

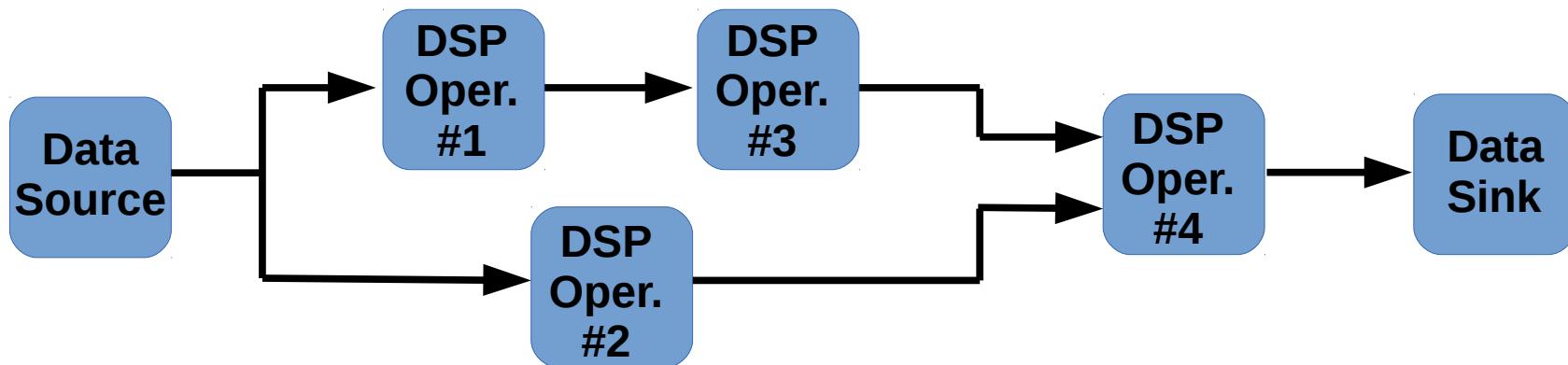
GNU Radio Runtime Operation

Johnathan Corgan
GRCON15



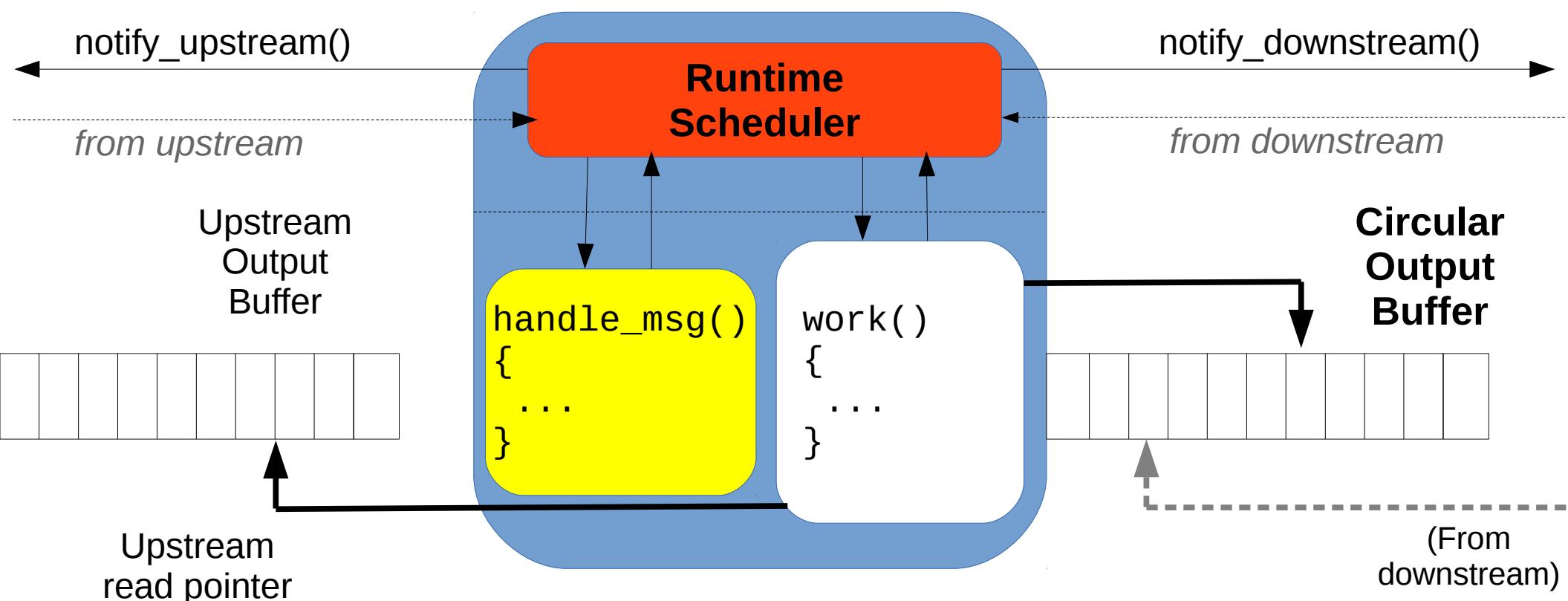
Data Flow Architecture

- The GNU Radio “runtime” is designed to stream large amounts of data in real-time
 - ...between parallel computational nodes
