

GNU Radio Advanced Scheduler



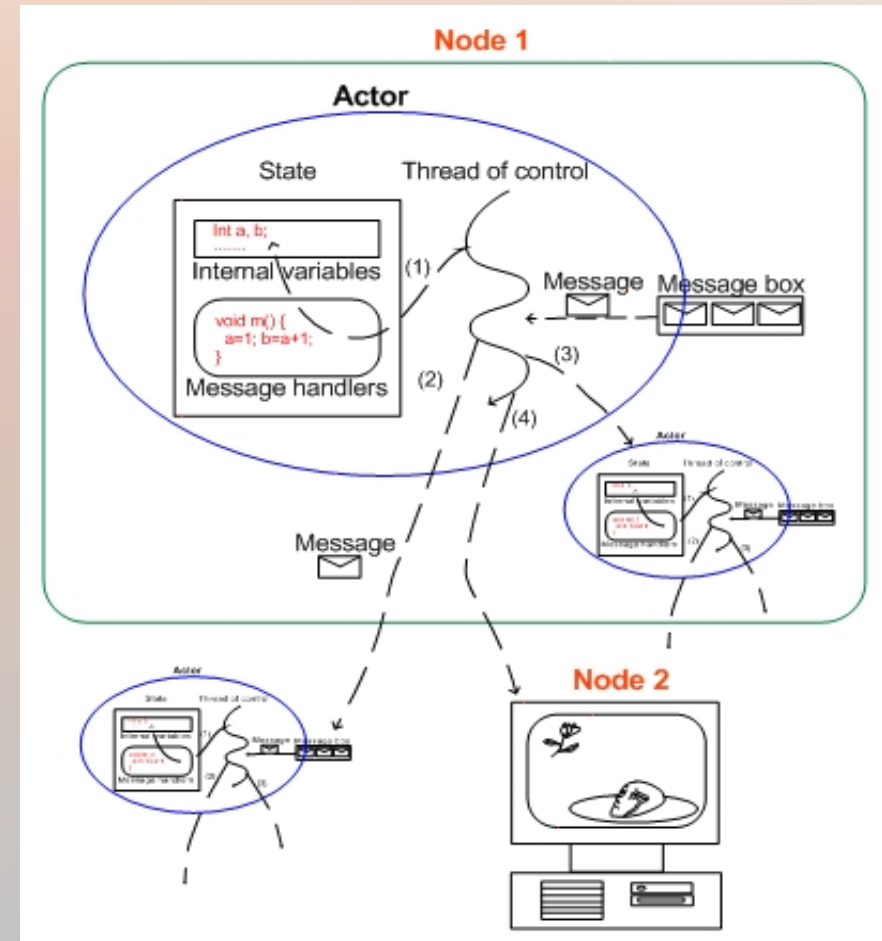
Dude: Josh Blum -
New scheduler features and stuff

GRAS - Project Goals

- Inherent thread safety – actor model
- New buffer model
- Zero copy support
 - In place buffers
 - Custom allocators
- Hooks for CPU/Node affinity
- Things that got fixed for free!

GRAS- Thread safety and the actor model

- Thread pool services actors
- Each actor is synchronous
- Messages in/replies to sender
- Messages to arbitrary actors



