

Performance Investigation
Part I – initial system review



Performance Investigation
Part II – Iterative



ArcObjects Instrumentation

- ArcObjects customization
  - Must be scalable
  - Tested for functionality
  - Not often tested for scalability
  - Code must be instrumented by development
- Instrumentation Benefits
  - Provides execution trace for debugging
  - Aids performance analysis and investigation
  - Useful for monitoring application in production

Instrumentation Example

- Actual InstruOutput
- [C[illegible] M] [User: iggypop] [Origin: [illegible]ion::IObjectClassEvents_OnCreate] [Message: OnCreate Called]
- [Call Time: 1:04:50 PM] [User: iggypop] [Origin: GDBLandClassExtension::IObjectClassEvents_OnCreate] [Message: CreateFeatureCopy]
- [all Time: 1:04:50 PM] [User: iggypop] [Origin: modDBFunctions::GetID] [Message: Getting New ID Value]
- [Call Time: 1:04:50 PM] [User: iggypop] [Origin: modDBFunctions::GetID] [M[illegible]Call Time: 1:04:50 PM] [User: iggypop] [Origin: modFunctions::CreateFeatureCopy] [Message: Starting Process for Feature Type: STAND]
- [Cldevent_id = 4049294 And GF_TYPE ='STAND' and date_complete is NULL]
- [Call Time: 1:04:50 PM] [User: iggypop] [Origin: modDBFunctions::GetOracleSequence] [Message: Getting New Sequence Value]
- [Call Time: 1:04:50 PM] [User: iggypop] [Origin: modDBFunctions::GetOracleSequence] [Message: New Sequence Value: 4427868]
- [Call Time: 1:04:50 PM] [User: iggypop] [Origin: modFunctions::CreateFeatureCopy] [Message: SQL: select stand_id from genisys.gis_stand where gf_id = 4049294]
- [Call Time: 1:04:51 PM] [User: iggypop] [Origin: modFunctions::CreateFeatureCopy] [Message: SQL: duplicate_stand (4427868, 4049294,'1371326004')]

Timing Information

Embedded SQL

Edit Code


```
pWsEdit.StartEditing
pWsEdit.StartEditOperation
pFeat = pUpdateCur.NextFeature
Do While not pFeat is Nothing
  pFeat.Value(1) = NewVal
  pFeat.Store
  pFeat = pUpdateCur.NextFeature
Loop
pWsEdit.StopEditOperation
pWsEdit.StopEditing (True)
```

All updates efficiently performed in one edit operation. Entire operation can be rolled back.

