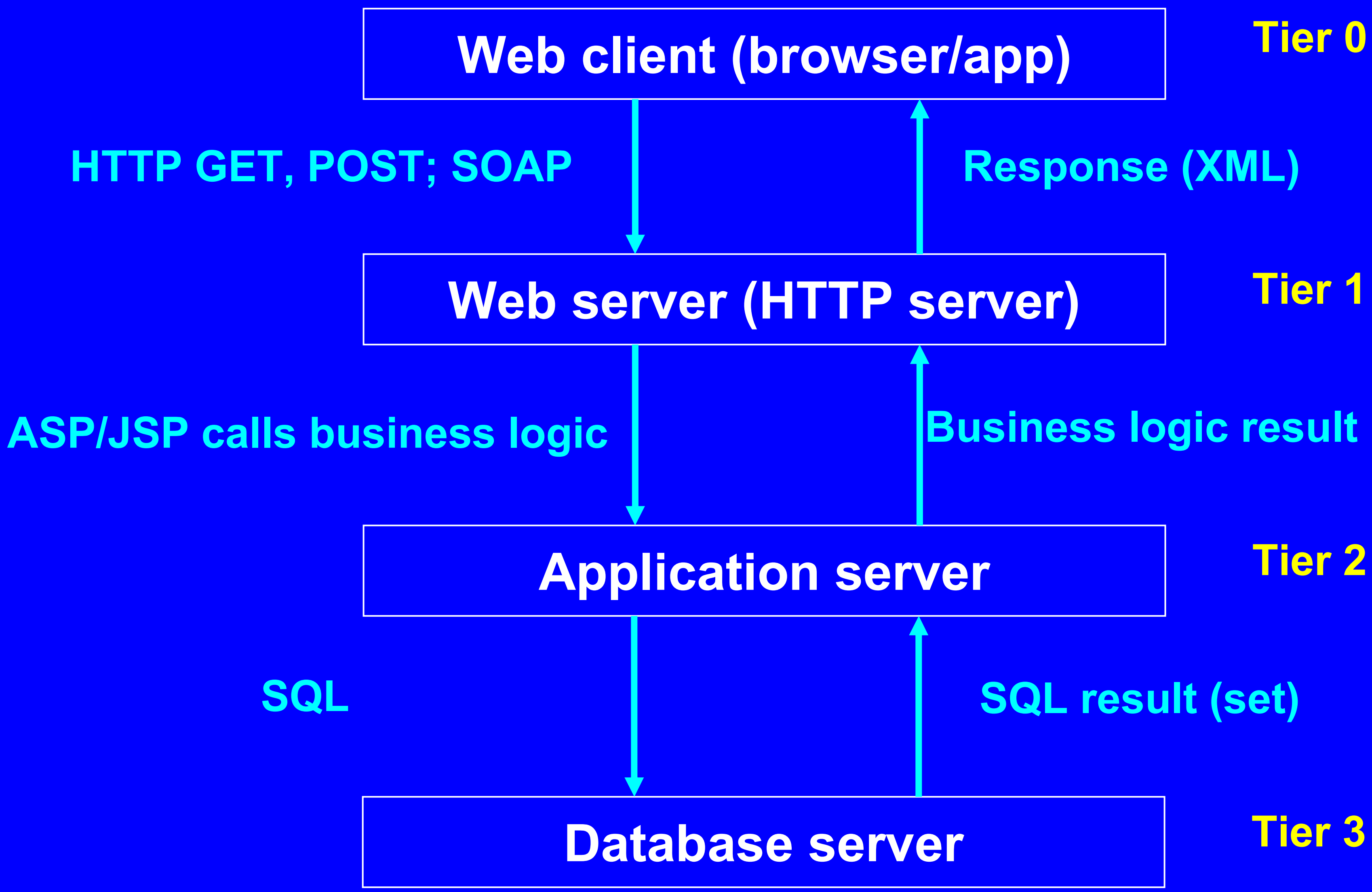
1.264 Lecture 19

System architecture

Intro to telecommunications

Architecture example

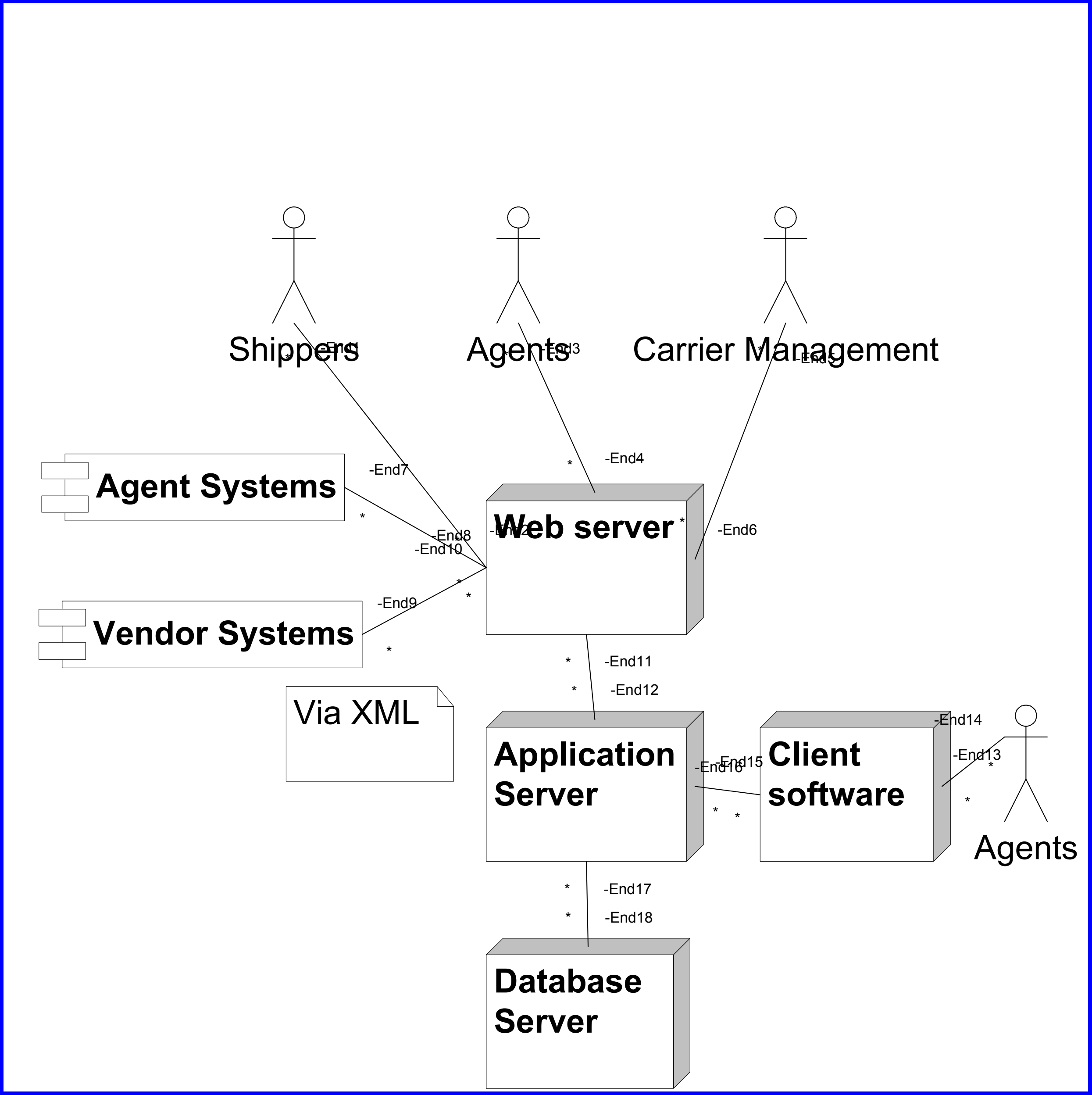


In HW7-9, your ASP pages include the business logic that is usually in the application server

Architecture example

- Example application reserves space on intermodal carrier trips (vessel, rail, …). Used by people:
  - Agents to sell freight transportation
  - Shippers to view trip schedules and details
  - Carrier management to manage vehicle and trip data
- Application is accessed by three mechanisms:
  - Web interface (shippers, agents, carrier management)
  - Standalone application (agents)
  - Other systems:
    - External agent systems
    - Vehicle maintenance companies that need physical data (vehicle type, fuel capacity, fuel type, …)

