Let’s go over the outline

What was particularly confusing?

What was clear?

What did you like?

What did you dislike?

Let’s talk about Figure 1
Figure 1: Actual running times for sorting 64-bit keys on a 32K Connection Machine CM-2. In the figure, the running times are divided by the number of keys per processor to permit extrapolation to machines with different numbers of processors. The term processor, as used in this paper, is a 32-bit wide so-called “Sprint” node, of which there are \( p = 1024 \) in a 32K CM-2. To determine the total running time of a sort involving \( n \) keys, multiply the time per key per processor in the figure by \( n/p \).
**Terms**

**Flynn’s Taxonomy** – in 1966 Michael Flynn suggested that computer architectures could be classified based on the number of instruction streams and data streams that they use.

<table>
<thead>
<tr>
<th>Single Instruction</th>
<th>Multiple Instructions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single Data</td>
<td>SISD</td>
</tr>
<tr>
<td>Multiple Data</td>
<td>SIMD</td>
</tr>
</tbody>
</table>
Terms

- **SIMD** – instructions executed in lock-step (e.g. a vector processor)
- **MIMD** – most parallel computers today are MIMDs (e.g. all the computers on the Top500)
- **Single Program Multiple Data (SPMD)** – all processors running same code but not necessarily in lock-step (e.g. the MPI programs we have been writing)
- **Multiple Program Multiple Data (MDMD)** – different processes running different code (e.g. master/worker)
Terms

- Interconnection network – network connecting the processors of a distributed memory system
  - Hypercube
  - Mesh (or torus)

- Sorting network – a network of comparators and wires that sorts a sequence of numbers (this provides a good model for parallel sorting)
Terms

Sorting Algorithms

- Flashsort – for partially ordered sequences – estimates location and then fixes errors
- Radix sort – based on digits
- Quicksort – pivoting, partitioning, recursing
- Sample sort – randomized sort, uses buckets
- Bitonic sort – merges bitonic sequences
## Architecture

<table>
<thead>
<tr>
<th>CM-2</th>
<th>NSCC</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>Each “processor” (sprint node) is a collection of 32 or 64 1-bit ALUs</em></td>
<td>64-bit processor</td>
</tr>
<tr>
<td>Local memory is between 256 KB and 4MB</td>
<td>“Local” memory is big – each pair of processors share a 6MB level-2 cache</td>
</tr>
<tr>
<td>&gt;1000 sprint nodes</td>
<td>8 processors</td>
</tr>
</tbody>
</table>

*A 64k CM-2 is comparable to a 2K 32-bit processor machine*
**Architecture**

<table>
<thead>
<tr>
<th>CM-2</th>
<th>NSCC</th>
</tr>
</thead>
<tbody>
<tr>
<td>&gt;1000 sprint node</td>
<td>8 processors</td>
</tr>
<tr>
<td>Interconnection network is a 11D Hypercube</td>
<td>Bus-based shared memory</td>
</tr>
<tr>
<td>Multiport capability</td>
<td>Does not apply because they all share a bus.</td>
</tr>
</tbody>
</table>

*A 64k CM-2 is comparable to a 2K 32-bit processor machine.*
Figure 2: The organization of a CM-2 Sprint node.

(Bellocchio et al, 1991)
A 64k CM-2 is comparable to a 2K 32-bit processor machine.

### CM-2
- Paris (CM-2’s assembly language) and high-level microcode
- Send (Point-to-Point communication)
- Scan
- Cube-swap

### NSCC
- C with MPI
- MPI_Send
- MPI_Scan
- MPI allows for virtual topologies but there is no special cube-swap