Bi-directional ORBs —
Two approaches

Peter Johansson
Congruent Software, Inc.

SBP-3 Working group
24 April, 2001
Normal command block ORB

```
<table>
<thead>
<tr>
<th>next_ORB</th>
</tr>
</thead>
<tbody>
<tr>
<td>data_descriptor</td>
</tr>
<tr>
<td>n</td>
</tr>
<tr>
<td>command</td>
</tr>
</tbody>
</table>
```
One buffer per ORB

- Either direct or indirect (page table)
- Required fields:
  
  \[
  \text{data\_descriptor, direction, spd, max\_payload, page\_table\_present, page\_size and data\_size}
  \]

- 92 bits total (round to three quadlets)
Indirect access to two buffers

- Typical ORB size remains 32 bytes
- Inefficient access through indirect descriptors
- ORB fields wasted
Direct access to two buffers

- ORB header increased by 12 bytes
- Efficient direct access via descriptors in ORB
- Compatible with FAST_START
FAST_START strategy

- Write *only* one page table to FAST_START
  - Included page table is for the first `data_descriptor` field in the ORB
- Software optimizes descriptor usage
  - If target needs data first, zero `direction` bit
  - Else target provides data first: set `direction` bit to one
- Target fetches other page table at its leisure
Operating system considerations

- Examine typical commands that would use bi-directional ORBs
  - RAID combined XOR and write
- Enhance API
  - ORB type selection (single or dual buffer)
  - Preserve ordered relationship of buffers from API to ORB
  - All buffer parameters are independent!
Contact information

Peter Johansson
Congruent Software, Inc.
98 Colorado Avenue
Berkeley, CA  94707

(510) 527-3926
(510) 527-3856 FAX

PJohansson@ACM.org