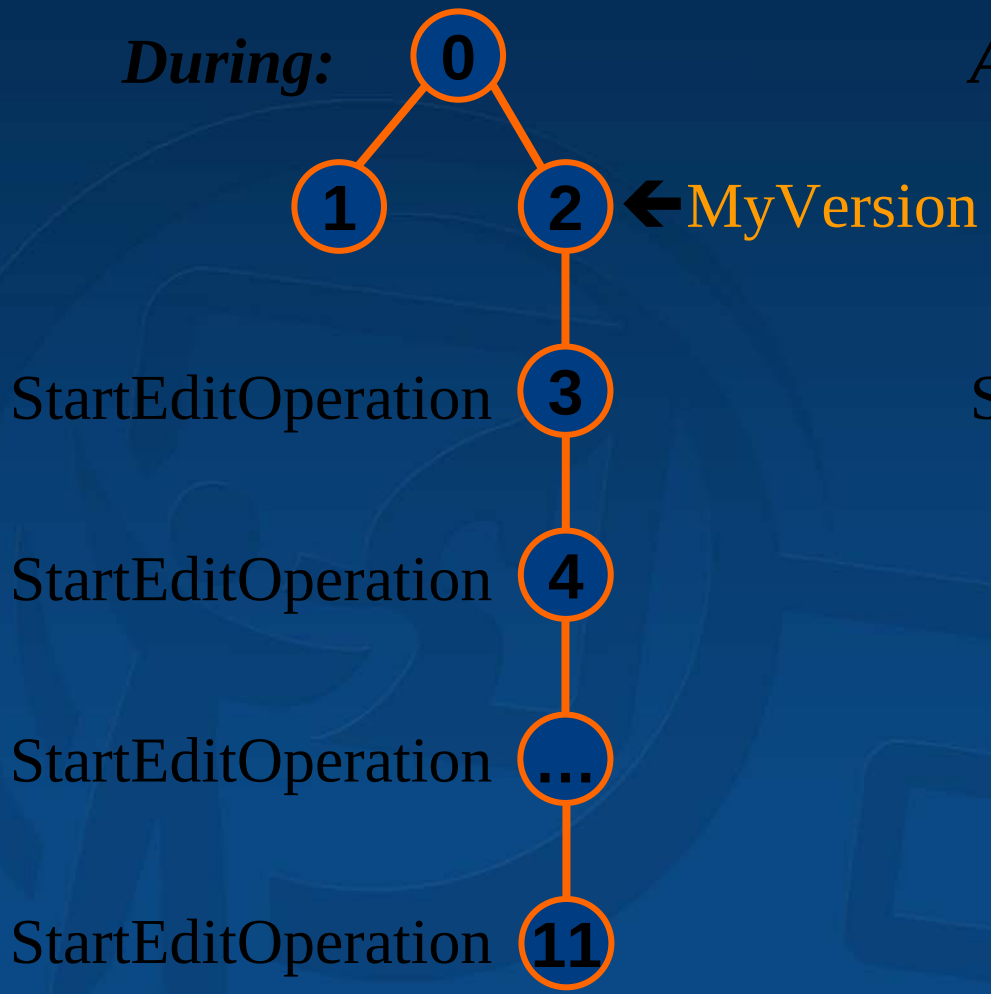
Resulting State Tree

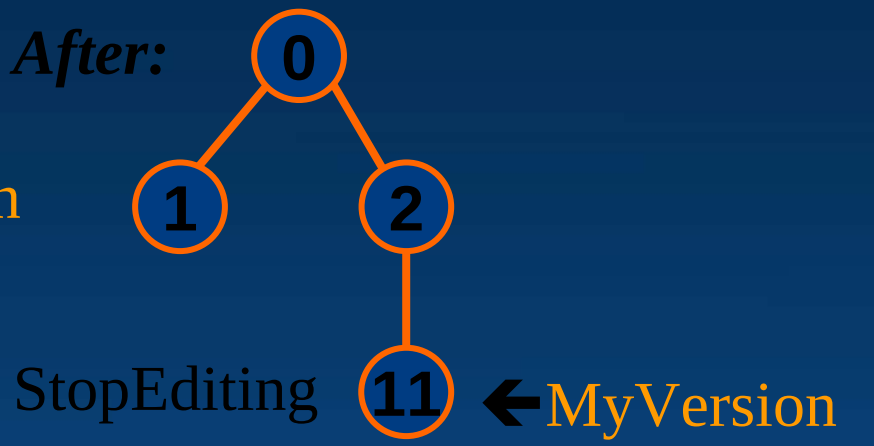
Edit Code

```
pWsEdit.StartEditing
pFeat = pUpdateCur.NextFeature
Do While not pFeat is Nothing
  pWsEdit.StartEditOperation
  pFeat.Value(1) = NewVal
  pFeat.Store
  pFeat = pUpdateCur.NextFeature
  pWsEdit.StopEditOperation
Loop
pWsEdit.StopEditing (True)
```

Not recommended for update loops, due to overhead for creating internal states.

