

The background is a solid blue color. In the center, there is a faint, hand-drawn style illustration of a globe. The globe is surrounded by the text "helping manage our world" in a cursive, handwritten font. Below the globe, the word "esri" is written in a similar font. At the bottom right, there is a small, stylized logo that looks like a globe with the letters "GIS" inside it.

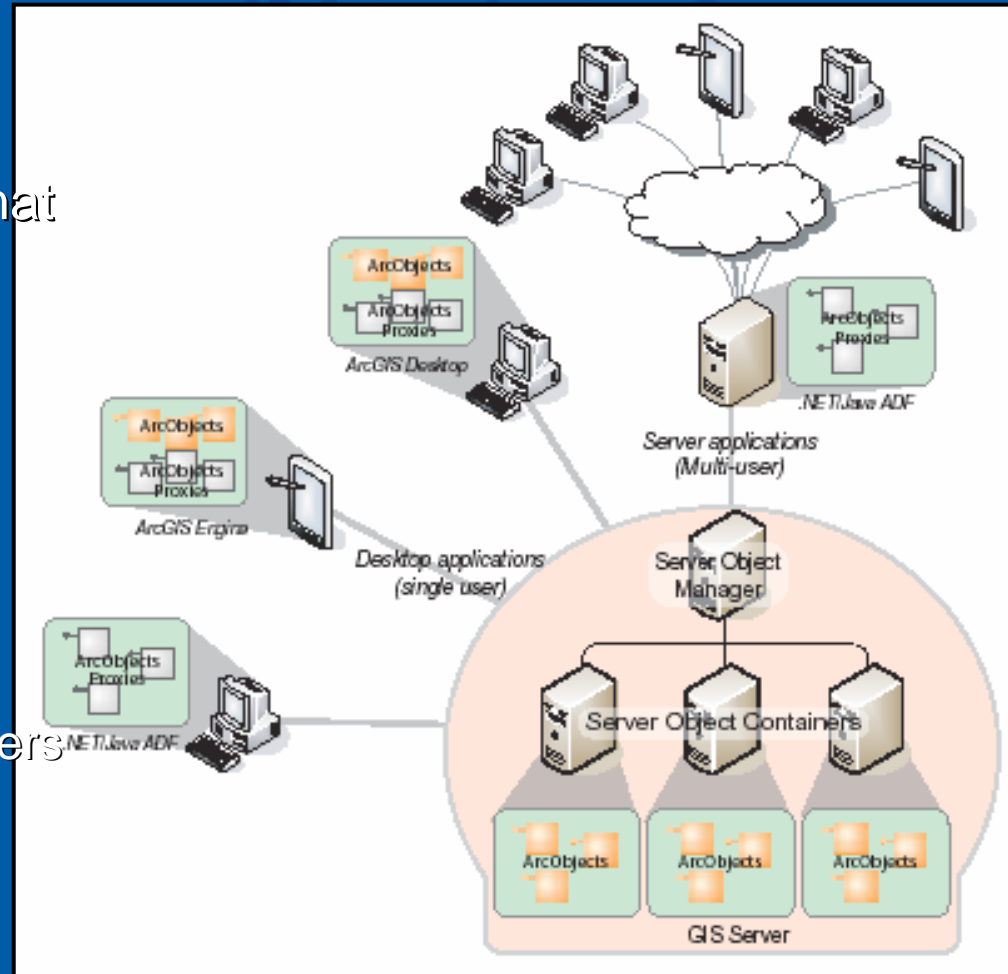
# ArcGIS Server Performance and Scalability Testing

# Network Analyst Samples on EDN

- Point to Point Driving Directions
- Closest Facility
- Routing with Time Windows
- Samples in C#, VB.Net, and Java
  - Includes a NetworkAnalyst utility class

