libtcr

Making the most of your cores
Philipp Reisner
Linuxkongress 2010, Nuremberg
We got cores
We will get more!
Non-uniform memory access
Cache coherency is expensive, ...
... will get more expensive!
Server programs,
Server programs,
Multi process - ... threaded - state machines
Server programs,
Multi process - ... threaded - state machines
complexity,
low - low - hard
Server programs,
Multi process - ... threaded - state machines
performance on UP,
low - better - best
Server programs,
Multi process - ... threaded – state machines
throughput scales with cores,
yes - yes - no
The winner is...
<table>
<thead>
<tr>
<th>complexity</th>
<th>multi proce</th>
<th>multi threaded</th>
<th>state machine</th>
</tr>
</thead>
<tbody>
<tr>
<td>uni processor</td>
<td>+</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>scales cores</td>
<td>+</td>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td>cache locality</td>
<td>-</td>
<td>-</td>
<td>+</td>
</tr>
<tr>
<td>complexity</td>
<td>+</td>
<td>+</td>
<td>-</td>
</tr>
</tbody>
</table>
A combination
program's state = what to do next
state machine = store state explicitly
threading = state implicitly on the stack
Switching stack = one assignment
libtrc is ...
...scheduling (stack switching) in user space on multiple kernel threads in parallel
N:M threading
Pipelining...
... with a thread per stage, is worst case for cache coherency
Process data on one CPU, read, process and write
Features, threads, mutexes, wait queues...
Features,
wait_fd, rearm_fd - 2 priorities
Features, tc_signal
Features, parallel for, parallel statement macro
Event sources,
FDs = FDs & timers, Unix signals, AIO
Internally, epoll
Internally, epoll and a pthread per core
Requires,
Linux 2.6.25
Area to improve, combination of cpu intense and IO multiplexing workloads
Drawbacks, Debugging aids
Need more information?

http://oss.linbit.com/libtcr
http://git.drbd.org/libtcr.git