Design for Scalability

Robin Qiu, Ph.D.
Division of Engineering
Pennsylvania State University - Great Valley
Malvern, PA 19355
Email: RobinQiu@PSU.EDU
Outline

- Needs for e-Business
- Scalability Definition
- Design for Scalability
  - Know the work load
  - Techniques
The success of e-business depends on organization’s ability to design and implement an infrastructure that yields the measures of high performance, availability, and reliability expected to support the business objectives for revenue and customer satisfaction.

The infrastructure consists of hardware, software, and network components you select for their ability to meet your needs of today and tomorrow.
Scalability

- Most e-businesses face the same challenges, the most significant of which are unpredictable growth and the ability to have solutions ready for unknown problems.

- Scalability is related to the features of performance (response time) and capacity (operations per unit of time).
Scalability Definition

Scalability refers to a component’s ability to adapt readily to a greater or lesser intensity of use, volume, or demand while still meeting business objectives.
Know Your Workload

Some typical Web site loads over a 24-hour period
Know Your Workload

Example of a retail site with seasonal peaks, growing from year to year
Know Your Workload

Examples of metrics for page hits per day
# Workload Patterns and Web Classification

<table>
<thead>
<tr>
<th>Category / Example</th>
<th>Publish / Subscribe</th>
<th>Online Shopping</th>
<th>Customer Self-Service</th>
<th>Trading</th>
<th>Web Services / B2B</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Search engines</strong></td>
<td></td>
<td>Exact inventory</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Media</strong></td>
<td></td>
<td>Inexact inventory</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Events</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Home banking</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Online stock trading</td>
</tr>
<tr>
<td><strong>Package tracking</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Auctions</td>
</tr>
<tr>
<td><strong>Travel arrangements</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Catalog</strong></td>
<td></td>
<td></td>
<td>Data is in legacy applications</td>
<td>Extremely time sensitive</td>
<td></td>
</tr>
<tr>
<td><strong>Either flat</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>High volatility</td>
</tr>
<tr>
<td><strong>(parts catalog)</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Multiple suppliers, multiple consumers</td>
</tr>
<tr>
<td><strong>or dynamic</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Transactions are complex and interact with back end</td>
</tr>
<tr>
<td><strong>(items change frequently, near real time)</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Few page authors and page layout changes less frequently.</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>User specific information: user profiles with data mining.</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Data is in legacy applications</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Multiple data sources, requirement for consistency</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Extremely time sensitive</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>High volatility</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Multiple suppliers, multiple consumers</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Transactions are complex</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Transactions are complex.</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Content**

- Dynamic change of the layout of a page, based on changes in content, or need.
- Many page authors and page layout changes frequently.
- High volume, non user specific access.
- Fairly static information sources.
## Workload Patterns and Web Classification (cont’d)

<table>
<thead>
<tr>
<th></th>
<th>Publish / Subscribe</th>
<th>Online Shopping</th>
<th>Customer Self-Service</th>
<th>Trading</th>
<th>Web Services / B2B</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Security</strong></td>
<td>Low</td>
<td>Privacy, nonrepudiation, integrity, authentication, regulations</td>
<td>Privacy, nonrepudiation, integrity, authentication, regulations (Banking); Low for others</td>
<td>Privacy, nonrepudiation, integrity, authentication, regulations</td>
<td>Privacy, nonrepudiation, integrity, authentication, regulations</td>
</tr>
<tr>
<td><strong>Percent Secure Pages</strong></td>
<td>Low</td>
<td>Medium</td>
<td>Medium</td>
<td>High</td>
<td>Medium</td>
</tr>
<tr>
<td><strong>Cross-session Info</strong></td>
<td>No</td>
<td>High</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>Searches</strong></td>
<td>Structured by category</td>
<td>Structured by category</td>
<td>Structured by category</td>
<td>Structured by category</td>
<td>Structured by category</td>
</tr>
<tr>
<td></td>
<td>Totally dynamic</td>
<td>Totally dynamic</td>
<td>Low volume</td>
<td>Low volume</td>
<td>Low volume</td>
</tr>
<tr>
<td></td>
<td>Low volume</td>
<td>High volume</td>
<td></td>
<td></td>
<td>Low to moderate volume</td>
</tr>
</tbody>
</table>