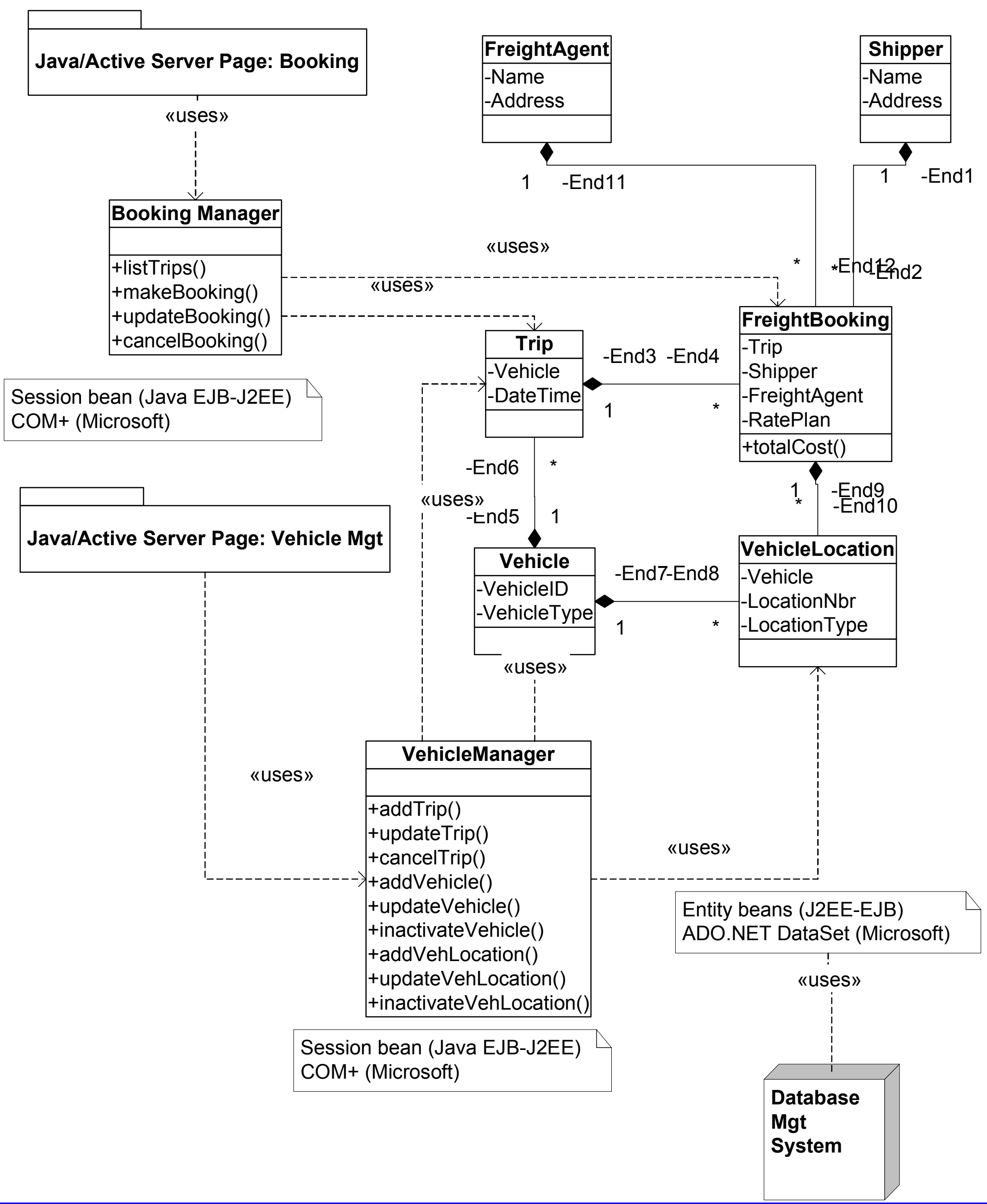
UML Diagram: Use Case



UML Diagram: Class Diagram

Web server

App server (session or business logic beans)