 - ...each operating completely independently.
- This is a very different model from traditional array processing environments.



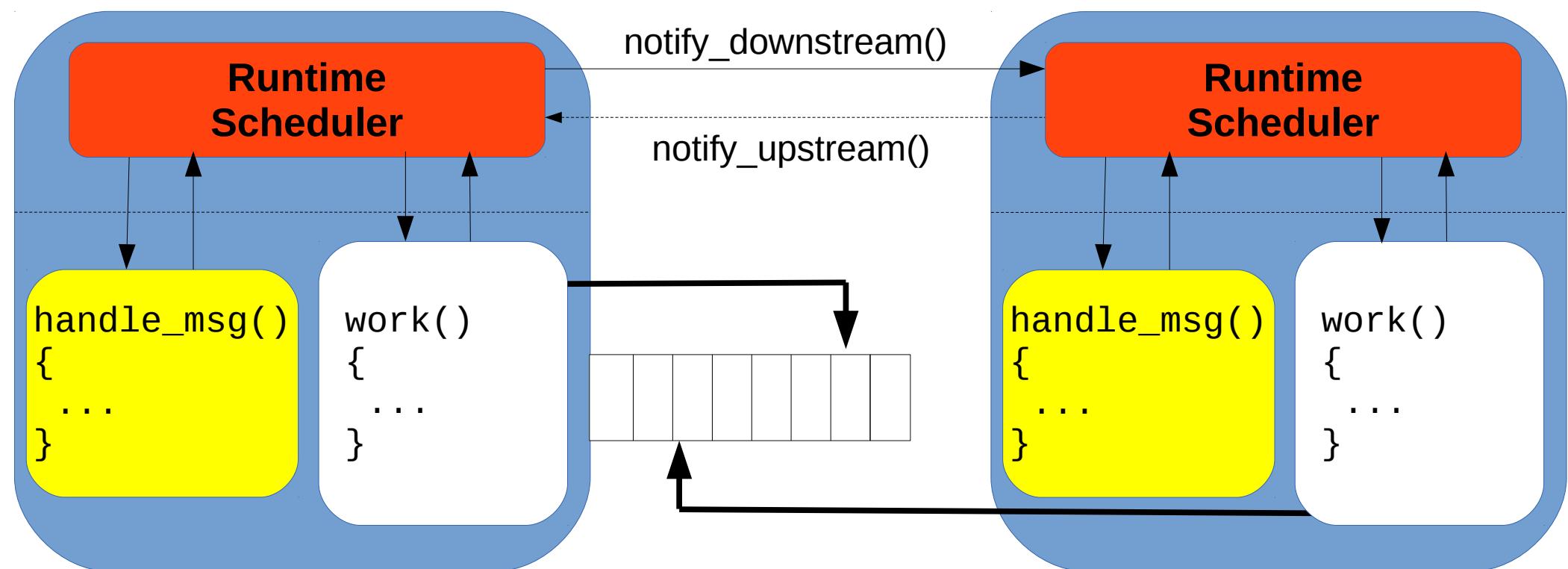
Anatomy of a GNU Radio Block

Each block has a completely independent scheduler running in its own execution thread.



