Parallel Programming and Heterogeneous Computing

FPGA Accelerators - Metal FS Hands-On

Max Plauth, Sven Köhler, Felix Eberhardt, *Lukas Wenzel* and Andreas Polze

Operating Systems and Middleware Group

Hasso Plattner Institut
Digital Engineering · Universität Potsdam

Getting started with Metal FS

The Metal FS project lives on GitHub:

`https://github.com/osmhpi/metalfs`

Watch the intro video first (link on `https://osm.hpi.de/parProg`)!

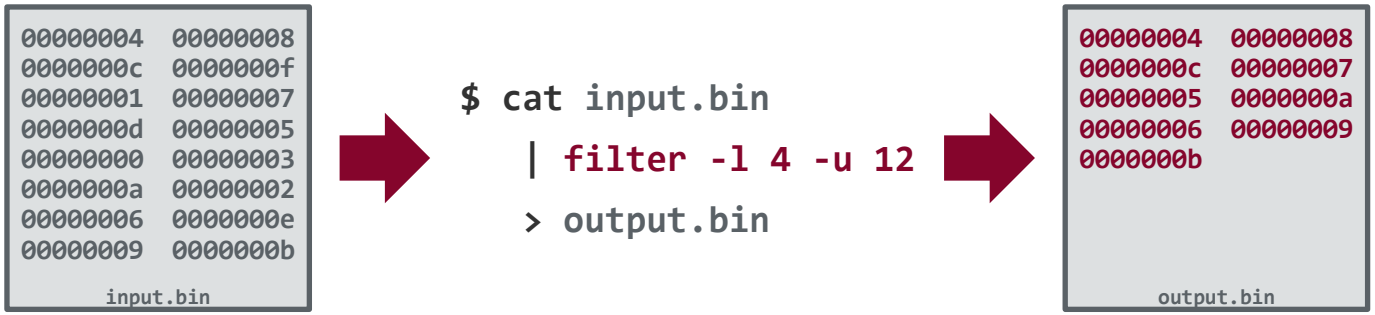
We are going to follow a modified version of this tutorial:

`https://metalfs.github.io/tutorial.html`

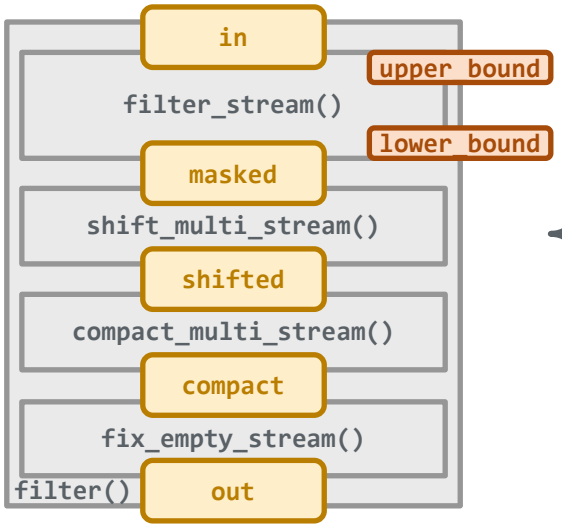
To follow along, you will need a platform with Docker and docker-compose.

Getting started with Metal FS

Build a Metal FS operator to filter unsigned 32-bit values.