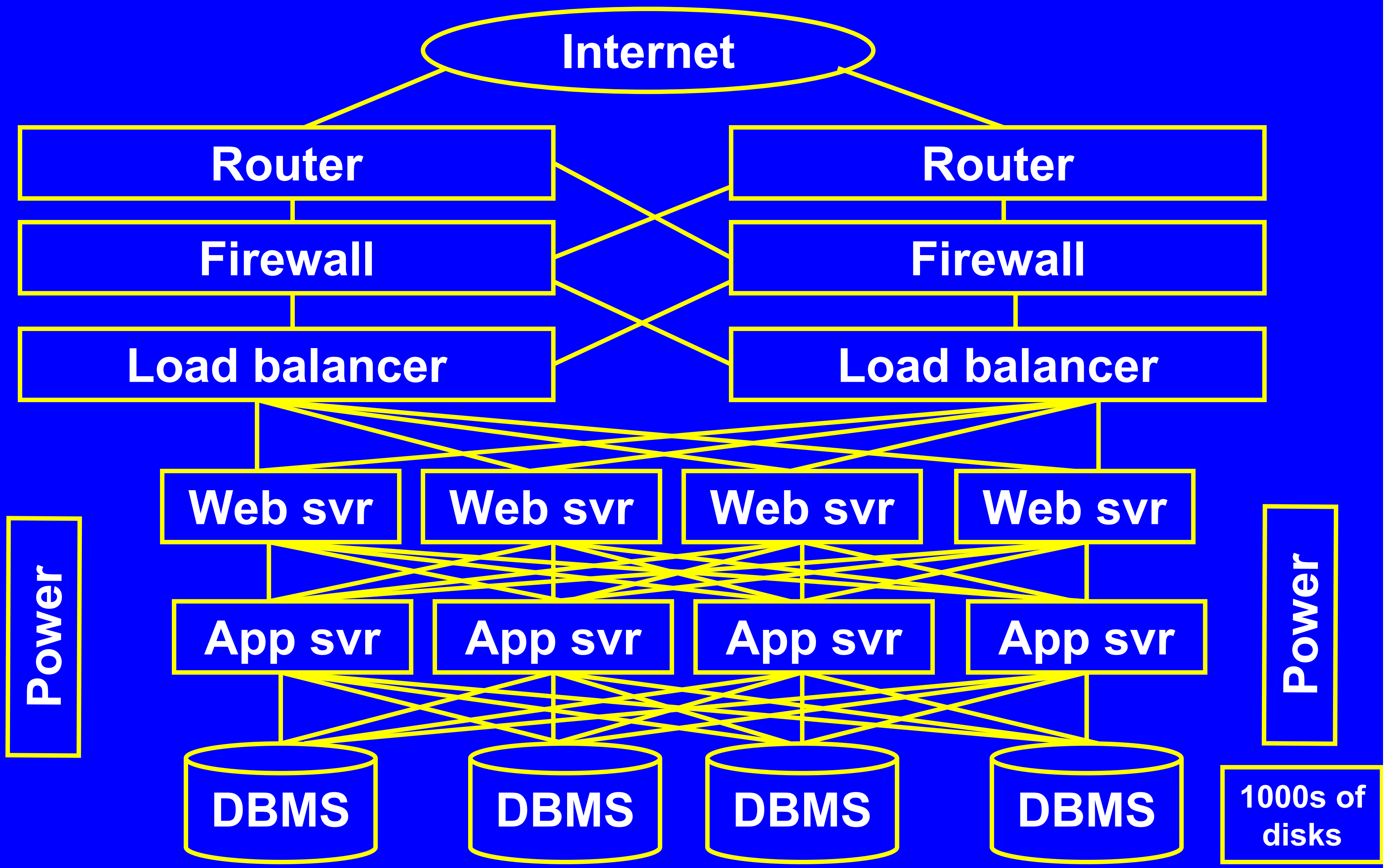
App server (entity or db beans)

Database server

Server-side architecture components

- Network interface architecture components
  - Router
  - Firewall
  - Load balancer
- Application components
  - Web server
  - Application server
  - Database server
  - (These components are managed within the Java EE 5 or Microsoft .NET frameworks)

Architecture components at ‘server’