# Built using ArcGIS Server

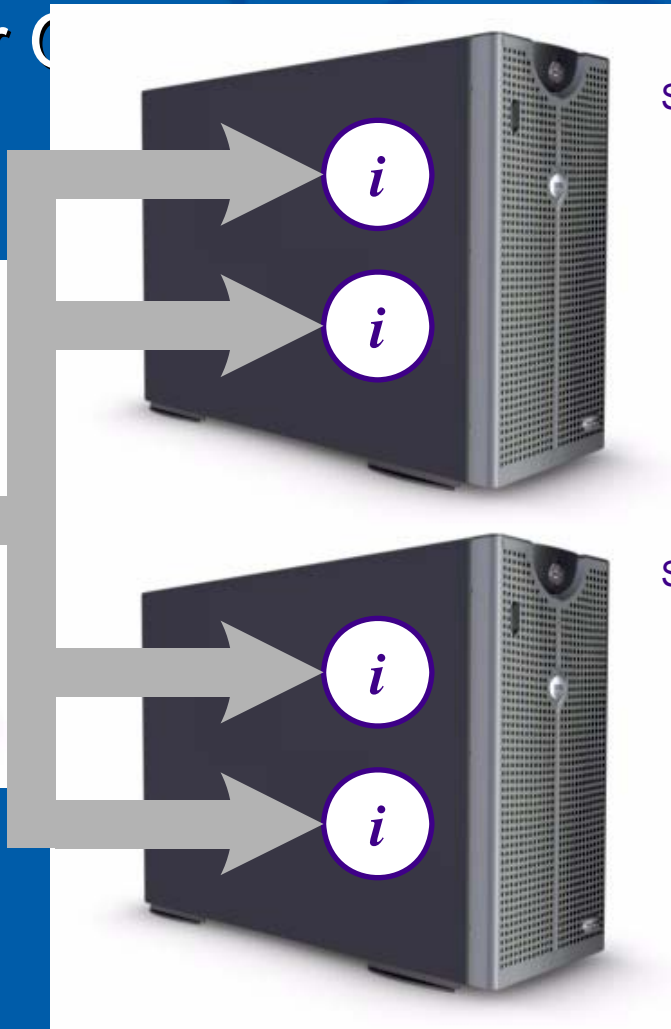
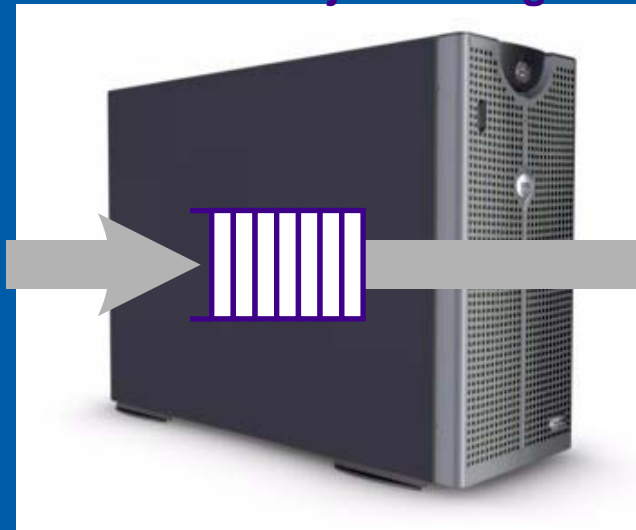
- Software components that can be used to build server-side GIS applications
- Designed for
  - Building centrally managed applications
  - Expose advanced GIS functionality
  - Supporting multiple users