Getting started with Metal FS

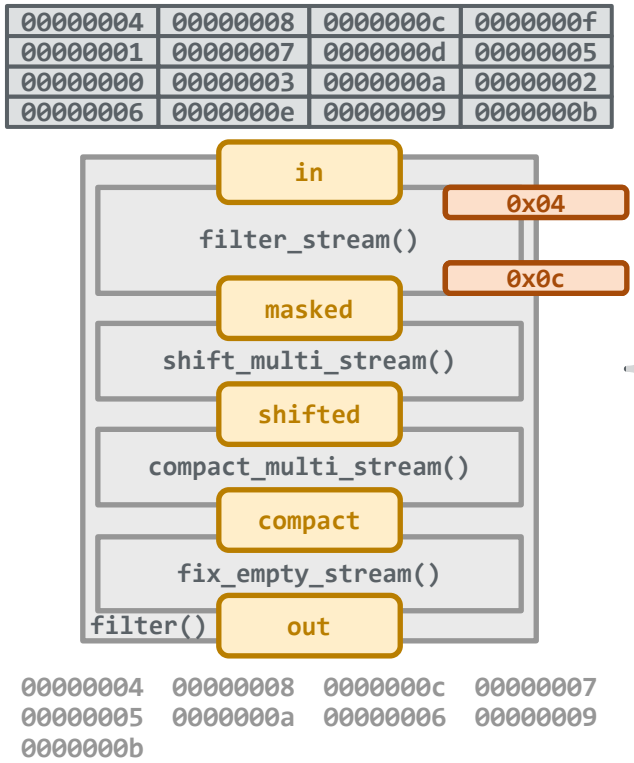



```
void filter(mtl_stream &in, mtl_stream &out,
            Value lower_bound, Value upper_bound) {
#pragma HLS INTERFACE axis port=in name=axis_input
#pragma HLS INTERFACE axis port=out name=axis_output
#pragma HLS INTERFACE s_axilite port=lower_bound bundle=control offset=0x100
#pragma HLS INTERFACE s_axilite port=upper_bound bundle=control offset=0x110
#pragma HLS INTERFACE s_axilite port=return bundle=control
  MaskedStream masked;
  ShiftedStream shifted;
  mtl_stream compact;

#pragma HLS DATAFLOW

  filter_stream(in, masked, lower_bound, upper_bound);

#if VALUE_COUNT > 1
  shift_multi_stream(masked, shifted);
  compact_multi_stream(shifted, compact);
#else
  compact_single_stream(masked, compact);
#endif

  fix_empty_stream(compact, out);
}
```

ParProg 2020
C3 FPGA Accelerators

Lukas Wenzel

Getting started with Metal FS

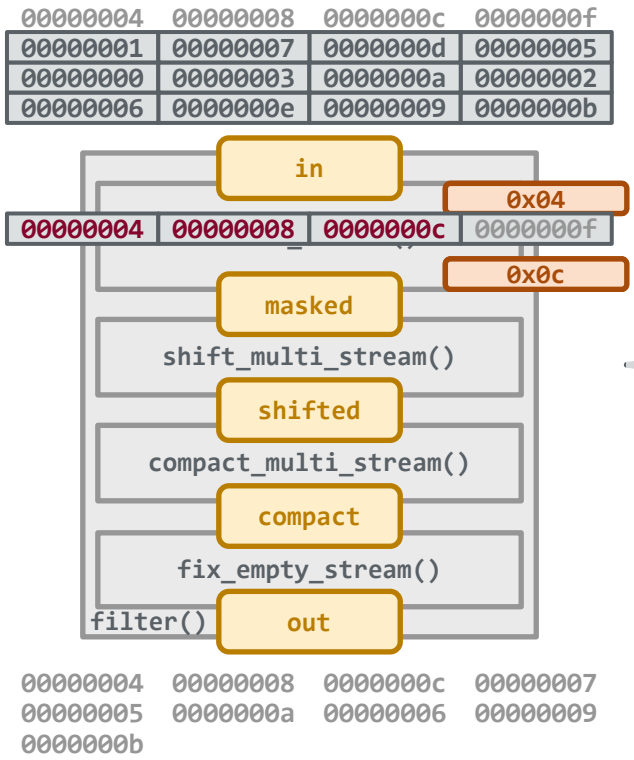



```
void filter(mtl_stream &in, mtl_stream &out,
            Value lower_bound, Value upper_bound) {
#pragma HLS INTERFACE axis port=in name=axis_input
#pragma HLS INTERFACE axis port=out name=axis_output
#pragma HLS INTERFACE s_axilite port=lower_bound bundle=control offset=0x100
#pragma HLS INTERFACE s_axilite port=upper_bound bundle=control offset=0x110
#pragma HLS INTERFACE s_axilite port=return bundle=control
  MaskedStream masked;
  ShiftedStream shifted;
  mtl_stream compact;

#pragma HLS DATAFLOW

  filter_stream(in, masked, lower_bound, upper_bound);

#if VALUE_COUNT > 1
  shift_multi_stream(masked, shifted);
  compact_multi_stream(shifted, compact);
#else
  compact_single_stream(masked, compact);
#endif

  fix_empty_stream(compact, out);
}
```