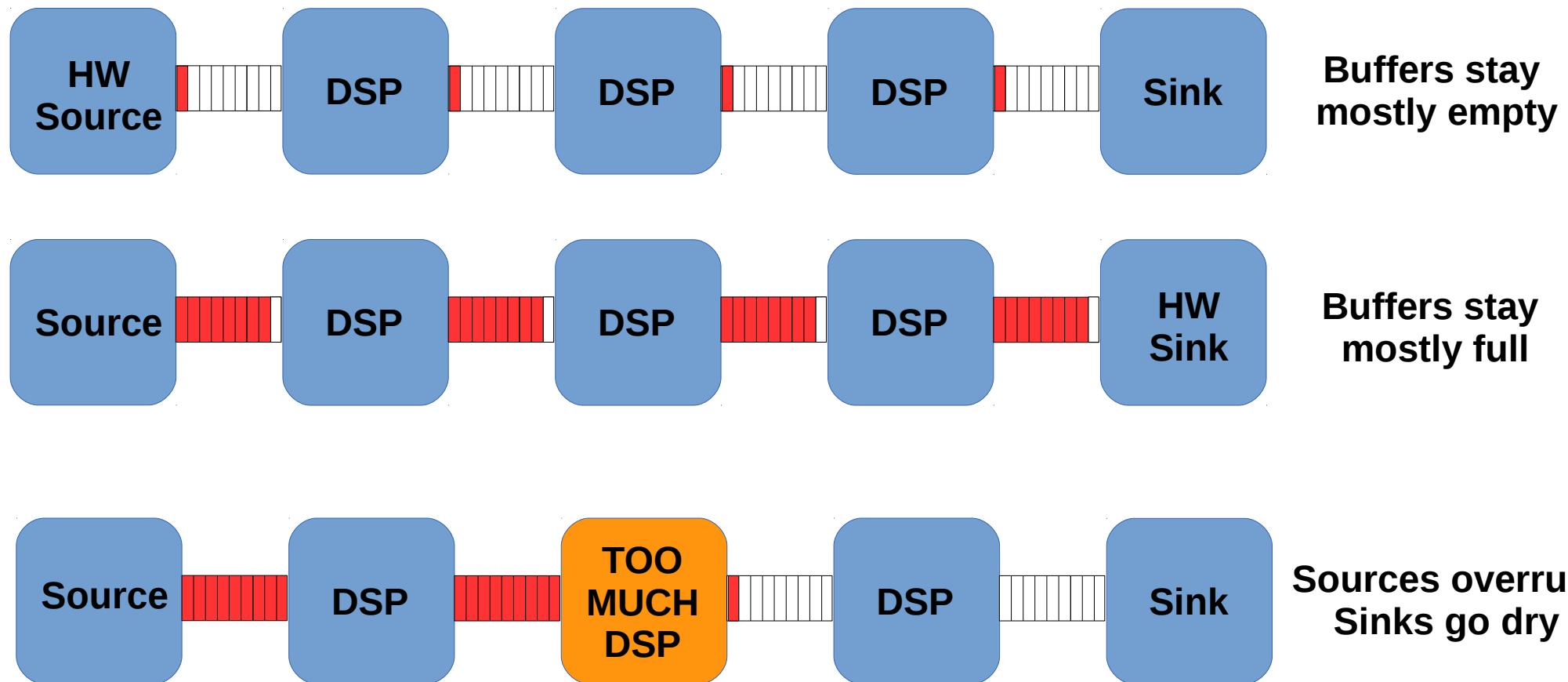
Inter-Block Flow Control

Blocks run as fast as the CPU, dataflow, and buffer space allows.



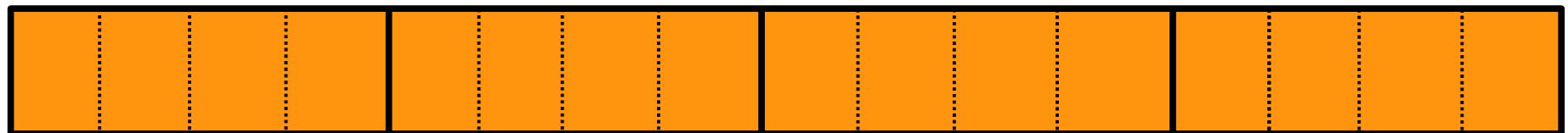
End-to-End Flow Control Considerations

Each flowgraph has a rate-setting block that should (almost) always be hardware.