# ArcGIS Server

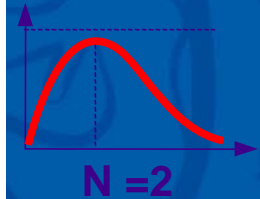
- Example: Server Object Manager

ArcGIS Server  
Server Object Manager

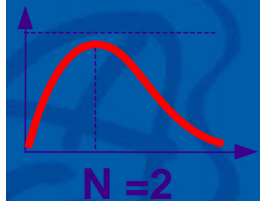


Instances

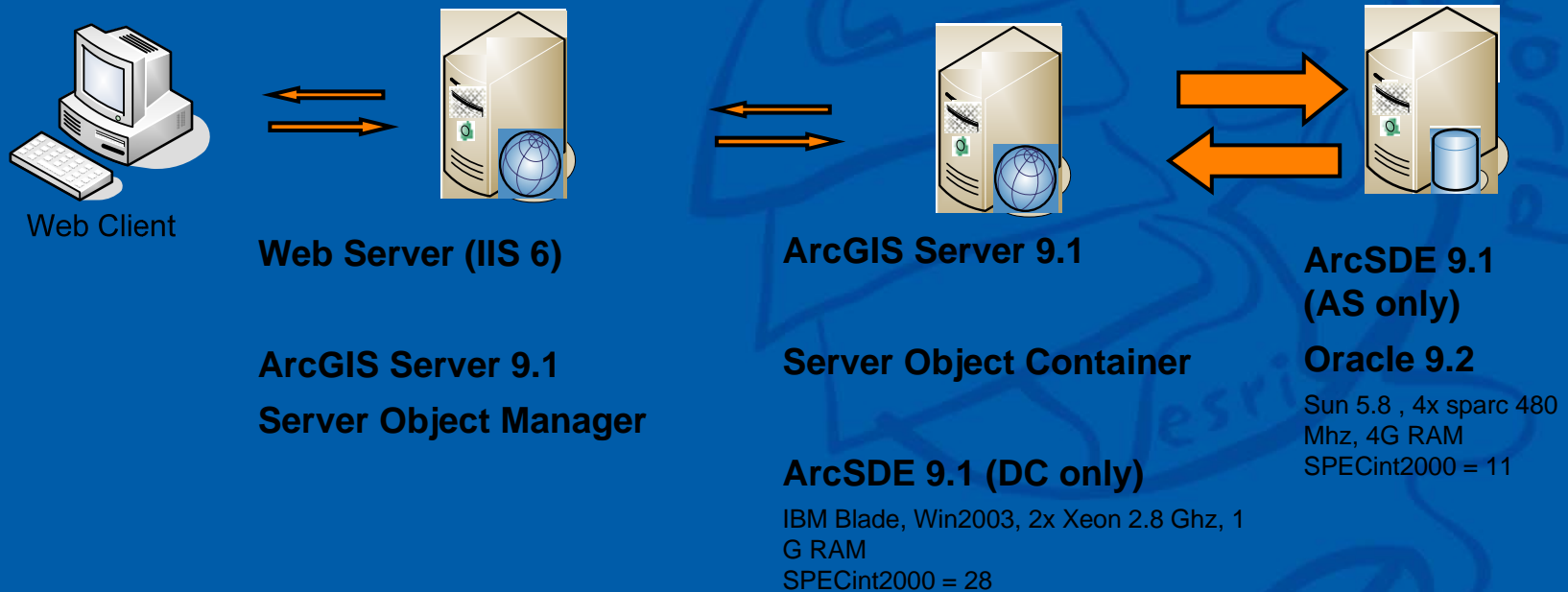
Server Object Containers  
2 instances  
1 CPU