ParProg 2020
C3 FPGA Accelerators

Lukas Wenzel

Getting started with Metal FS

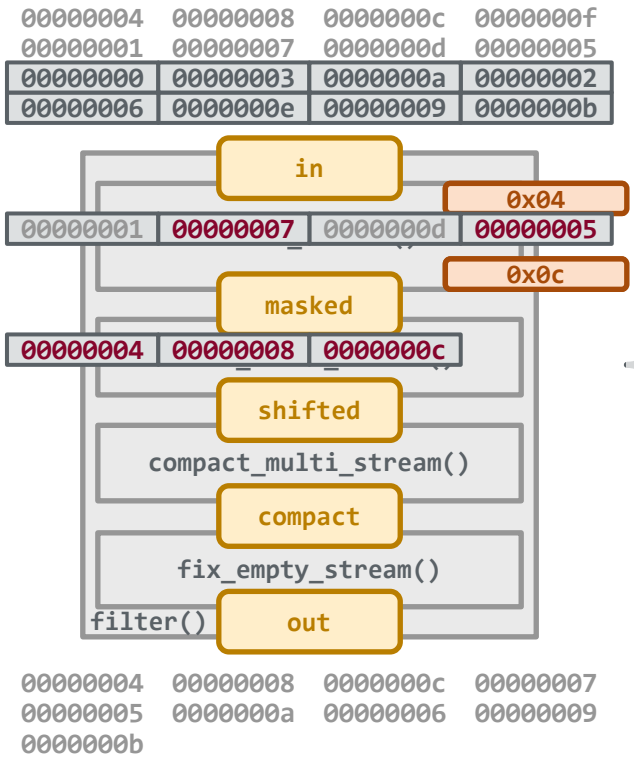



```
void filter(mtl_stream &in, mtl_stream &out,
            Value lower_bound, Value upper_bound) {
#pragma HLS INTERFACE axis port=in name=axis_input
#pragma HLS INTERFACE axis port=out name=axis_output
#pragma HLS INTERFACE s_axilite port=lower_bound bundle=control offset=0x100
#pragma HLS INTERFACE s_axilite port=upper_bound bundle=control offset=0x110
#pragma HLS INTERFACE s_axilite port=return bundle=control
  MaskedStream masked;
  ShiftedStream shifted;
  mtl_stream compact;

#pragma HLS DATAFLOW

  filter_stream(in, masked, lower_bound, upper_bound);

#if VALUE_COUNT > 1
  shift_multi_stream(masked, shifted);
  compact_multi_stream(shifted, compact);
#else
  compact_single_stream(masked, compact);
#endif

  fix_empty_stream(compact, out);
}
```

ParProg 2020
C3 FPGA Accelerators

Lukas Wenzel

Getting started with Metal FS



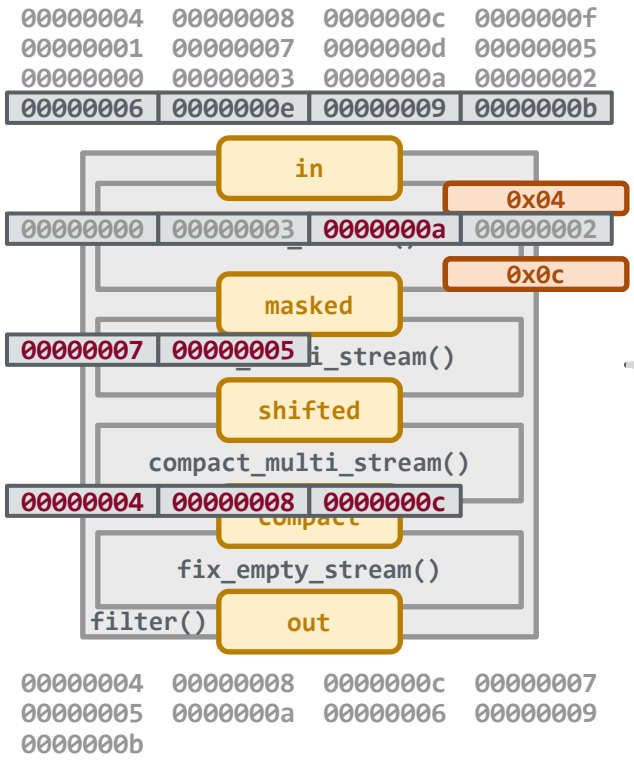

```
void filter(mtl_stream &in, mtl_stream &out,
            Value lower_bound, Value upper_bound) {
#pragma HLS INTERFACE axis port=in name=axis_input
#pragma HLS INTERFACE axis port=out name=axis_output
#pragma HLS INTERFACE s_axilite port=lower_bound bundle=control offset=0x100
#pragma HLS INTERFACE s_axilite port=upper_bound bundle=control offset=0x110
#pragma HLS INTERFACE s_axilite port=return bundle=control
  MaskedStream masked;
  ShiftedStream shifted;
  mtl_stream compact;

#pragma HLS DATAFLOW

  filter_stream(in, masked, lower_bound, upper_bound);

#if VALUE_COUNT > 1
  shift_multi_stream(masked, shifted);
  compact_multi_stream(shifted, compact);
#else
  compact_single_stream(masked, compact);
#endif

  fix_empty_stream(compact, out);
}
```