Buffer Allocation and Latency Considerations

- GNU Radio creates buffers to accommodate several conflicting constraints
 - **Page size/item size boundary alignment**
 - Double sized to optimize read/write
 - Downstream decimation/interpolation
 - Block requested `set_output_multiple()`
 - Downstream block `history()` setting
- Default is 32K bytes and increased as needed



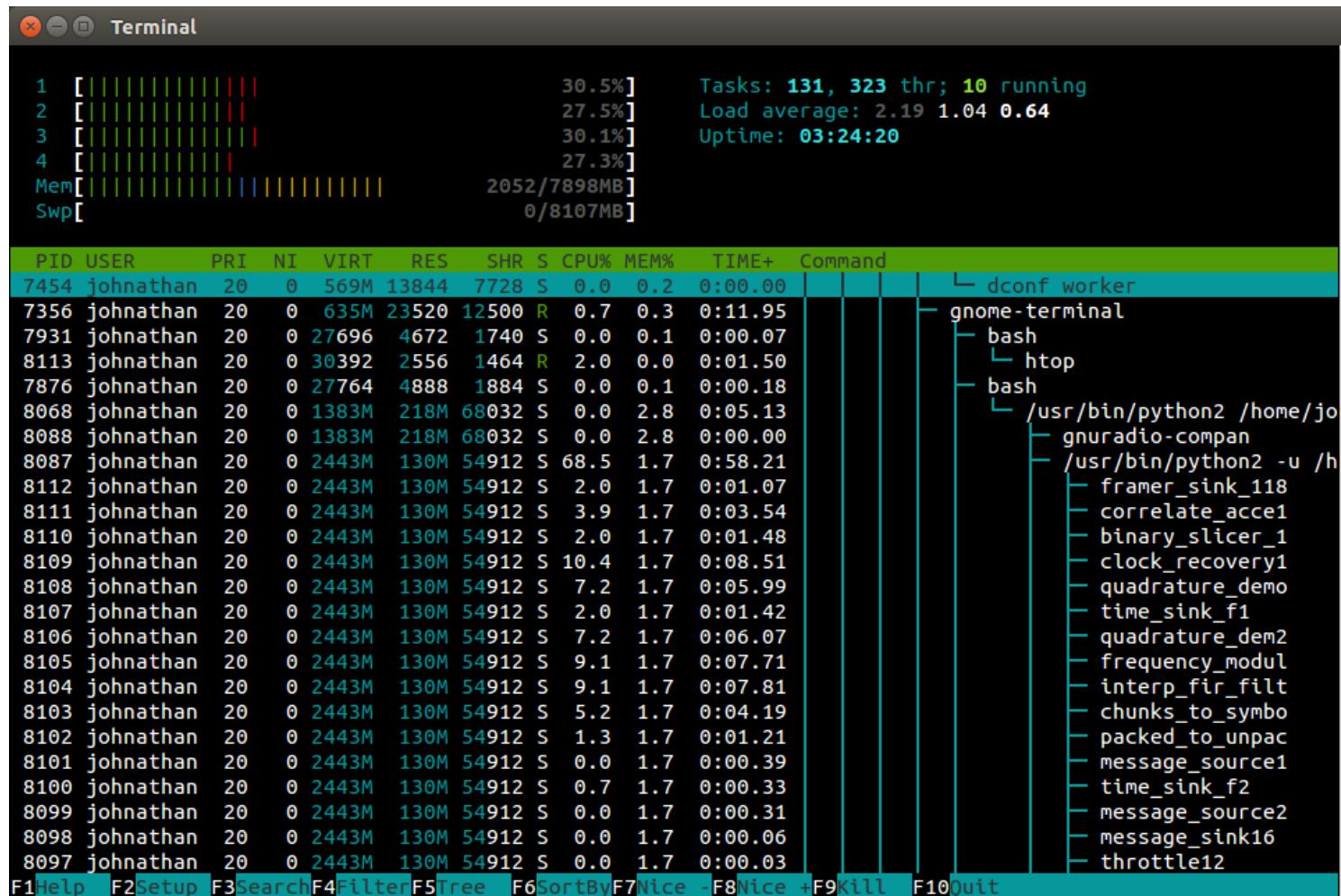
Danger, Will Robinson!

- It is possible to override the scheduler's buffer allocation algorithm for everything but page/item size:

```
set_min_output_buffer(items)  
set_max_output_buffer(items)
```
- This can cause unexpected behavior and performance tuning/testing is essential.
- If it breaks, you get to keep both pieces.

