Building a More Efficient Data Center – from Servers to Software

Aman Kansal
Data centers growing in number,

Microsoft has more than 10 and less than 100 DCs worldwide

"Data Centers have become as vital to the functioning of society as power stations."

The Economist
... size,
Each data center is 7 to 11.5 times the size of a football field.
..., and efficiency!

More apps have online components
  – Music, office s/w, ...

Lower cost DC ⇒ new scenarios
  – Improved speech recognition
  – Video on wireless HD/retina display tablets
    • Better encoding needs more compute: HEVC reduces bitrate by 50%

Please speak or enter your flight number...I’m sorry I did not get that, please speak or enter your flight number...
Inside a Data Center

Power Distribution

IT Equipment (servers, network)

Cooling

Power Usage Effectiveness (PUE)

\[
PUE = \frac{\text{Total Facility Power Usage}}{\text{IT Equipment Power Usage}}
\]

How did Microsoft improve PUE from near 2.0 to 1.05 in five years?
Measuring Data Centers

Collect, archive, and understand operations data

Operation monitoring, Capacity planning, Device provisioning, Resource control

Source: Jie Liu
Older Cooling Design

Hot air is not contained
New and Improved

Containment: tightly guide air-flow
Use outside air: locate in cooler region
Operate servers hotter

1989-2005
Cold + hot aisles
PUE = 1.5 - 2

2008
Containers
PUE = 1.2-1.5

2011
Custom Module
PUE = 1.05- 1.15
Inside a Module

- Reduce building cost
- Enable modular growth
- Pre-fab’d, go live faster
PUE ≈ 1, are we done?

For given IT load, not wasting excess energy, but we can reduce
- Power required for same app
- Infrastructure
Beyond PUE

- Servers+Networking: 65%
- Power and Cooling Infrastructure: 18%
- Energy Usage: 13%
- Other Infrastructure: 4%

Data from: James Hamilton
Renewable energy
Shut down servers
Move load to where energy cheaper
Reduce peak power
Reduce infra (dual utility instead of generator)

Optimize IT
Design apps to use servers efficiently.

Right size servers for apps hosted.

Cheaper servers

Fewer servers
Cheaper Servers
Obvious

No one installs s/w from a CD on 1000s of servers: *remove the optical drive*

*Use blades*: share fans, power supplies
### High Cost Components

<table>
<thead>
<tr>
<th>Component</th>
<th>Cost Range</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU</td>
<td>$300-1500/socket</td>
<td>Eg. Intel Xeon E5</td>
</tr>
<tr>
<td>Memory</td>
<td>$20-30/GB</td>
<td>64GB = $1280+</td>
</tr>
<tr>
<td>Hard disk</td>
<td>$100-300/TB</td>
<td>SATA vs. SAS, 3 - 6Gbps, 7.2 – 15 RPM</td>
</tr>
<tr>
<td>SSD</td>
<td>$1000-5000/TB</td>
<td>Vary by brand/perf.</td>
</tr>
</tbody>
</table>

Prices in January 2013
Right-size server to app needs

Bing
- Web crawling, index management, query lookup
- Major load: Index lookup
- Highly latency critical

Hotmail
- UI, mail protocols, spam filtering, storage
- Major load: Retrieve data from mailboxes
- Stores several petabytes of data, IOPS intensive

Cosmos
- Highly parallelized data storage and analysis
- Major load: distributed storage and batched compute
- Throughput intensive
# App Resource Usage

<table>
<thead>
<tr>
<th>App</th>
<th>Memory Capacity</th>
<th>Memory BW</th>
<th>Disk Capacity</th>
<th>Disk BW</th>
<th>Network BW</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hotmail</td>
<td>92%</td>
<td>NA</td>
<td>75%</td>
<td>0.91%</td>
<td>27%</td>
</tr>
<tr>
<td>Cosmos</td>
<td>39%</td>
<td>1.1%</td>
<td>52%</td>
<td>0.68%</td>
<td>9%</td>
</tr>
<tr>
<td>Bing</td>
<td>88%</td>
<td>1.8%</td>
<td>30%</td>
<td>1.10%</td>
<td>10%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>App</th>
<th>CPU Utilization</th>
<th>Memory Bandwidth</th>
<th>Disk Bandwidth</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hotmail</td>
<td>67%</td>
<td>NA</td>
<td>71%</td>
</tr>
<tr>
<td>Cosmos</td>
<td>88%</td>
<td>1.6%</td>
<td>8%</td>
</tr>
<tr>
<td>Bing</td>
<td>97%</td>
<td>5.8%</td>
<td>36%</td>
</tr>
</tbody>
</table>
CPU: Frequency

