Microsoft rolled out a 64 bit version of its Windows operating systems on Monday. As compared with existing 32-bit versions, 64-bit Windows will handle 16 terabytes of virtual memory, as compared to 4 GB for 32-bit Windows. System cache size jumps from 1 GB to 1 TB, and paging-file size increases from 16 TB to 512 TB.

### Protocol Family Concept

- **Key to protocol families** is that communication occurs **logically** at the same level of the protocol, called **peer-to-peer**... but is implemented via services at the next lower level.
- **Encapsulation**: carry higher level information within lower level “envelope”
- **Fragmentation**: break packet into multiple smaller packets and reassemble

### Protocol for Network of Networks

- **Transmission Control Protocol/Internet Protocol (TCP/IP)**
  - This protocol family is the **basis of the Internet**, a WAN protocol
  - IP makes best effort to deliver
  - TCP guarantees delivery
  - TCP/IP so popular it is used even when communicating locally: even across homogeneous LAN

### TCP/IP packet, Ethernet packet, protocols

- Application sends message
- TCP breaks into 64KIB segments, adds 20B header
- IP adds 20B header, sends to network
- If Ethernet, broken into 1500B packets with headers, trailers (24B)
- All Headers, trailers have length field, destination,

### Overhead vs. Bandwidth

- Networks are typically advertised using peak bandwidth of network link: e.g., 100 Mbits/sec Ethernet (“100 base T”)
- Software overhead to put message into network or get message out of network often limits useful bandwidth
- Assume overhead to send and receive = 320 microseconds (\(\mu s\)), want to send 1000 Bytes over “100 Mbit/s” Ethernet
  - Network transmission time: 1000Bx8b/B / 100Mb/s = 8000b / (1000b/\(\mu s\)) = 80 \(\mu s\)
  - Effective bandwidth: 8000b/(320+80)\(\mu s\) = 20 Mb/s
Magnetic Disks

- **Purpose:**
  - Long-term, nonvolatile, inexpensive storage for files
  - Large, inexpensive, slow level in the memory hierarchy (discuss later)

**Processor** (active)

**Computer**

**Control** (brain)

**Datapath** (brawn)

**Memory**

(passive) where programs, data live when running

**Devices**

Input

Output

**Devices**

Mouse

Keyboard

Display

Network

**Disk**

**Printer**

**Actuator**

Moves head (end of arm) over track (seek), wait for sector rotate under head, then read or write

**Platter**

Several platters, with information recorded magnetically on both surfaces (usually)

**Tracks**

Bits recorded in tracks, which in turn divided into sectors (e.g., 512 Bytes)

**Inner Track**

**Outer Track**

**Sector**

**Head**

**Arm**

**Controller**

**Spindle**

**Actuator**

**Disk Device Terminology**

**Disk Device Performance**

**Outer Track**

**Inner Sector**

**Head**

**Arm**

**Controller**

**Disk Latency = Seek Time + Rotation Time + Transfer Time + Controller Overhead**

- Seek Time depends on tracks move arm, seek speed of disk
- Rotation Time depends on speed disk rotates, how far sector is from head
- Transfer Time depends on data rate (bandwidth) of disk (bit density), size of request

**Data Rate: Inner vs. Outer Tracks**

- To keep things simple, originally same # of sectors/track
  - Since outer track longer, lower bits per inch
- Competition decided to keep bits/inch (BPI) high for all tracks ("constant bit density")
  - More capacity per disk
  - More sectors per track towards edge
  - Since disk spins at constant speed, outer tracks have faster data rate
  - Bandwidth outer track 1.7X inner track!

**Disk Performance Model /Trends**

- **Capacity**: + 100% / year (2X / 1.0 yrs)
  - Over time, grown so fast that # of platters has reduced (some even use only 1 now!)
- **Transfer rate (BW)**: + 40%/yr (2X / 2 yrs)
- **Rotation+Seek time**: – 8%/yr (1/2 in 10 yrs)
- **Areal Density**
  - Bits recorded along a track: Bits/Inch (BPI)
  - # of tracks per surface: Tracks/Inch (TPI)
  - We care about bit density per unit area Bits/Inch²
  - Called Areal Density = BPI x TPI
- **MB$/**: > 100%/year (2X / 1.0 yrs)
  - Fewer chips + areal density

**Historical Perspective**

- Form factor and capacity drives market, more than performance
  - 1970s: Mainframes ⇒ 14" diam. disks
  - 1980s: Minicomputers, Servers ⇒ 8", 5.25" diam. disks
  - Late 1980s/Early 1990s:
    - Pizzabox PCs ⇒ 3.5 inch diameter disks
    - Laptops, notebooks ⇒ 2.5 inch disks
    - Palmtops didn’t use disks, so 1.8 inch diameter disks didn’t make it
Use Arrays of Small Disks...