Performance Monitoring Tools

htop: thread and core aware process monitor



Performance Monitoring Tools

perf top: fine-grained process profiler

Terminal

```
Samples: 408K of event 'cycles', Event count (approx.): 37068776097
12.35% libgnuradio-runtime-3.7.8git.so.0.0.0  [...] gr::fast_atan2f(float, float)
12.30% libgnuradio-analog-3.7.8git.so.0.0.0  [...] gr::analog::frequency_modulator_fc_impl::work(int, std::vector<gr::tag_t, std::allocator<gr::tag_t>> const*)
12.09% libvolk.so.1.0                         [...] volk_32f_x2_dot_prod_32f_u_sse
  3.70% libgnuradio-analog-3.7.8git.so.0.0.0  [...] gr::analog::quadrature_demod_cf_impl::work(int, std::vector<gr::tag_t, std::allocator<gr::tag_t>> const*)
  2.95% libgnuradio-filter-3.7.8git.so.0.0.0   [...] gr::filter::kernel::fir_filter_fff::filter(float const*)
  2.50% libgnuradio-digital-3.7.8git.so.0.0.0  [...] gr::digital::clock_recovery_mm_ff_impl::general_work(int, int, std::vector<gr::tag_t, std::allocator<gr::tag_t>> const*)
  1.69% libgnuradio-digital-3.7.8git.so.0.0.0  [...] gr::digital::chunks_to_symbols_bf_impl::work(int, std::vector<gr::tag_t, std::allocator<gr::tag_t>> const*)
  1.51% i965_dri.so                           [...] 0x000000000001c2097
  1.37% libvolk.so.1.0                         [...] volk_32fc_x2_multiply_conjugate_32fc_u_sse3
  1.29% libgnuradio-blocks-3.7.8git.so.0.0.0  [...] gr::blocks::count_bits32(unsigned int)
  1.20% python2.7                            [...] PyEval_EvalFrameEx
  0.99% libpthread-2.19.so                   [...] pthread_mutex_lock
  0.91% python2.7                            [...] 0x0000000000015ac44
  0.89% perf                                 [...] 0x000000000000cb14a
  0.83% libc-2.19.so                          [...] __memcpy_sse2_unaligned
  0.82% libgnuradio-runtime-3.7.8git.so.0.0.0 [...] gr::block_executor::run_one_iteration()
  0.80% libgnuradio-filter-3.7.8git.so.0.0.0  [...] gr::filter::interp_fir_filter_fff_impl::work(int, std::vector<gr::tag_t, std::allocator<gr::tag_t>> const*)
  0.79% libpthread-2.19.so                   [...] pthread_mutex_unlock
  0.65% libgnuradio-filter-3.7.8git.so.0.0.0  [...] gr::filter::mmse_fir_interpolator_ff::interpolate(float)
  0.61% libgnuradio-digital-3.7.8git.so.0.0.0 [...] gr::digital::correlate_access_code_bb_impl::work(int, std::vector<gr::tag_t, std::allocator<gr::tag_t>> const*)
  0.58% _digital_swig.so                     [...] std::vector<gr::tag_t, std::allocator<gr::tag_t>>::~vector
  0.58% libvolk.so.1.0                        [...] volk_32fc_s32fc_multiply_32fc_u_sse3
  0.55% libgnuradio-blocks-3.7.8git.so.0.0.0 [...] gr::blocks::packed_to_unpacked_bb_impl::general_work(int, int, std::vector<gr::tag_t, std::allocator<gr::tag_t>> const*)
  0.55% libgnuradio-runtime-3.7.8git.so.0.0.0 [...] boost::detail::sp_counted_base::release()
  0.55% libv8.so                             [...] 0x00000000000388b4a
  0.54% libgnuradio-pmt-3.7.8git.so.0.0.0   [...] pmt::intrusive_ptr_add_ref(pmt::pmt_base*)
  0.52% [kernel]                            [...] __switch_to
  0.52% libQtGui.so.4.8.6                  [...] 0x000000000002e58e0
no symbols found in /usr/sbin/dnsmasq, maybe install a debug package?
```



Performance Monitoring Tools

- GNU Radio performance counters monitor a variety of block related values:
 - Output and upstream buffer fullness
 - Cycles spent in work()
 - Data throughput
- Can be exported via ControlPort to an external monitoring application
 - Requires (patched) Apache Thrift installation
 - Some issues with Thrift upstream bugginess