10
Characterizing Your Workload

<table>
<thead>
<tr>
<th></th>
<th>Publish / Subscribe</th>
<th>Online Shopping</th>
<th>Customer Self-Service</th>
<th>Trading</th>
<th>Web Services / B2B</th>
<th>Your Workload</th>
</tr>
</thead>
<tbody>
<tr>
<td>High Volume, Dynamic, Transactional, Fast Growth</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Volume of User-specific Responses</td>
<td>Low</td>
<td>Low</td>
<td>Medium</td>
<td>High</td>
<td>Medium</td>
<td></td>
</tr>
<tr>
<td>Amount of Cross-session Information</td>
<td>Low</td>
<td>High</td>
<td>High</td>
<td>High</td>
<td>High</td>
<td></td>
</tr>
<tr>
<td>Volume of Dynamic Searches</td>
<td>Low</td>
<td>High</td>
<td>Low</td>
<td>Low</td>
<td>Medium</td>
<td></td>
</tr>
<tr>
<td>Transaction Complexity</td>
<td>Low</td>
<td>Medium</td>
<td>High</td>
<td>High</td>
<td>High</td>
<td></td>
</tr>
<tr>
<td>Transaction Volume Swing</td>
<td>Low</td>
<td>Medium</td>
<td>Medium</td>
<td>High</td>
<td>High</td>
<td></td>
</tr>
<tr>
<td>Data Volatility</td>
<td>Low</td>
<td>Low</td>
<td>Low</td>
<td>High</td>
<td>Medium</td>
<td></td>
</tr>
<tr>
<td>Number of Unique Items</td>
<td>High</td>
<td>Medium</td>
<td>Low</td>
<td>Medium</td>
<td>Medium</td>
<td></td>
</tr>
<tr>
<td>Number of Page Views</td>
<td>High</td>
<td>Medium</td>
<td>Low</td>
<td>Medium</td>
<td>Medium</td>
<td></td>
</tr>
<tr>
<td>Percent Secure Pages (privacy)</td>
<td>Low</td>
<td>Medium</td>
<td>Medium</td>
<td>High</td>
<td>High</td>
<td></td>
</tr>
<tr>
<td>Use of Security (authentication, Integrity, nonrepudiation)</td>
<td>Low</td>
<td>High</td>
<td>High</td>
<td>High</td>
<td>High</td>
<td></td>
</tr>
<tr>
<td>Other Characteristics</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>High</td>
</tr>
</tbody>
</table>
Determine the Component Most Affected

<table>
<thead>
<tr>
<th></th>
<th>Edge Servers</th>
<th>Web Presentation Server</th>
<th>Web Application Server</th>
<th>Security Servers</th>
<th>Transaction Servers</th>
<th>Data Servers</th>
<th>Network</th>
</tr>
</thead>
<tbody>
<tr>
<td>High volume, dynamic, transactional, fast growth</td>
<td>High</td>
<td>High</td>
<td>High</td>
<td>High</td>
<td>High</td>
<td>High</td>
<td>High</td>
</tr>
<tr>
<td>High percent user-specific responses</td>
<td>Low</td>
<td>Low</td>
<td>High</td>
<td>Low</td>
<td>High</td>
<td>High</td>
<td>Low</td>
</tr>
<tr>
<td>High percent cross-session information</td>
<td>Low</td>
<td>Med</td>
<td>High</td>
<td>Low</td>
<td>Low</td>
<td>Med</td>
<td>Low</td>
</tr>
<tr>
<td>High volume of dynamic searches</td>
<td>Med</td>
<td>High</td>
<td>High</td>
<td>Low</td>
<td>Med</td>
<td>High</td>
<td>Med</td>
</tr>
<tr>
<td>High transaction complexity</td>
<td>Low</td>
<td>Low</td>
<td>High</td>
<td>Med</td>
<td>High</td>
<td>High</td>
<td>Low</td>
</tr>
<tr>
<td>High transaction volume (swing)</td>
<td>Low</td>
<td>Med</td>
<td>High</td>
<td>Low</td>
<td>High</td>
<td>High</td>
<td>Low</td>
</tr>
<tr>
<td>High data volatility</td>
<td>Low</td>
<td>High</td>
<td>High</td>
<td>Low</td>
<td>Med</td>
<td>High</td>
<td>Low</td>
</tr>
<tr>
<td>High number unique items</td>
<td>Low</td>
<td>Low</td>
<td>Med</td>
<td>Low</td>
<td>High</td>
<td>High</td>
<td>Low</td>
</tr>
<tr>
<td>High number page views</td>
<td>High</td>
<td>High</td>
<td>Low</td>
<td>Low</td>
<td>Low</td>
<td>Low</td>
<td>High</td>
</tr>
<tr>
<td>High percent secure pages (privacy)</td>
<td>Low</td>
<td>High</td>
<td>Low</td>
<td>High</td>
<td>Low</td>
<td>Low</td>
<td>Low</td>
</tr>
<tr>
<td>High security</td>
<td>Low</td>
<td>High</td>
<td>High</td>
<td>High</td>
<td>High</td>
<td>Low</td>
<td>Low</td>
</tr>
</tbody>
</table>
Basic Objectives of Scalability