Resulting State Tree





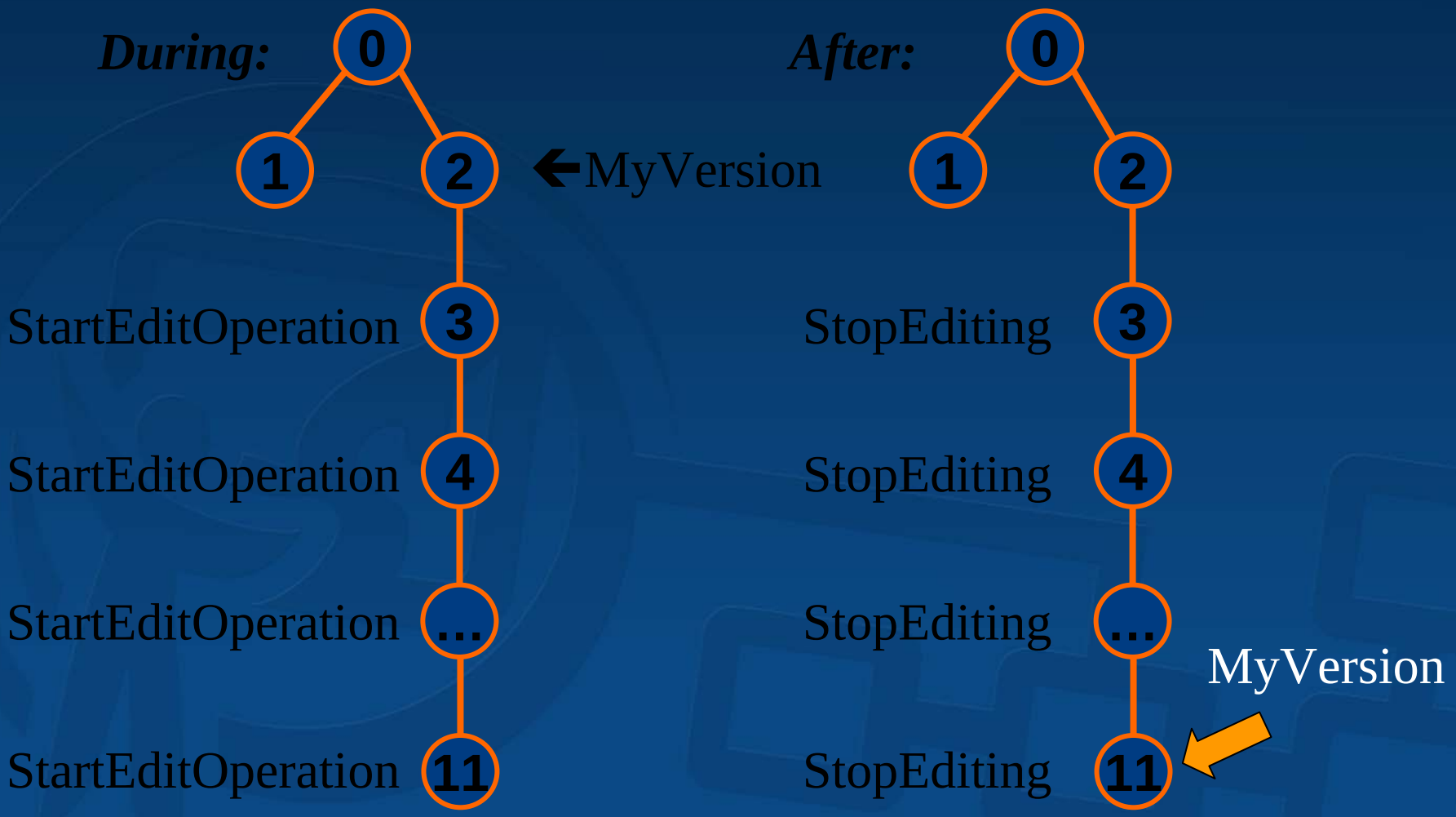
Edit Code

```
pFeat = pUpdateCur.NextFeature
Do While not pFeat is Nothing
   pWsEdit.StartEditing
   pWsEdit.StartEditOperation
   pFeat.Value(1) = NewVal
   pFeat.Store
   pFeat = pUpdateCur.NextFeature
   pWsEdit.StopEditOperation
   pWsEdit.StopEditing(True)
Loop
```

This is never appropriate:

- High overhead
- Persistent state for every edit
- Cannot be aborted
- May require reconcile on each row

Resulting State Tree



Performance and Scalability Testing

- Testing ArcGIS Enterprise Systems
  - Why
    - Develop and deliver systems that satisfy client needs and meet their expectations
    - Reduce the overall cost of delivering the software
  - When
    - Development, UAT, Investigation
  - What
    - ArcGIS spatial and attribute data, display, editing …
  - How
    - Manually, COTS Tools, In-House Developed tools
  - What to look for
    - Development of scripts, multiple levels of testing

Performance and Scalability Testing

- Testing
  - Identify and eliminate performance problems *before* they get into production
  - ***Performance testing and analysis must occur throughput development!!!***
  - In late cycle QA, should be a formality with no surprises.
  - A surprise at this point will delay product release or potentially kill a product.

Session Evaluations Reminder

Session Attendees:

Please turn in your session evaluations.

. . . Thank you