Server Object Containers  
2 instances  
1 CPU

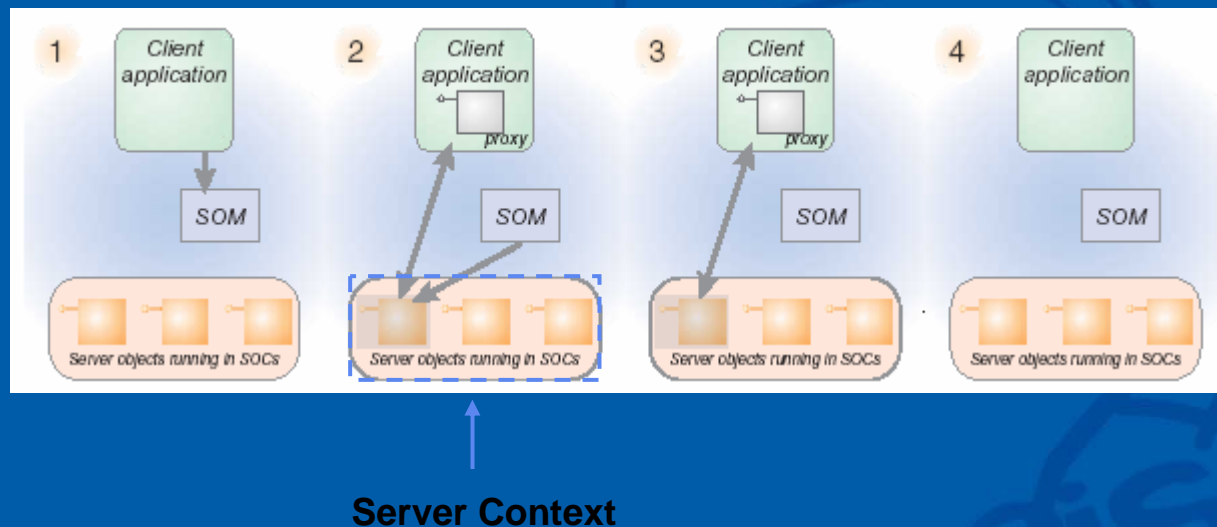


# ArcGIS Server Case Study Hardware Environment



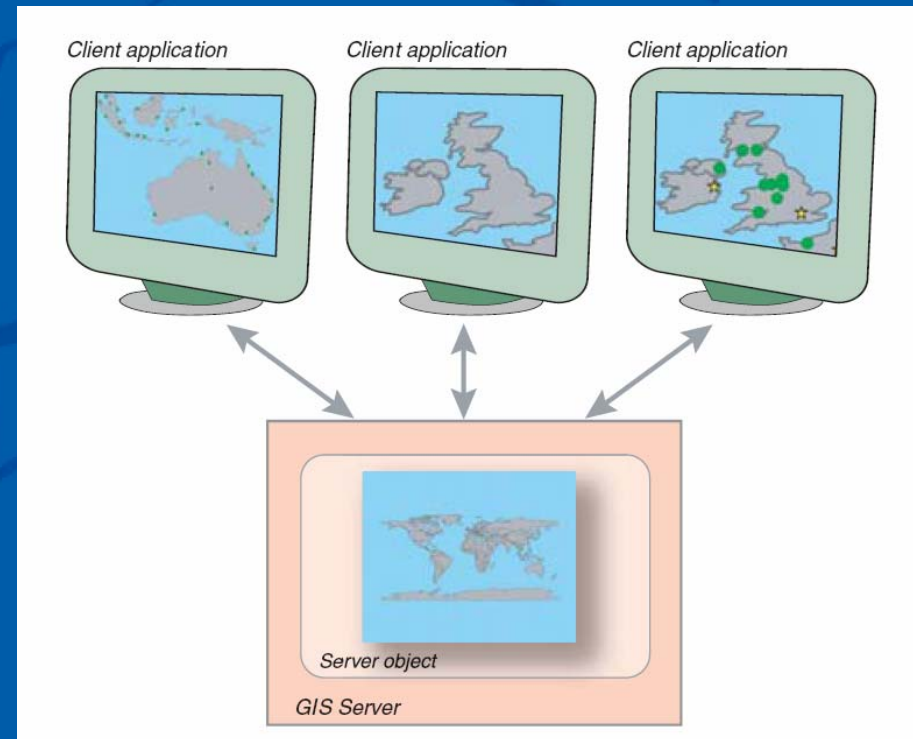
# Stateless use of Pooled Server Objects

1. Connect to SOM and request server object
2. SOM accesses *context* and returns a proxy
3. Client accesses server object through proxy
4. Client releases server object *back to the pool*