- Increase the capacity or speed of the component.
- Improve the efficiency of the component/system.
- Shift or reduce the load on the component.
Component vs. System

As one increases the scalability of one component, the result may change the dynamics of the site service, thereby moving the "hot spot" or bottleneck to another component. The scalability of the infrastructure depends on the ability of each component to scale to meet increasing demands.
Scalability vs. Performance

![Graph showing scalability vs. performance. The x-axis represents the number of operations, and the y-axis represents response time. The graph compares the current load to the scaling target, highlighting the acceptable response time against the original and improved performance curves.](image-url)
Examples of Techniques

- If your objective is to increase the speed of a component, you would consider using a faster or special machine and/or creating a machine cluster.
- If a large number of requests can be handled using a cache instead of the database, the overall load on the database is reduced, affording greater scalability for the entire system.
- Etc.
### Examples of Techniques

<table>
<thead>
<tr>
<th>ID</th>
<th>Scaling Technique</th>
<th>Increase Capacity/Speed</th>
<th>Improve Efficiency</th>
<th>Shift / Reduce Load</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Use a faster machine</td>
<td>x</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Create a machine cluster</td>
<td>x</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Use a special machine</td>
<td>x</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Segment the workload</td>
<td></td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>5</td>
<td>Batch requests</td>
<td></td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Aggregate user data</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Manage connections</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Cache data and requests</td>
<td></td>
<td>x</td>
<td>x</td>
</tr>
</tbody>
</table>

**Systematic Approach**
Design for Scalability

1. Understand the application environment

   - Knowing your workload pattern (publish/subscribe and customer self-service, for example) determines where to focus your scalability efforts, and which scaling techniques to apply.
Design for Scalability

1. Understand the application environment

   For existing environments, the first step is to identify all components and understand how they relate to each other. The most important task is to understand the requirements and flow of the existing application(s) and what can or cannot be altered. The application is key to the scalability of any infrastructure, so a detailed understanding is mandatory to scale effectively. At a minimum, your analysis must include a breakdown of transaction types and volumes as well as a graphic view of the components in each tier.
Component Latency Analysis
Design for Scalability

2. Categorize your workload

- All site types, like yours, are considered to have high-volumes of dynamic transactions. Your site type will become clear as you evaluate your site for the other characteristics that pertain to transaction complexity, volume swings, data volatility, security, and others.

  - Characterizing workload pattern
Design for Scalability

3. Determine architecture
   - Networks
   - Software technology
   - Hardware
   - Component network mapping
   - Relationship
   - Deployment and maintainability
Design for Scalability -- Architecture

Architectural choices typically driven by:
- Security, isolation from the open Internet
- Geographical considerations - separate data centers
- Separation of function (presentation and business logic)

* Firewalls between tiers
4. Determine the components most affected

- This step involves mapping the most important site characteristics to each component. Once again, from a scalability viewpoint, the key components of the infrastructure are the edge servers, the Web application servers, security services, transaction and data servers, and the network.

- *Determining the components most affected*
Design for Scalability

5. Select the scaling techniques to apply to scale the workload

- Use a faster machine
  - This technique applies to the edge servers, the Web presentation server, the Web application server, the directory and security servers, the existing transaction and data servers, the network, and the Internet firewall.
Design for Scalability

5. Select the scaling techniques to apply to scale the workload

- Create a cluster of machines
  - This technique applies to the Web presentation server, the Web application server, and the directory and security servers.
  - Parallelism in machine clusters typically leads to improvements in response time. Also, system availability is improved due to failover safety in replicas.
Design for Scalability

5. Select the scaling techniques to apply to scale the workload

- Use appliance servers
  - This technique applies to the edge servers, the Web presentation server, the directory and security servers, the network, and the Internet firewall.
  - The goal is to improve the efficiency of a specific component by using a special purpose machine to perform the required action.
  - These machines tend to be dedicated machines that are very fast and optimized for a specific function.
  - Examples are network appliances and routers with cache, such as the IBM WebSphere Edge Server.
Design for Scalability

5. Select the scaling techniques to apply to scale the workload
   • Segment the workload
     • This technique applies to the Web presentation server, the Web application server, the data server, the intranet firewall, and the network.
     • The goal is to split up the workload into manageable chunks thereby obtaining more consistent and predictable response time.
5. Select the scaling techniques to apply to scale the workload

- Segment the workload

  - This technique applies to the Web presentation server, the Web application server, the data server, the intranet firewall, and the network.