<table>
<thead>
<tr>
<th>Relative Performance</th>
<th>Search 1.87GHz</th>
<th>Search 2.13GHz</th>
<th>Search 2.40GHz</th>
<th>Cosmos 2GHz</th>
<th>Cosmos 2.30GHz</th>
<th>Cosmos 2.66GHz</th>
<th>Cosmos 3.00GHz</th>
<th>Cosmos 3.13GHz</th>
</tr>
</thead>
</table>
CPU: Number of Cores

<table>
<thead>
<tr>
<th>Relative Performance</th>
<th>4 CORES</th>
<th>8 CORES</th>
<th>4 CORES</th>
<th>8 CORES</th>
</tr>
</thead>
<tbody>
<tr>
<td>SEARCH</td>
<td>1</td>
<td>1.95</td>
<td>1</td>
<td>1.8</td>
</tr>
<tr>
<td>COSMOS</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Cost

But power and cost also increase with frequency and number of cores

Figure of merit:

\[
\frac{\text{Performance}}{\text{Power (W)} \times \text{Cost}($)}
\]
Assumption: Server Price = $2000 + CPUs, Server Power = 150W + CPUs
## App Resource Usage: Disk

<table>
<thead>
<tr>
<th>App</th>
<th>Memory Capacity</th>
<th>Memory BW</th>
<th>Disk Capacity</th>
<th>Disk BW</th>
<th>Network BW</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hotmail</td>
<td>92%</td>
<td>NA</td>
<td>75%</td>
<td>0.91%</td>
<td>27%</td>
</tr>
<tr>
<td>Cosmos</td>
<td>39%</td>
<td>1.1%</td>
<td>52%</td>
<td>0.68%</td>
<td>9%</td>
</tr>
<tr>
<td>Bing</td>
<td>88%</td>
<td>1.8%</td>
<td>30%</td>
<td>1.10%</td>
<td>10%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>App</th>
<th>CPU Utilization</th>
<th>Memory Bandwidth</th>
<th>Disk Bandwidth</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hotmail</td>
<td>67%</td>
<td>NA</td>
<td>71%</td>
</tr>
<tr>
<td>Cosmos</td>
<td>88%</td>
<td>1.6%</td>
<td>8%</td>
</tr>
<tr>
<td>Bing</td>
<td>97%</td>
<td>5.8%</td>
<td>36%</td>
</tr>
</tbody>
</table>
Disk

Bandwidth optimizations

- Hotmail: Mix hot and cold data to spread bandwidth
- Striping/mirroring instead of RAID

Latency

- Use memory to cache data

Flash storage

- Expensive per byte stored but cheaper in bandwidth
  - Bandwidth is not a bottleneck for above apps
- Flash may potentially enhance memory
Memory

Low latency for interactive apps demands high memory capacity

- Bing is memory bound
- Hotmail: SQL index uses available memory for caching
- Cosmos: disk bound, smaller memory sufficient
- Rising popularity of memcached

Halving the processor cache did not degrade performance for Bing and Cosmos

- Cache does not significantly reduce memory access
## Scale Up or Scale Out

Are two cheaper servers better than one higher capability server?

<table>
<thead>
<tr>
<th></th>
<th>UP (1S)</th>
<th>DP (2S)</th>
<th>MP (4S)</th>
<th>MP (8S)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPUs</td>
<td>1</td>
<td>2</td>
<td>4</td>
<td>8</td>
</tr>
<tr>
<td>Cores per CPU</td>
<td>4</td>
<td>8</td>
<td>24</td>
<td>48</td>
</tr>
<tr>
<td>Memory</td>
<td>8</td>
<td>16</td>
<td>48</td>
<td>96</td>
</tr>
<tr>
<td>Drives</td>
<td>2</td>
<td>3</td>
<td>8</td>
<td>16</td>
</tr>
</tbody>
</table>
Fewer Servers
Over-provisioning Dilemma

Load varies with time

Provision more or less?