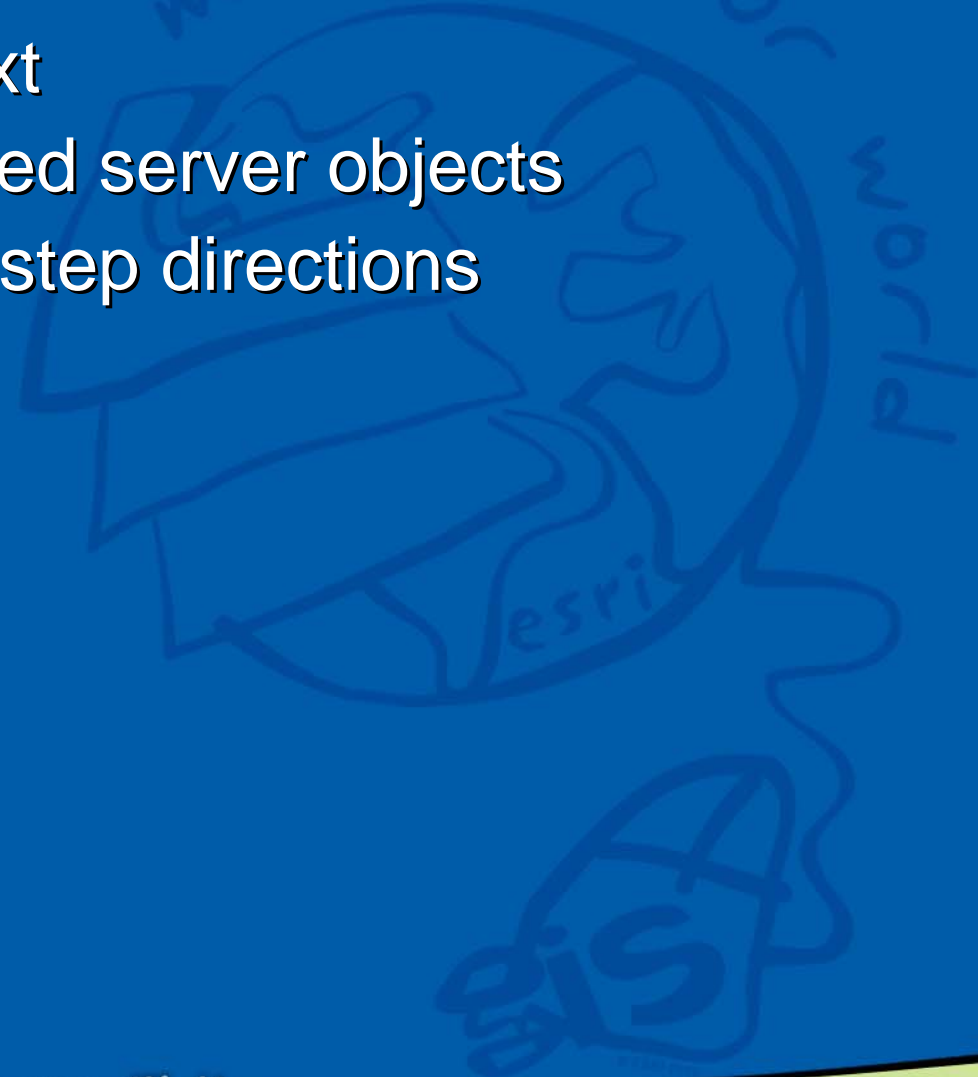
# Shallowly Stateful Applications

- Most GIS web applications must maintain some state information (Extent of the map, layer visibility, etc.)
- Use the web server's session state management
- Cannot change the properties of a pooled server object
- Scale better due to shared object pool



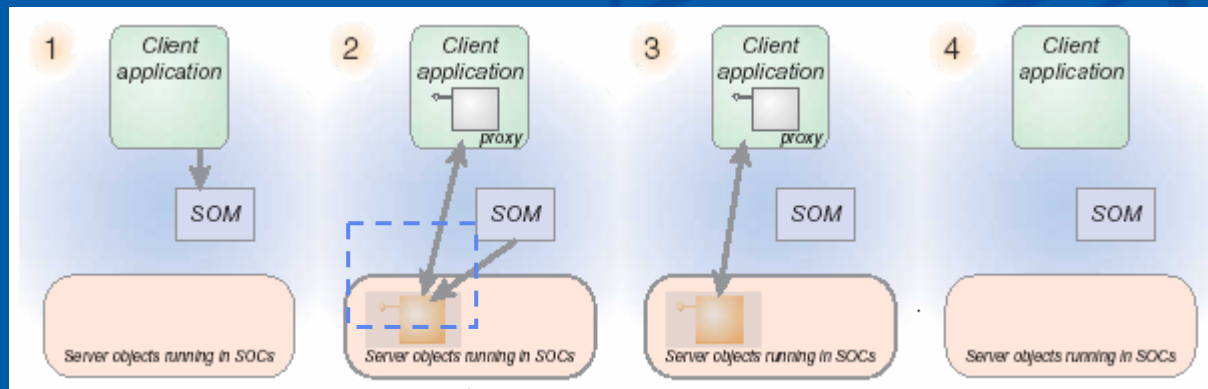
# Network Analyst and Shallowly Stateful Applications

- Serializable context
- Allows use of pooled server objects
- Example: Step by step directions