- Katz and Patterson asked in 1987: Can smaller disks be used to close gap in performance between disks and CPUs?

Conventional: 4 disk designs

Disk Array: 1 disk design

Replace Small Number of Large Disks with Large Number of Small Disks! (1988 Disks)

<table>
<thead>
<tr>
<th>Capacity</th>
<th>IBM 3390K</th>
<th>IBM 3.5&quot; 0061</th>
<th>x70</th>
</tr>
</thead>
<tbody>
<tr>
<td>Volume</td>
<td>20 GBytes</td>
<td>320 MBytes</td>
<td>23 GBytes</td>
</tr>
<tr>
<td>Power</td>
<td>11 cu. ft.</td>
<td>0.1 cu. ft.</td>
<td>11 cu. ft. 9X</td>
</tr>
<tr>
<td>Data Rate</td>
<td>15 MB/s</td>
<td>1.5 MB/s</td>
<td>120 MB/s 8X</td>
</tr>
<tr>
<td>I/O Rate</td>
<td>600 I/Os/s</td>
<td>55 I/Os/s</td>
<td>3900 I/Os/s 6X</td>
</tr>
<tr>
<td>MTTF</td>
<td>250 Khrs</td>
<td>50 Khrs</td>
<td>??? Hrs</td>
</tr>
<tr>
<td>Cost</td>
<td>$250K</td>
<td>$2K</td>
<td>$150K</td>
</tr>
</tbody>
</table>

Disk Arrays potentially high performance, high MB per cu. ft., high MB per KW, but what about reliability?

Array Reliability

- Reliability - whether or not a component has failed
  - measured as Mean Time To Failure (MTTF)
- Reliability of N disks = Reliability of 1 Disk / N (assuming failures independent)
  - 50,000 Hours / 70 disks = 700 hour
- Disk system MTTF: Drops from 6 years to 1 month!
- Disk arrays too unreliable to be useful!

Redundant Arrays of (Inexpensive) Disks

- Files are “striped” across multiple disks
- Redundancy yields high data availability
  - Availability: service still provided to user, even if some components failed
- Disks will still fail
- Contents reconstructed from data redundantly stored in the array
  - Capacity penalty to store redundant info
  - Bandwidth penalty to update redundant info

Berkeley History, RAID-I

- RAID-I (1989)
  - Consisted of a Sun 4/280 workstation with 128 MB of DRAM, four dual-string SCSI controllers, 28 5.25-inch SCSI disks and specialized disk stripping software
  - Today RAID is > $27 billion dollar industry, 80% nonPC disks sold in RAIDs

“RAID 0”: No redundancy = “AID”

- Assume have 4 disks of data for this example, organized in blocks
- Large accesses faster since transfer from several disks at once

This and next 5 slides from RAID.edu, http://www.acnc.com/04_01_00.html

PDF created with pdfFactory Pro trial version www.pdffactory.com
RAID 1: Mirror data
- Each disk is fully duplicated onto its “mirror”
- Very high availability can be achieved
- Bandwidth reduced on write:
  - 1 Logical write = 2 physical writes
- Most expensive solution: 100% capacity overhead

RAID 3: Parity
- Parity computed across group to protect against hard disk failures, stored in P disk
- Logically, a single high capacity, high transfer rate disk
- 25% capacity cost for parity in this example vs. 100% for RAID 1 (5 disks vs. 8 disks)

RAID 4: parity plus small sized accesses
- RAID 3 relies on parity disk to discover errors on Read
- But every sector has an error detection field
- Rely on error detection field to catch errors on read, not on the parity disk
- Allows small independent reads to different disks simultaneously

Inspiration for RAID 5
- Small writes (write to one disk):
  - Option 1: read other data disks, create new sum and write to Parity Disk (access all disks)
  - Option 2: since P has old sum, compare old data to new data, add the difference to P:
    1 logical write = 2 physical reads + 2 physical writes to 2 disks
- Parity Disk is bottleneck for Small writes:
  Write to A0, B1 => both write to P disk

RAID 5: Rotated Parity, faster small writes
- Independent writes possible because of interleaved parity
- Example: write to A0, B1 uses disks 0, 1, 4, 5, so can proceed in parallel
- Still 1 small write = 4 physical disk accesses

“And in conclusion...”
- Magnetic Disks continue rapid advance:
  60%/yr capacity, 40%/yr bandwidth, slow on seek, rotation improvements, MB/$ improving 100%/yr?
- Designs to fit high volume form factor
- RAID
  - Higher performance with more disk arms per $  
  - Adds option for small # of extra disks
  - Today RAID is > $27 billion dollar industry, 80% nonPC disks sold in RAIDs; started at Cal