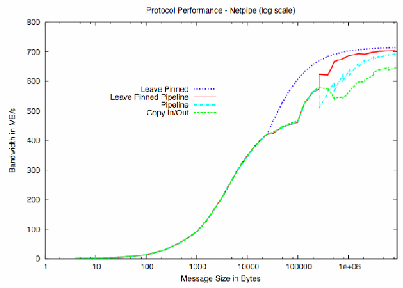
Memory Manager Fun

Brian Barrett

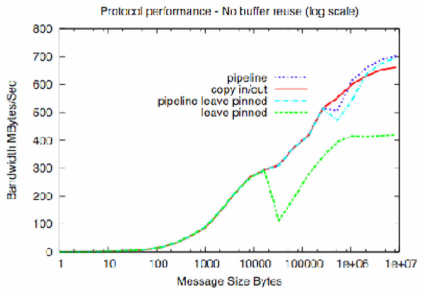
Latency/Bandwidth, Oh My!

- The Problem: NetPIPE as a benchmark
  - Gives Latency / Bandwidth with high buffer reuse
- Many NICs require pre-pinning for RDMA
  - Pinning expensive
  - Max performance requires “lazy unpinning”
- Lazy pinning leads to the dark side
  - Calling free() on pinned memory bad
  - MPI semantics don’t require special memory

Simple NetPIPE



But remove the reuse….

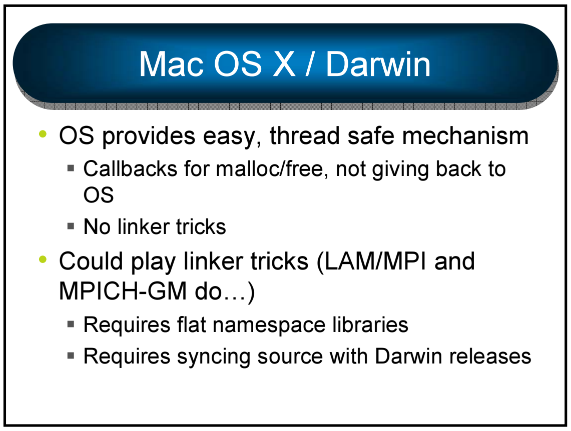


Our Strategy

- Allow lazy unpinning of memory
  - Linux and OS X only
- Red/black tree to store pinned page lists
- Intercept `malloc/free`
  - `malloc` allows optimized red/black storage
  - `free` intercepted to do unpinning
- Performance cost…
  - Searching for page…
  - N times (once per existing mpool)

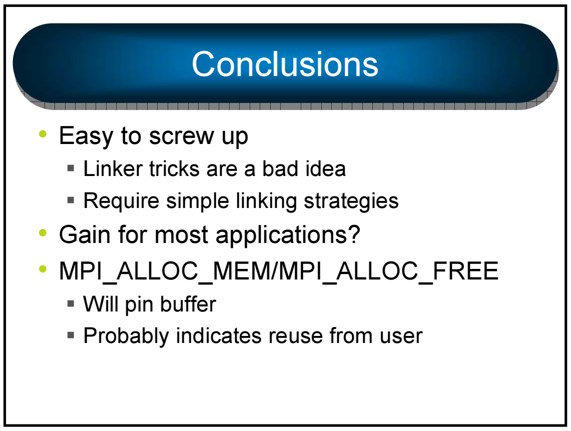
Linux

- Two models: intercept free or mallopt
- mallopt(M_TRIM_THRESHOLD, -1)
  - Can lead to degenerate malloc cases
- Intercept `free` (GAH!)
  - Linker tricks - provide our own copy of ptmalloc2
  - Linker tricks are a bad idea!
  - Only deregister when ptmalloc2 giving memory back to OS
- GLIBC malloc hooks not thread safe - not useable



Mac OS X / Darwin

- OS provides easy, thread safe mechanism
  - Callbacks for malloc/free, not giving back to OS
  - No linker tricks
- Could play linker tricks (LAM/MPI and MPICH-GM do…)
  - Requires flat namespace libraries
  - Requires syncing source with Darwin releases



Conclusions

- Easy to screw up
  - Linker tricks are a bad idea
  - Require simple linking strategies
- Gain for most applications?
- MPI_ALLOC_MEM/MPI_ALLOC_FREE
  - Will pin buffer
  - Probably indicates reuse from user