  - The goal is to split up the workload into manageable chunks thereby obtaining more consistent and predictable response time.
5. Select the scaling techniques to apply to scale the workload

- **Batch requests**
  - This technique applies to the Web presentation server, the Web application server, the directory and security servers, the existing business applications, and the database.
  - The goal is to reduce the number of requests sent between requesters and responders (such as between tiers or processes) by allowing the requester to define new requests that combine multiple requests.
5. Select the scaling techniques to apply to scale the workload

- Aggregate user data
  - This technique applies to the Web presentation server, the Web application server, and the network.
  - The goal is to allow rapid access to large customer data controlled by existing system applications and support personalization based on customer specific data.
5. Select the scaling techniques to apply to scale the workload

- **Manage connections**
  - This technique applies to the Web presentation server, the Web application server, and the database.
  - The goal is to minimize the number of connections needed for an end-to-end system, as well as to eliminate the overhead of setting up connections during normal operations.
Design for Scalability

5. Select the scaling techniques to apply to scale the workload

- Cache
  - Caching applies to the edge server, the Web presentation server, the Web application server, the network, the existing business applications, and the database.
  - The goal is to improve the performance and scalability by reducing the length of the path traversed by a request and the resulting response, and by reducing the consumption of resources by components.
Scalable Solution - WebSphere
Additional Techniques

- One of the most important techniques for high-volume sites with complex transaction workloads is to push as much work as possible toward the network.
  - In general, the use of caching provides the quickest response time and lowers the system load since the network cached pages can be served directly to users without a hit to the Web server.
  - Akamai, WebEx, etc.
Additional Techniques

- Other practices considered most significant are flattening of product catalogs and fast-path shop design.
  - e.g., catalog pages are flattened into static Web pages that can be cached like other static pages for the event site. For every 100 browsers on the site, only 1 to 5 actually buy anything. 95% of the time, browsing is done without accessing the catalog database. The frequently accessed pages are even cached out in the network so the Web server load is lower too.
Typical Web-Based Info System
## Computing Power Scale Up

<table>
<thead>
<tr>
<th>FRONT END</th>
<th>MID-TIER</th>
<th>BACK END</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Front End Diagram" /></td>
<td><img src="image" alt="Mid-Tier Diagram" /></td>
<td><img src="image" alt="Back End Diagram" /></td>
</tr>
</tbody>
</table>

### Scale Up
- IA-based clusters
- 32-way systems
- 16-way systems
- 8-way systems
- 4-way systems
- 2-way systems

### Scale Out
- More servers
Example – IBM WebSphere Edge Server

Version 2.0 Features includes:

- **Application Offload** shifts the burden of serving composed, personalized, dynamic content from the application server to Edge Servers placed at network edges by offloading back-end servers and peering links.

- **Content Distribution** deploys published Web content to caches and "rehosting servers" throughout the network.

- **Enhanced Caching** improves response time by offloading back-end servers and peering links as a forward, reverse, or transparent proxy. Cache and invalidate dynamic content generated by WebSphere Application Server including Java (TM) Server Pages (JSPs) components and servlets.
Version 2.0 Features includes (cont’d):

- **Enhanced Load Balancing** improves server selection, load optimization, and fault tolerance.

- **Security** ensures only authorized users can access cached and non-cached Web resources. This integrated solution helps ensure control at the edge of your network.

- **Transactional Quality of Service** allocates computing and network resources according to a transaction's business value and interoperates with networking hardware. Prioritize and add security to traffic based on user classes and groups, optimizing resource utilization and improving network scalability.
Example – WebLogic Web Clustering
Example – WebLogic Web Clustering
Optimizing Component for Maintainability

- Good tools
- Coding standards
- No fancy tricks
- No assumptions
- Documentation
- No incompatible codes
Conclusions

- Design information systems for
  - Scalability
    - Hardware
    - Software
    - Networks
  - Integration
  - etc.