Router, Firewall, Load Balancer

- Router routes IP (TCP/IP and UDP/IP) packets
  - Routing table: either send packet to local machine or to another router
- Firewall examines all packets
  - Checks IP address, port
  - Can extract and store contents from all Internet protocols if suspicious packets are seen, especially HTTP/SOAP these days
  - Has security language to set traffic rules
- Load balancer sends HTTP packets to Web servers
  - Balances based on HTTP header: client IP, cookie, …
  - Many inexpensive servers function as one large single virtual server
  - Use with multiple heterogeneous hardware platforms
  - Add new servers, and sites without interruption to service
  - Google runs thousands of Linux servers to index Web

Web server, App server, Database server

- **Web server:**
  - **Handles HTTP requests and responses**
  - **Reads and writes XML and HTML**
- **Application server:**
  - **Runs programs that implement business logic. These are processes, while the database stores entities.**
  - **Processes tend to be simple, modular.**
  - **Designed from UML use cases, sequence diagrams… E.g.,**
    - **Order entry; receiving, shipping, payment, engineering quotes…**
  - **Think of it as a pre-written main program with many pre-written components for common tasks**
- **Database server:**
  - **Persistent storage of data**
  - **Sharing of data across applications and organizations**
  - **Repository of business rules, within database and to guide XML document exchange**

System architecture and configuration

- Done early in system development or configuration project
  - Assess users, applications, system software, networks, hardware
  - Configurations are complex and changing
  - Information is almost always wrong on which estimates are based
- Successful configurations are usually overbuilt
- Organization of all system components (hardware, software, network) is the <u>system architecture</u>
- Don’t forget about development, staging, QA servers/network
  - Spend your entire budget, always!
- Each server does just one thing:
  - HTTP server (Web server)
  - Session logic (app server)
  - SQL (database server)
  - We can understand and characterize its task this way. If a box handles many functions, sizing and managing it is impossible

Virtual Storage Hierarchy

Level	Speed	Size
Registers	1 ns	1KB per CPU
Cache (L1, L2)	1-3ns	64KB-2MB per CPU
Main memory	5-70ns	30GB per system
Disk storage	10,000,000ns (10-15 millisec)	10TB per system
Network virtual memory	100,000,000ns (100 millisec)	10TB per network (LAN)

Each level caches for the one above it, with 95%+ hit rate

Even 1-2% misses degrade performance: lost cycles

Levels go from fast, expensive, small -> slow, cheap, large

Virtual Storage Hierarchy

- Servers are not just big PCs; they are organized differently
- Memory is used to stage data to the processor in servers
  - Memory use for program execution is secondary
- Cache is used for program execution. Many programs have 1MB footprint, so 2MB cache allows OS and database to run, for example.
- Disk is used only for persistence
  - Disks are 1,000,000+ times slower than memory and are the bottleneck
  - Servers use many small disks (rather than single large disks as on a PC) for performance
  - RAID: Redundant array of independent disks
  - Use RAID 1+0 (mirror+stripe), almost never RAID-0, -1 or -5
- Why does all this matter?
  - Applications grow big quickly if successful.
  - Many companies have had big crashes (Ebay, others)

Performance metrics

- **Metrics (database server as an example)**
  - **Throughput:** I/O operations/second, data transfer rate/second
    - This is system view
    - If throughput is low, must buy more servers than really needed
  - **Latency**: seek time (seconds), response time (seconds)
    - This is user view
    - If latency is high, users leave your site (8 seconds max)
  - **Utilization:** percentage of data transfer rate, disk capacity used
    - This is future view
    - If avg utilization > 60%, your systems will crawl on busy days
  - **Efficiency**: can system actually meet its theoretical limits?
    - Is the software written well enough?
    - Will the 20 user database really support 20 users on your box?
- **Units**
  - Ethernet data transfer: 100 Mbit/sec = 12.50 MB/sec
  - Disk data transfer: 10.00 MB/sec
  - Modem data transfer: 56,000 bits/sec= 0.007 MB/sec
- **Use comparable units when configuring systems!**

Exercises

- Discuss:
  - Name the three layers of a three tier architecture
    - What layers are usually missing or combined in two tier architecture?
  - What does a load balancer do?
    - What happens if you don’t have a load balancer and a Web server goes down? How do you remedy the situation?
  - Why do we still use mechanical disks?
  - If we have a ‘fast’ 100 Mbps Ethernet LAN and two ‘slow’ disks with a data transfer rate of 10 MBps, where is the bottleneck?
    - How would you relieve the bottleneck?