https://en.wikipedia.org/wiki/Actor_model

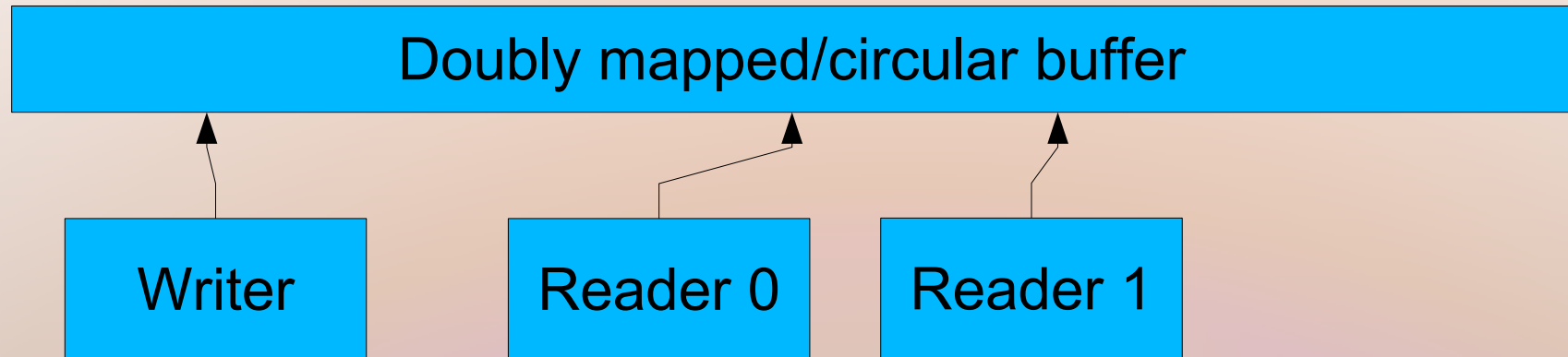
GRAS - Theron

C++ concurrency library

- C++ implementation of the actor model
- Fast, lots of benchmarks
- Hooks for affinity
- Multiple thread pools
- Result: I don't have to think about thread synchronization mechanisms

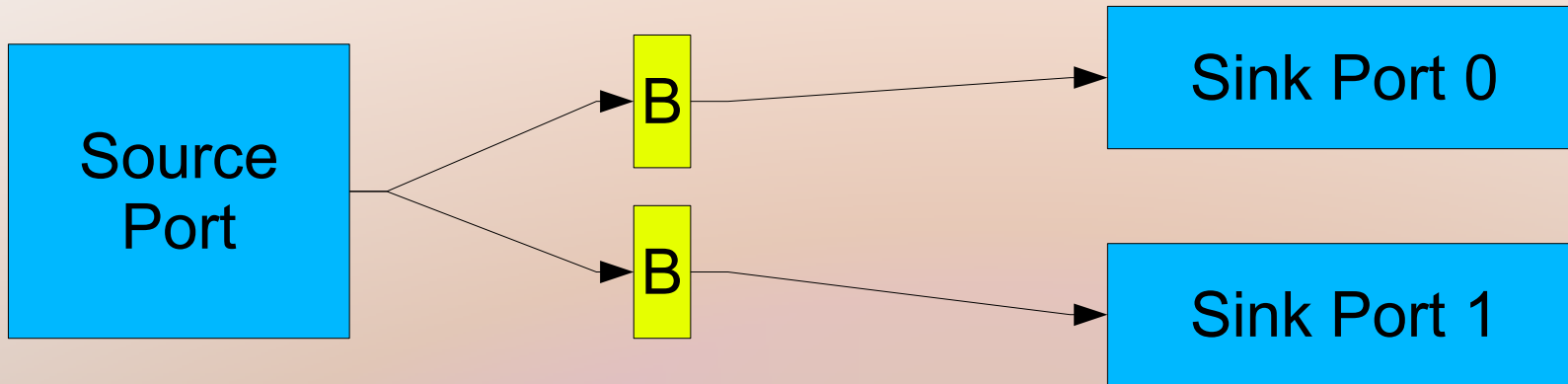
<http://www.theron-library.com/>

GRAS – Current buffer model



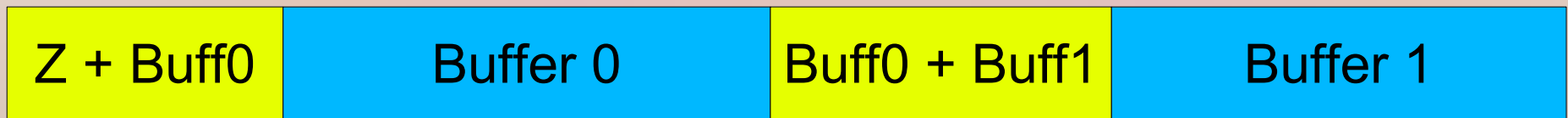
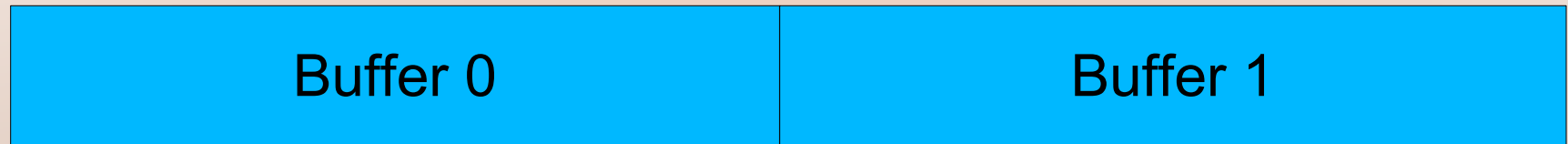
- Pros – very strait forward implementation
 - Simple back-pressure model
 - Edge cases are very simple
- Cons – cant do “smarter” things w/ memory
 - Buffer is glued to an output port
 - Cant in-place or choose arbitrary memory