ParProg 2020
C3 FPGA Accelerators

Lukas Wenzel

Chart **4.4**

Getting started with Metal FS

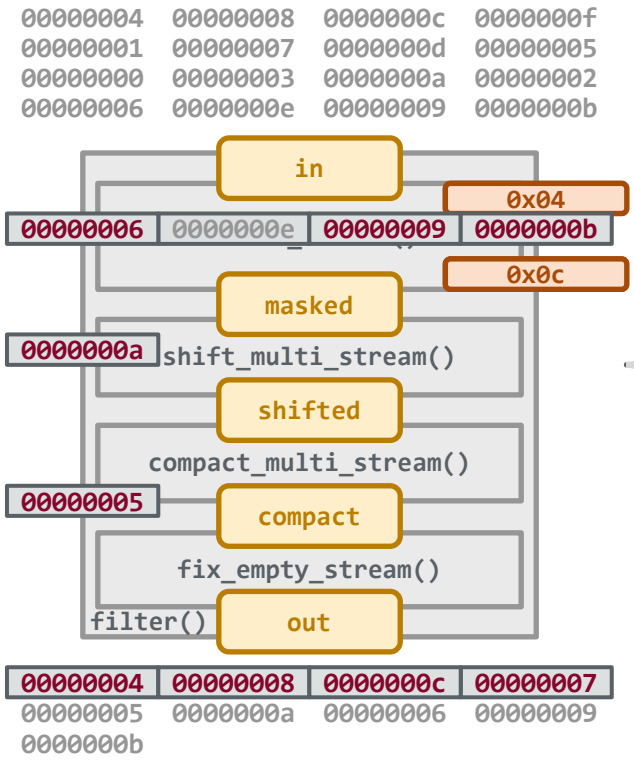



```
void filter(mtl_stream &in, mtl_stream &out,
            Value lower_bound, Value upper_bound) {
#pragma HLS INTERFACE axis port=in name=axis_input
#pragma HLS INTERFACE axis port=out name=axis_output
#pragma HLS INTERFACE s_axilite port=lower_bound bundle=control offset=0x100
#pragma HLS INTERFACE s_axilite port=upper_bound bundle=control offset=0x110
#pragma HLS INTERFACE s_axilite port=return bundle=control
  MaskedStream masked;
  ShiftedStream shifted;
  mtl_stream compact;

#pragma HLS DATAFLOW

  filter_stream(in, masked, lower_bound, upper_bound);

#if VALUE_COUNT > 1
  shift_multi_stream(masked, shifted);
  compact_multi_stream(shifted, compact);
#else
  compact_single_stream(masked, compact);
#endif

  fix_empty_stream(compact, out);
}
```