# Stateful Use of Non-pooled Server Objects

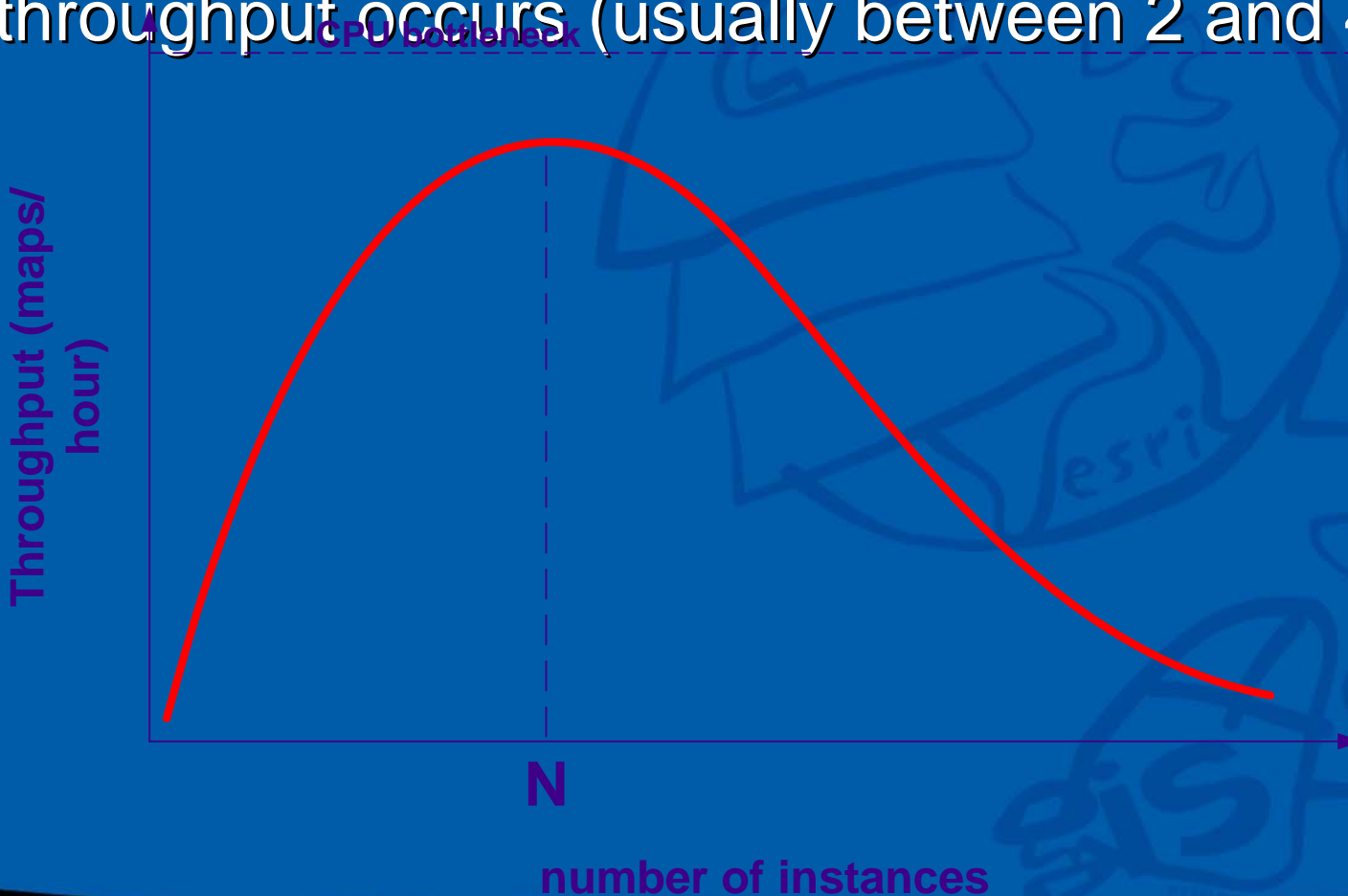
1. Connect to SOM and request server object
  2. SOM creates object
  3. SOM accesses the **context** and returns a proxy
  4. Client accesses server object through proxy
  5. Client releases server object, *object is destroyed*
- Example: editing applications with undo / redo