Solution

- Discuss:
  - Three tier architecture: Web, application, database svrs
    - What layers are missing or combined in two tier architecture? Web and application are combined
  - What does a load balancer do? Directs HTTP packets to Web servers
    - What happens if you don't have a load balancer and a Web server goes down? How do you remedy the situation?
    - Users lose service abruptly. You reconfigure your router to not use the link/hop to the down server; users restart their transaction
  - Why do we still use mechanical disks?
    - Persistence; all other storage loses state if it loses power (though flash memory is becoming feasible)
    - Cheapness for large amounts of storage
  - If we have a 'fast' 100 Mbps Ethernet LAN and two 'slow' disks with a data transfer rate of 10 MBps, where is the bottleneck?
    - 2 x 10 Mbps x 8 bits/byte = 160 Mbps > 100 Mbps of the LAN. LAN is bottleneck
    - How would you relieve the bottleneck? Upgrade to 1 Gbps LAN

Introduction to the telecom network

- **Pre-1990:**
  - **Over 90% of telecom traffic is voice**
- **1990-2000:**
  - **Data becomes over 90% of traffic**
- **2005:**
  - **App-app and object-object communications becomes 90% of data traffic**
- **2006-2015:**
  - **Billions of sensors, smart cards, cameras, terminals, …**
    - **Devices (e.g. home appliances) self-diagnosed, controlled over network**
    - **Product level RFID tags for supply chain, consumer operation**
    - **Pervasive environmental, traffic, energy, other sensors accessed via Web**
    - **Broadband wireless widespread**
    - **Fiber optics to most residences and businesses**
    - **Converged network for voice, data, video**
    - **Web servers (in devices), Web services pervasive**
    - **Pervasive video surveillance (http://www.mediaeater.com/cameras/locations.html)**

Telecom future motivation

- Telecom in supply chain, transport, manufacturing:
  - Supply chain, transportation, manufacturing, other applications will have machine-machine or object-object communications via HTTP and XML
  - Data will be stored in relational databases, which will become more complex over time
  - Wireless and wireline communications, using a variety of technologies will tie the ‘communications skin’ together
  - The information delivery must be as well designed as physical delivery: capacity, integration, performance, quality, reliability…
- You will need to configure wireless and wireline networks to access data, sensors and applications
  - Manufacturing facilities, public transport systems, construction sites, over-the-road transport all present unique challenges

Central issue: network access ("dial tone")

- Who can use the network, and how easy is it?
- Voice:
  - Voice dial tone is universal access to network
  - Reliable, universal, standard (though poor) user interface
  - Supports many advanced services, is a model for data services in many ways
- Data:
  - World Wide Web, browsers are the first version of data dial tone, after many years of research and experimentation
  - Data dial tone not truly reliable, becoming widely available (especially wireless), standard and good user interface
  - Advanced services (e.g. bandwidth management, security, etc.) are flawed
  - Issues: PCs need to be always on, net needs to always be connected, device needs to be easier to use and manage than a PC
- Video:
  - Video dial tone still remains to be worked out
  - Interactive video not workable; streaming video problematic
  - Awaits fiber optic networks or broadband wireless to end users

Central issue: convergence

- Many separate networks currently exist:
  - Consumer voice (circuit-switched, VOIP)
  - Corporate voice (circuit-switched, PBXs, private nets)
  - Consumer data (Internet protocols)
  - Corporate data (private nets, frame relay, ATM, Internet)
  - Consumer video (cable TV, satellite)
  - Corporate video (satellite, private circuits)
  - Telecom carriers (private nets)
  - Governments (private nets)
- In the future, most of these (except secure government nets) will be on the Internet backbone
  - Moving your organization or project into this world will be necessary, and hard
    - Can move too quickly or too slowly
    - Can choose right or wrong technologies, carriers
    - Can choose right or wrong application strategy (centralized or not..)