ParProg 2020
C3 FPGA Accelerators

Lukas Wenzel

Getting started with Metal FS

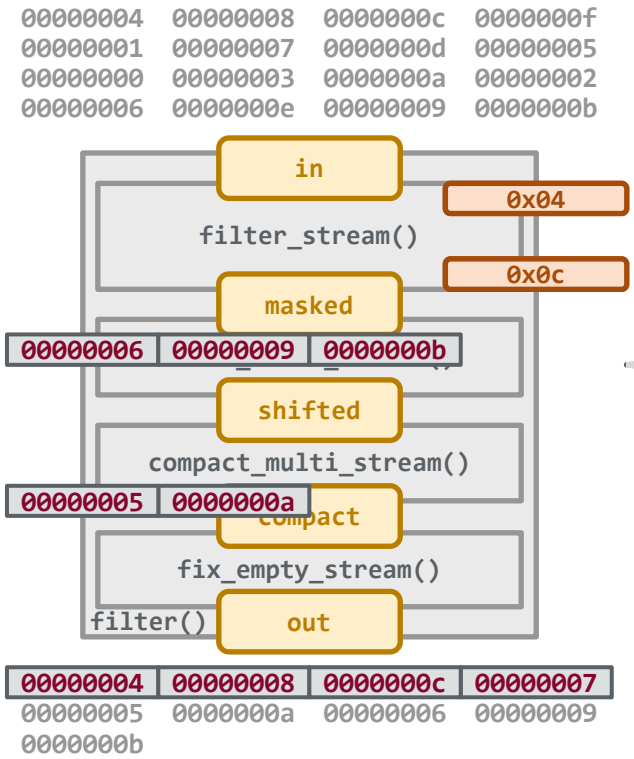



```
void filter(mtl_stream &in, mtl_stream &out,
            Value lower_bound, Value upper_bound) {
#pragma HLS INTERFACE axis port=in name=axis_input
#pragma HLS INTERFACE axis port=out name=axis_output
#pragma HLS INTERFACE s_axilite port=lower_bound bundle=control offset=0x100
#pragma HLS INTERFACE s_axilite port=upper_bound bundle=control offset=0x110
#pragma HLS INTERFACE s_axilite port=return bundle=control
  MaskedStream masked;
  ShiftedStream shifted;
  mtl_stream compact;

#pragma HLS DATAFLOW

  filter_stream(in, masked, lower_bound, upper_bound);

#if VALUE_COUNT > 1
  shift_multi_stream(masked, shifted);
  compact_multi_stream(shifted, compact);
#else
  compact_single_stream(masked, compact);
#endif

  fix_empty_stream(compact, out);
}
```

ParProg 2020
C3 FPGA Accelerators

Lukas Wenzel

Getting started with Metal FS

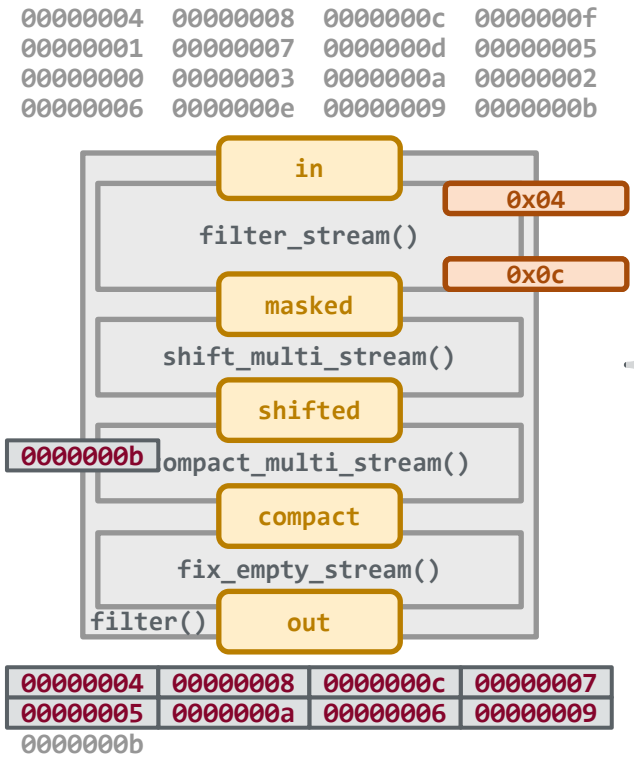



```
void filter(mtl_stream &in, mtl_stream &out,
            Value lower_bound, Value upper_bound) {
#pragma HLS INTERFACE axis port=in name=axis_input
#pragma HLS INTERFACE axis port=out name=axis_output
#pragma HLS INTERFACE s_axilite port=lower_bound bundle=control offset=0x100
#pragma HLS INTERFACE s_axilite port=upper_bound bundle=control offset=0x110
#pragma HLS INTERFACE s_axilite port=return bundle=control
  MaskedStream masked;
  ShiftedStream shifted;
  mtl_stream compact;

#pragma HLS DATAFLOW

  filter_stream(in, masked, lower_bound, upper_bound);

#if VALUE_COUNT > 1
  shift_multi_stream(masked, shifted);
  compact_multi_stream(shifted, compact);
#else
  compact_single_stream(masked, compact);
#endif

  fix_empty_stream(compact, out);
}
```