Server Context

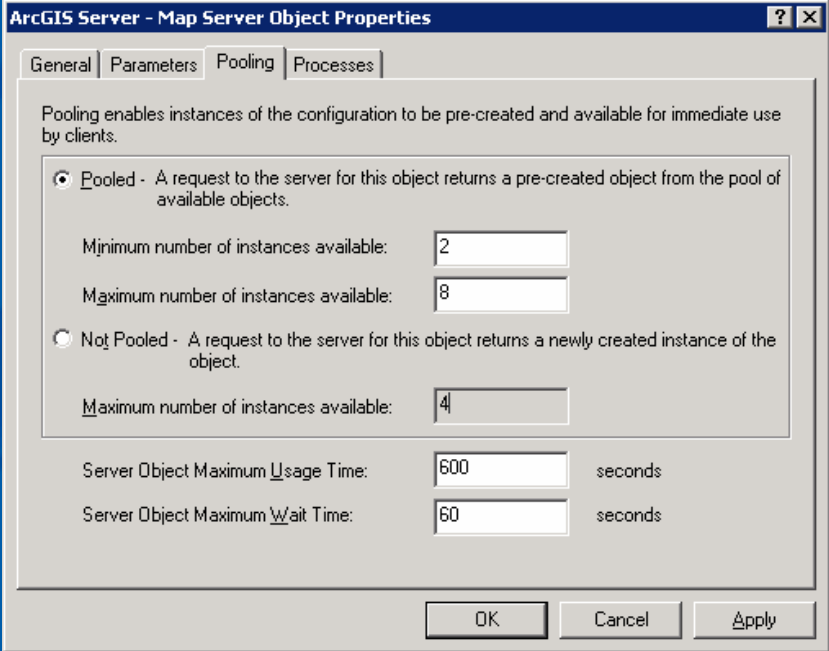
# ArcGIS Server Optimum Number of Instances

- Set instances to level where maximum throughput occurs (usually between 2 and 4)



# ArcGIS Server – Configuring Pooled Instances

- Define Min-Max instances
- Instances are distributed across all host servers



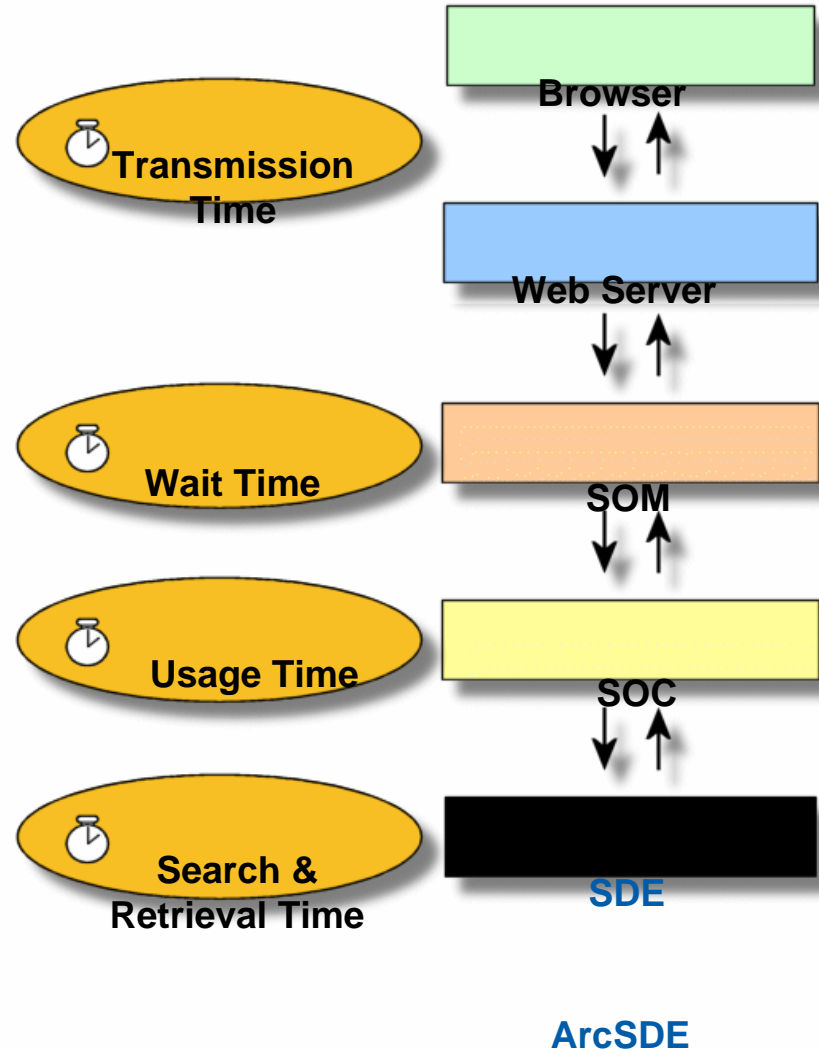
The screenshot shows the 'ArcGIS Server - Map Server Object Properties' dialog box with the 'Pooling' tab selected. The dialog has four tabs: 'General', 'Parameters', 'Pooling', and 'Processes'. The 'Pooling' tab contains the following settings:

- A descriptive text: "Pooling enables instances of the configuration to be pre-created and available for immediate use by clients."
- Two radio buttons for pooling options:
  - Pooled** - A request to the server for this object returns a pre-created object from the pool of available objects.
  - Not Pooled** - A request to the server for this object returns a newly created instance of the object.
- Input fields for the 'Pooled' option:
  - Minimum number of instances available: 2
  - Maximum number of instances available: 8
- Input field for the 'Not Pooled' option:
  - Maximum number of instances available: 4
- Server Object Maximum Usage Time: 600 seconds
- Server Object Maximum Wait Time: 60 seconds

At the bottom of the dialog are three buttons: 'OK', 'Cancel', and 'Apply'.

# ArcGIS Server: Response Time Factors

- Four main factors of response time
- Multiple tiers
- Performance bottleneck can occur in each tier



# Monitor Statistics and Log Files

- Creation Time
- Wait Time
- Usage Time

**ArcGIS Server Properties**

General | Hosts | Directories | **Statistics** | Types

Server Obj: SanDiego.MapServ

Host(s): <All>

Stat. Type: SO. Usage Time

Interval: Last 60 minutes

Show Statistics

Statistics Time Range

Start Time: 2005-07-19T16:55:00

Server Object Usage Time:  
 Total number of requests: 2  
 Number of requests succeeded: 2  
 Number of requests timed out: 0  
 Avg usage time: 1.921500 Seconds  
 Min usage time: 1.734000 Seconds  
 Max usage time: 2.109000 Seconds  
 Sum usage time: 3.843000 Seconds

C:\Program Files\ArcGIS\log\Server-20050719-175021.xml - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Home Search Favorites Media

Address C:\Program Files\ArcGIS\log\Server-20050719-175021.xml Go Links

2005-07-19T17:54:09	Detailed	SanDiego.MapServer	Server Context created.	nelson02	3144	1.59999999999911E-02	4006
2005-07-19T17:54:11	Detailed	SanDiego.MapServer	Server Context released.	nelson02	3916	1.734000000000001	4007
2005-07-19T17:54:11	Detailed	SanDiego.MapServer	Server Context created.	nelson02	3256	0	4006
2005-07-19T17:54:13	Detailed	SanDiego.MapServer	Server Context released.	nelson02	3868	2.108999999999998	4007

Done My Computer

# Log File Behavior

- Modify Server.cfg in folder <install directory>/cfg
  - *C:\Program Files\ArcGIS\cfg\Server.cfg*

<Properties>

<LogPath> C:\Program Files\ArcGIS\log </LogPath>

<LogSize>10</LogSize>

<LogLevel>5</LogLevel>

</Properties>

# Quantifying Web Performance

- Client Response Time
  - Definition: How “fast” does web application appear to remote browser hitting the site
  - Measured via TTFB (time to first byte)
  - Measured via TTLB (time to last byte)
- Machine Throughput
  - Definition: How many client requests can a server handle under load
  - Measured in number of requests/sec
  - Impacts number of servers you need to buy

# Visual Studio 2005 Team System

- Tester tools are now on par with Developer Tools
- Unit, Web, Load, and Manual tests
- Measures code coverage

