Karlsruhe Institute of Technology

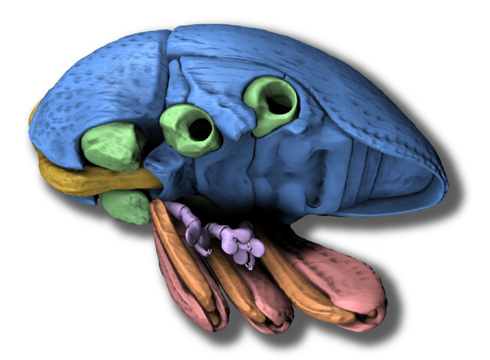
Institute for Data Processing and Electronics
Herrmann-von-Helmholtz-Platz 1
D-76344 Eggenstein-Leopoldshafen

A Parallel Computing Framework for Real-Time Tomographic Reconstruction with GPUs

Matthias Vogelgesang, Suren A. Chilingaryan
matthias.vogelgesang@kit.edu

Motivation

- Computer Tomography (CT) at synchrotron sites reveals fine spatial and temporal details of biological and technological processes
- Therefore, we develop system for
  - Fast image acquisition (beamline and detector),
  - Fast image processing (*this* work) and
  - Fast and big storage
- Real-time CT becomes only possible using GPUs as shown in our previous work [1]

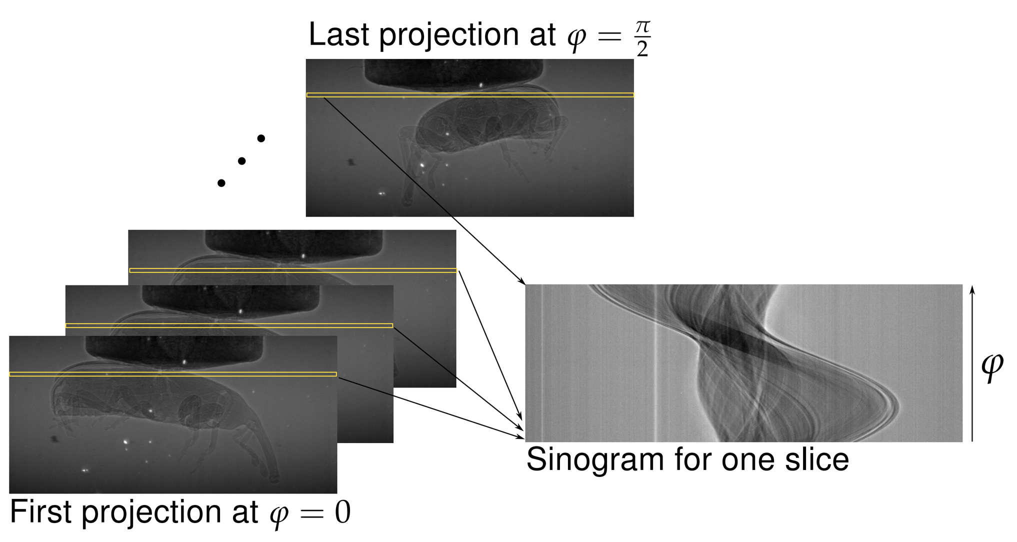


Segmented head of a reconstructed weevil

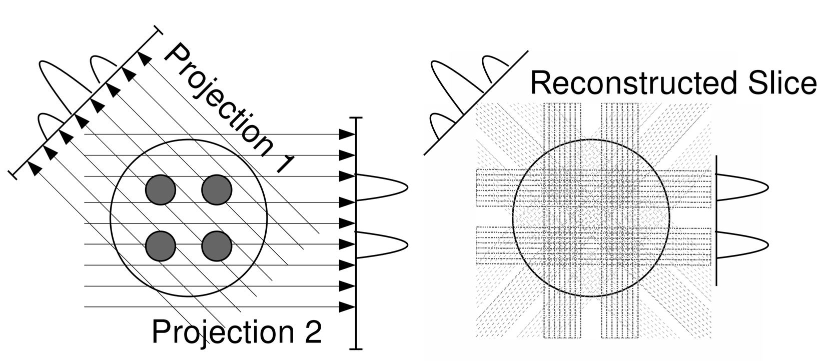
Background

The filtered back-projection (FB) for CT reconstruction consists of the following steps:

1. Reduce noise in projections
2. Generate sinograms from cleaned projections taken at angles between 0 and π
3. Transfer sinograms into Fourier domain using FFT
4. Filter with a high-frequency emphasizing filter
5. Transfer back into spatial domain
6. Back-projection to compute attenuation factors



Artifacts in a generated sinogram



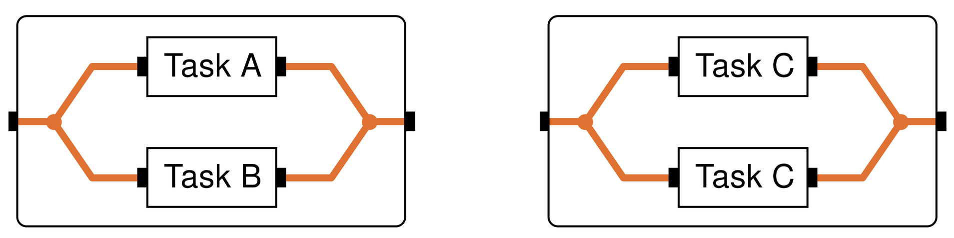
Forward-projection Back-projection

Our Approach

Integrating all reconstruction steps provides opportunities to increase performance and flexibility. Hardware-awareness and

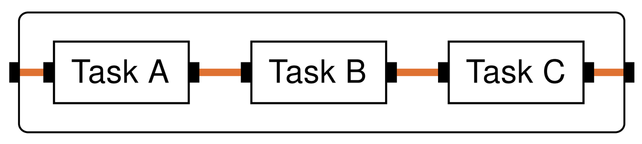
Using an integrated approach offers optimal performance, because expensive writes to disk and unnecessary copy operations between host and GPU can be avoided.

- Our framework is built on top of an *extended* image processing pipeline
- A pipeline stage can be a **Split** container

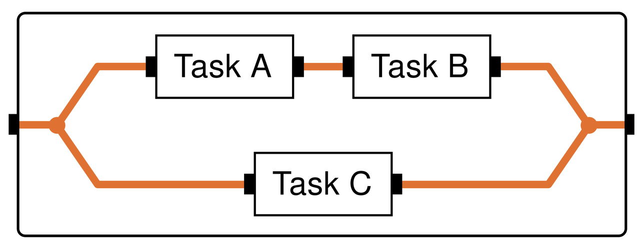


(a) Task Parallel (b) Data Parallel

- A **Sequence** container resembling a classic pipeline and used for overlapping parallelism



- Or a composition using both types



The atomic pipeline stages represent *computational* tasks, which

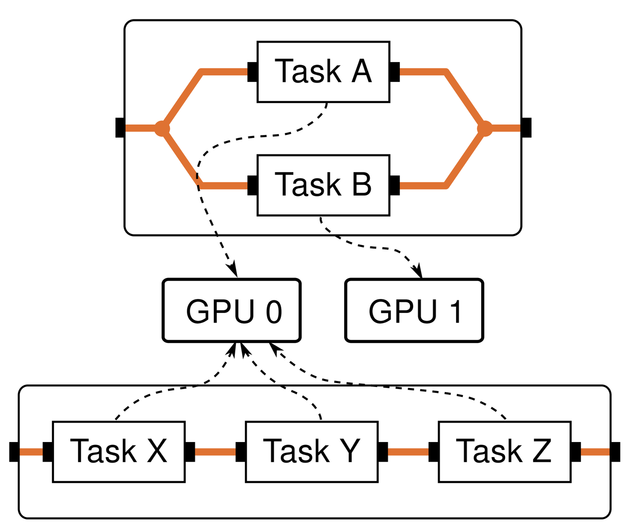
- receive data as pointer to **Buffers** from an input queue,
- process data on host CPU or GPU (by implicit copying),
- push processed data into its output queue.