ParProg 2020
C3 FPGA Accelerators

Lukas Wenzel

Getting started with Metal FS

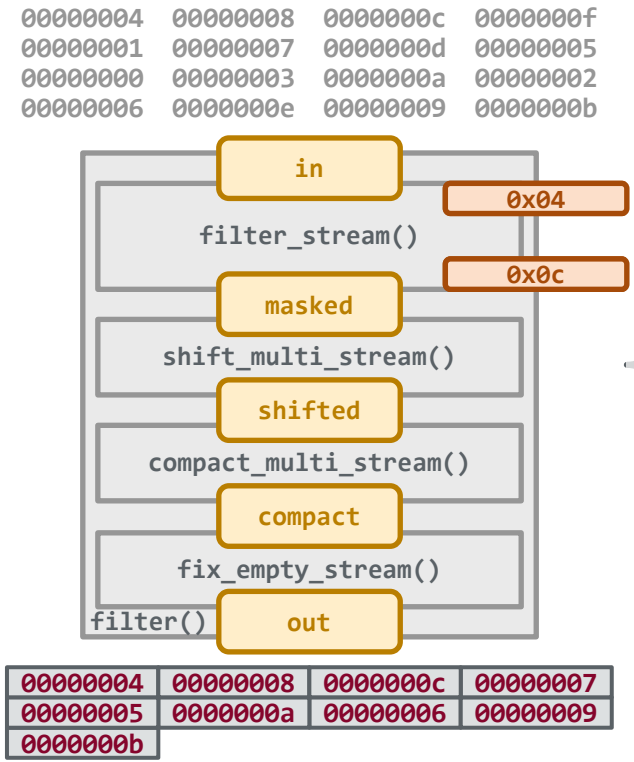



```
void filter(mtl_stream &in, mtl_stream &out,
            Value lower_bound, Value upper_bound) {
#pragma HLS INTERFACE axis port=in name=axis_input
#pragma HLS INTERFACE axis port=out name=axis_output
#pragma HLS INTERFACE s_axilite port=lower_bound bundle=control offset=0x100
#pragma HLS INTERFACE s_axilite port=upper_bound bundle=control offset=0x110
#pragma HLS INTERFACE s_axilite port=return bundle=control
  MaskedStream masked;
  ShiftedStream shifted;
  mtl_stream compact;

#pragma HLS DATAFLOW

  filter_stream(in, masked, lower_bound, upper_bound);

#if VALUE_COUNT > 1
  shift_multi_stream(masked, shifted);
  compact_multi_stream(shifted, compact);
#else
  compact_single_stream(masked, compact);
#endif

  fix_empty_stream(compact, out);
}
```

ParProg 2020
C3 FPGA Accelerators

Lukas Wenzel

Getting started with Metal FS




```
void filter(mtl_stream &in, mtl_stream &out,
            Value lower_bound, Value upper_bound) {
#pragma HLS INTERFACE axis port=in name=axis_input
#pragma HLS INTERFACE axis port=out name=axis_output
#pragma HLS INTERFACE s_axilite port=lower_bound bundle=control offset=0x100
#pragma HLS INTERFACE s_axilite port=upper_bound bundle=control offset=0x110
#pragma HLS INTERFACE s_axilite port=return bundle=control
  MaskedStream masked;
  ShiftedStream shifted;
  mtl_stream compact;

#pragma HLS DATAFLOW

  filter_stream(in, masked, lower_bound, upper_bound);

#if VALUE_COUNT > 1
  shift_multi_stream(masked, shifted);
  compact_multi_stream(shifted, compact);
#else
  compact_single_stream(masked, compact);
#endif

  fix_empty_stream(compact, out);
}
```

ParProg 2020
C3 FPGA Accelerators

Lukas Wenzel

HPI Hasso Plattner Institut
Digital Engineering · Universität Potsdam

And now for a break and some cold brewed Bancha.

*or beverage of your choice