Performance Monitoring Tools

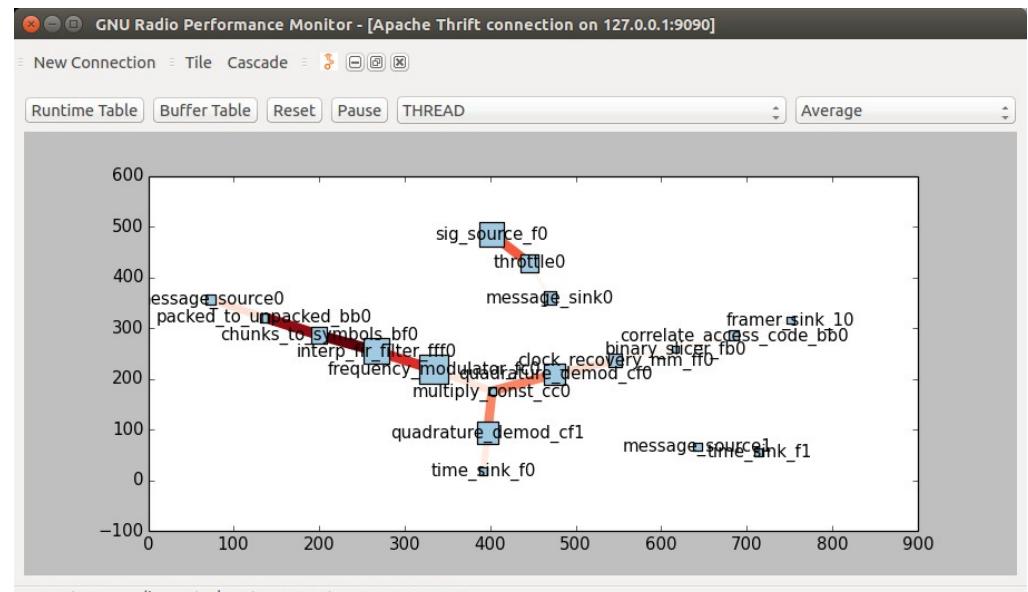
gr-ctrlport-monitor

GNU Radio Control Port Monitor - [Apache Thrift connection on 127.0.0.1:9090]

Statistic Key (Source Block :: Stat Name)	Current Value	Units	Description
quadrature_demod_cf1::avg noutput_items	3641.44238...		Average noutput items
quadrature_demod_cf1::avg input % full	[0.1171259...		Average of how full input buffers are
quadrature_demod_cf0::work time	188905.0		clock cycles in call to work
quadrature_demod_cf0::var work time	636704307...		Var. clock cycles in call to work
quadrature_demod_cf0::var output % full	[1.3962345...		Var. of how full output buffers are
quadrature_demod_cf0::var nproduced	4882062.0		Var. items produced
quadrature_demod_cf0::var noutput_items	4882062.0		Var. noutput items
quadrature_demod_cf0::var input % full	[1.3689736...		Var. of how full input buffers are
quadrature_demod_cf0::total work time	105934131...		Total clock cycles in calls to work
quadrature_demod_cf0::output % full	[0.2506103...		how full output buffers are
quadrature_demod_cf0::nproduced	4096.0		items produced
quadrature_demod_cf0::noutput_items	4096.0		noutput items
quadrature_demod_cf0::input % full	[0.4999389...		how full input buffers are
quadrature_demod_cf0::avg work time	108100.984...		Average clock cycles in call to work
quadrature_demod_cf0::avg output % f...	[0.1368253...		Average of how full output buffers are
quadrature_demod_cf0::avg nproduced	3449.94824...		Average items produced
quadrature_demod_cf0::avg noutput_items	3449.94824...		Average noutput items
quadrature_demod_cf0::avg input % full	[0.1341539...		Average of how full input buffers are
packed_to_unpacked_bb0::work time	14704.0		clock cycles in call to work
packed_to_unpacked_bb0::var work time	261297968.0		Var. clock cycles in call to work
packed_to_unpacked_bb0::var output ...	[1.1221640...		Var. of how full output buffers are

Current GNU Radio Control Port Query Latency: 91.031075 ms

gr-perf-monitorx



Complete, detailed statistics on every block