GRAS – Message based buffer model



- New model means new possibilities
- Blocks pass reference counted buffers
- Pros – arbitrary buffer
 - Custom allocation – zero copy
 - Output buffer can be input - inplace
- Cons – complicates history and input reqs
 - Small memcpys

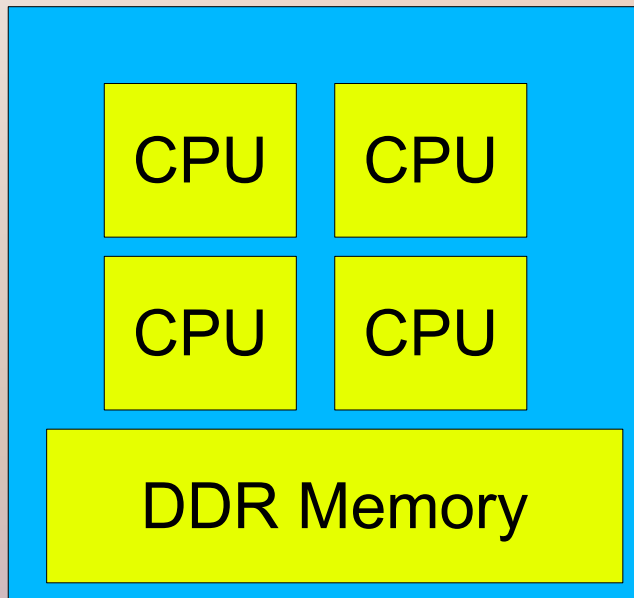
GRAS – Dealing w/ history



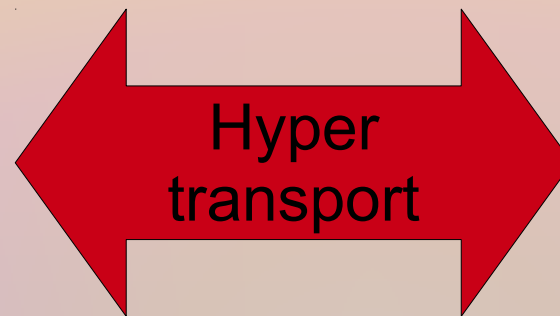
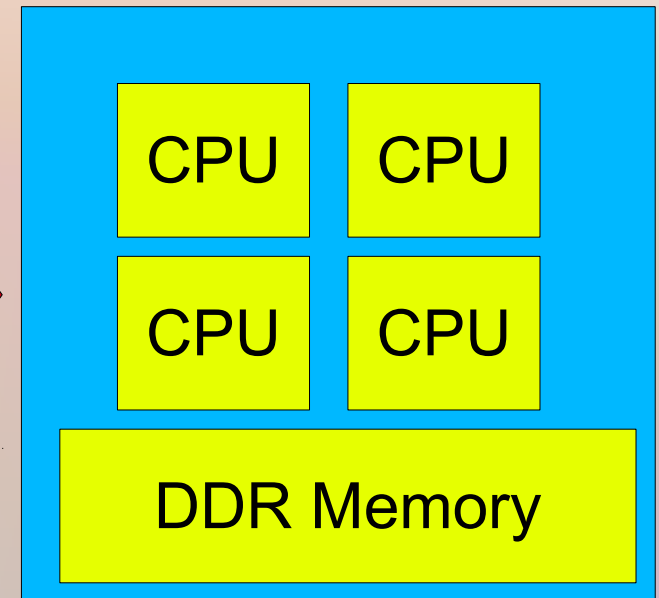
- Memcpy to meet history requirements – sux
- History reqs \ll buffer size (memcpys are small)
- Needs benchmarking

GRAS – Affinity Hooks

Node 0



Node 1



- Memory bandwidth penalties for jumping nodes
- Flow graph topology based node association
- Or multiple flow graphs, 1 per node... etc

GRAS – Things that got fixed

- Dynamic reconfiguration of flowgraph
 - Just change the topology and commit
- Hierarchical block fixes
 - connect to self to self - By bye kludge copy
 - Don't need `gnuradio::get_initial_sptr` for hiers!

GRAS – Whats Done

- GNU Radio basically working – QA tests
- In-place buffers
- Custom input and output allocators
- Dynamic reconfiguration

GRAS - TODO

- All QA tests passing
- GR hooks for setting thread pool
 - Creating thread pool w/ set affinities
- GR Extras features
 - PMT message passing – do native, many to one
 - Python blocks support
- Follow and support Theron development
- Benchmark, benchmark, benchmark!