Messenger load with time
Over-provisioning (contd.)

Large difference between peak and typical

<table>
<thead>
<tr>
<th>MIPS/Core</th>
<th>Disk MBps/Core</th>
<th>MIPS/Disk MBps</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Avg+2Sigma Max</td>
<td>Avg+2Sigma Max</td>
</tr>
<tr>
<td>Amdahl</td>
<td>1</td>
<td>8</td>
</tr>
<tr>
<td>Hotmail</td>
<td>1059 0.32 25.22</td>
<td>3271 42</td>
</tr>
<tr>
<td>Cosmos</td>
<td>3698 0.24 2.73</td>
<td>15173 1357</td>
</tr>
<tr>
<td>Bing</td>
<td>1849 0.17 5.73</td>
<td>10643 323</td>
</tr>
</tbody>
</table>
Growth Granularity

- Built capacity
- Actual demand

Growth vs. Time graph
Consolidate in a Shared Cloud

Pack hundreds, thousands of apps on shared infrastructure: keep utilization high

CPU + memory

storage
Consolidation Can Hurt Performance

- **Low utilization**
  - High idle energy

- **High utilization**
  - Low throughput

**Sweet Spot**
Measurement

Power and performance for a toy web service with CPU and disk access
Virtualize to Isolate Resources

Not enough

Up to 125% degradation in Intel Core 2 Duo, Nehalem, AMD Opteron

Up to 40% measured on Google data center apps [Tang et al, ISCA’11]
CPU: Isolation is not perfect

Shared Cache

VM1

VM2

CPU1

CPU2

Memory Bandwidth

DRAM

Shared resource contention

Isolated

Isolated
Interference Can Be Modeled

Individual modeling to predict all co-located sets

Modeling Tool

Tunable Cache Intensity

Sets

Ways

Application Perf. Degr. (%)

Measured

Predicted

[Govindan et al, SOCC 2011]
CPU: Performance Aware Consolidation

- Customer VMs
- Cloud Deployment API
- Staging Servers
- Consolidation Algorithm
- Hosting Racks
Consolidating Storage

Allocate required storage capacity

But performance depends on I/O bandwidth
Bandwidth is Not Additive
Sufficient Bandwidth

\[ B_{\text{max}}(A_i) = \text{maximum bandwidth that app A can use within performance bound} \]

\[ B(A_i) = \text{current bandwidth usage of app A} \]

\[ \sum_{i=0}^{n} B(A_i) < \min_{i=1 \ldots n} B_{\text{max}}(A_i) \]
Bandwidth Varies Over Time

More users active at certain times => more photos, emails
### Storage Consolidation Savings

<table>
<thead>
<tr>
<th>Strategy</th>
<th>Energy Savings</th>
<th>Performance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Capacity only</td>
<td>2.31</td>
<td>0.623</td>
</tr>
<tr>
<td>Bandwidth</td>
<td>1.35</td>
<td>0.970</td>
</tr>
<tr>
<td>Capacity, Bandwidth, Dynamics</td>
<td>3.18</td>
<td>0.982</td>
</tr>
</tbody>
</table>

Average savings across 10 Microsoft data center applications, relative to when hosted without consolidation (in research).
Summary: Don’t forget the biggest slice

1. Look beyond energy use: infrastructure, IT
2. Use cheaper servers: tune for app needs
   - CPU: fastest is not most efficient
   - Storage: capacity is cheap, optimize for fast access (cache in RAM, stripe)
   - Memory: larger RAM benefits interactive apps
3. Use fewer servers: do not waste idle capacity
   - Consolidate: do more with less
   - Bin packing is not enough, preserve performance
Acknowledgments

Sriram Govindan
Jie Liu
Suman Nath
Sriram Shankar
Kushagra Vaid
Feng Zhao

Christina Delimitrou
Christos Kozyrakis
Harold Lim
Alan Roytman
Shekhar Srikantaiah
Questions?

http://www.facebook.com/EfficientDataCenter