Implementation

- Implementation of the framework and all CT reconstruction stages using C, OpenCL, GLib and GObject
  - Each step from the reconstruction process is mapped to one task stage utilizing the GPU
- No explicit synchronization in task stages, because of *asynchronous* queues between two stages

Scheduling

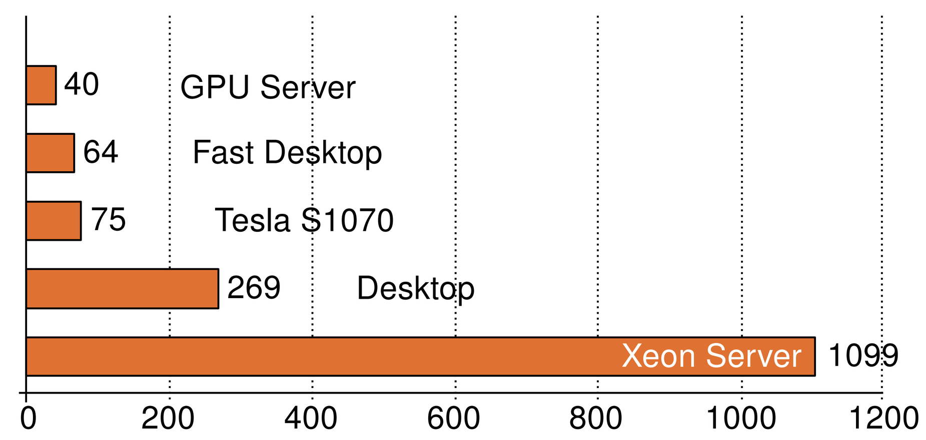
High performance is reached with a combination of multi-threading – each task stage runs in its own thread of execution – and GPU assigment strategies:



- For pipelining preserve data on same GPU
- Otherwise distribute work on different GPUs

Results

- First unified image processing framework for synchrotron CT based on OpenCL
- Fast processing



- Simple configuration


```
{
  "type" : "sequence",
  "elements" : [
    { "type" : "task", "plugin" : "reader" },
    { "type" : "task", "plugin" : "fft",
      "properties" : { "dimensions"  : 1 }
    },
    { "type" : "task", "plugin" : "filter",
      "properties" : { "filter-type" : "ramp" }
    },
    { "type" : "task", "plugin" : "ifft",
      "properties" : { "dimensions"  : 1 }
    },
    { "type" : "task", "plugin" : "backproject",
      "properties" : {
        "axis-pos"   : 413.5,
        "angle-step" : 0.01256637
      }
    },
    { "type" : "task", "plugin" : "writer" }
  ]
}
```


JSON configuration for filtered back-projection

- Fast prototyping

```
from gi.repository import Framework
# construct a pipeline object
pipeline = Framework.Pipeline()
# get default Sequence stage
root = pipeline.get_root()
# add task stages and execute
bp = pipelin.get_plugin('backproject')
bp.set_properties("axis-pos", 413.5, \
  "angle-step" : 0.012566)
root.add_element(pipeline.get_plugin('reader'))
root.add_element(bp)
root.add_element(pipeline.get_plugin('writer'))
root.run()
```

Python example to setup reconstruction

References

[1] S. Chilingaryan, A. Mirone, A. Hammersley, C. Ferrero, L. Helfen, A. Kopmann, T. dos Santos Rolo, and P. Vagovic, "A GPU-Based Architecture for Real-Time Data Assessment at Synchrotron Experiments," *IEEE Transactions on Nuclear Science*, 2011.