JURASSIC - GPU acceleration

Current porting status of a climate code

Lars Hoffmann, JSC; Thorsten Hater, JSC; Benedikt Anlauf, IBM
l.hoffmann@fz-juelich.de, t.hater@fz-juelich.de, anlaufb@de.ibm.com
Juelich Rapid Spectral Simulation Code (JURASSIC)

model description and applications

PADC Opening Workshop, Jülich, 13 October 2015 | Lars Hoffmann (JSC)
What is JURASSIC?

- JURASSIC is a fast solver for atmospheric radiative transfer in the mid-infrared (4 – 15 µm) spectral region.
What is JURASSIC?

- The Earth’s long-wave radiation budget is largely controlled by greenhouse gases and clouds.
What is JURASSIC?

- JURASSIC is used to retrieve atmospheric data from ESA and NASA satellite experiments.
What is JURASSIC?

- Example application for PADC: Retrieval of stratospheric temperature from NASA’s AIRS/Aqua experiment.
  - AIRS provided $\sim$100 TByte IR radiance spectra since 2002.
  - Stratospheric temperatures are retrieved from 4 and 15 $\mu$m radiances with JURASSIC.
  - Data shared with scientists studying atmospheric dynamics.
  - Retrieval takes $\sim$1 s CPU-time per radiance spectrum.
  - Processing of 14 years of data requires $\sim$5 million core-h.
What is JURASSIC?

- Example application for PADC: Retrieval of stratospheric temperature from NASA’s AIRS/Aqua experiment.

- AIRS provided $\sim$100 TByte IR radiance spectra since 2002.

- Stratospheric temperatures are retrieved from 4 and 15 $\mu$m radiances with JURASSIC.

- Data shared with scientists studying atmospheric dynamics.

- Retrieval takes $\sim$1 s CPU-time per radiance spectrum.

- Processing of 14 years of data requires $\sim$5 million core-h.
What is JURASSIC?

- Example application for PADC: Retrieval of stratospheric temperature from NASA’s AIRS/Aqua experiment.

- AIRS provided \( \sim 100 \text{ TByte} \) IR radiance spectra since 2002.

- Stratospheric temperatures are retrieved from 4 and 15 \( \mu \text{m} \) radiances with JURASSIC.

- Data shared with scientists studying atmospheric dynamics.

- Retrieval takes \( \sim 1 \text{ s CPU-time} \) per radiance spectrum.

- Processing of 14 years of data requires \( \sim 5 \text{ million core-h} \).
What is JURASSIC?

- Example application for PADC: Retrieval of stratospheric temperature from NASA’s AIRS/Aqua experiment.
  
  - AIRS provided \( \sim 100 \) TByte IR radiance spectra since 2002.
  
  - Stratospheric temperatures are retrieved from 4 and 15 \( \mu \)m radiances with JURASSIC.
  
  - Data shared with scientists studying atmospheric dynamics.
  
  - Retrieval takes \( \sim 1 \) s CPU-time per radiance spectrum.
  
  - Processing of 14 years of data requires \( \sim 5 \) million core-h.
What is JURASSIC?

- Example application for PADC: Retrieval of stratospheric temperature from NASA’s AIRS/Aqua experiment.

- AIRS provided $\sim 100$ TByte IR radiance spectra since 2002.

- Stratospheric temperatures are retrieved from 4 and 15 $\mu$m radiances with JURASSIC.

- Data shared with scientists studying atmospheric dynamics.

- Retrieval takes $\sim 1$ s CPU-time per radiance spectrum.

- Processing of 14 years of data requires $\sim 5$ million core-h.
What is JURASSIC?

- Example application for PADC: Retrieval of stratospheric temperature from NASA’s AIRS/Aqua experiment.

  - AIRS provided $\sim$100 TByte IR radiance spectra since 2002.
  - Stratospheric temperatures are retrieved from 4 and 15 $\mu$m radiances with JURASSIC.
  - Data shared with scientists studying atmospheric dynamics.
  - Retrieval takes $\sim$1 s CPU-time per radiance spectrum.
  - Processing of 14 years of data requires $\sim$5 million core-h.
How does JURASSIC work?

- Step 1: Apply a raytracer considering refraction $n = n(p, T)$ to determine the line-of-sight (LOS) in the atmosphere.
How does JURASSIC work?

- Step 2: Determine the spectral mean emissivity
  \( \bar{\varepsilon} = \int_{\Delta \nu} \varepsilon_{\nu}(p, T, u) \, d\nu \) at every point of the LOS.

- JURASSIC avoids monochromatic calculations by applying look-up tables for spectral mean emissivity!
How does JURASSIC work?

- Step 3: Solve the radiative transfer equation:

\[ I(x) = I(0) \tau(0, x) + \int_0^x B(x') \frac{d\tau(x', x)}{dx'} \, dx' \]
How does JURASSIC work?

- about 10,000 lines of code (in C)
- efficient MPI/OpenMP hybrid parallelization
How does JURASSIC work?

- Future work will focus on remote sensing of aerosols and clouds.

- Leads to complex radiative transfer calculations due to scattering at particles.
Performance & porting concept
Benedikt Anlauf, IBM
Agenda

About Jurassic

**Performance and kernels**

Prototypes: GPU-suitability

Going forward
Performance and kernels

Kernel:
- Table search and linear interpolation between two values
  - 65% of runtime (~ 5% via different code path)
- raytracing algorithm
  - 5% of runtime
  - directly prior to table lookup
- 15% Init and I/O
- 15% scattered across small functions within control flow

Code structure:
- Deep nested loop structure
- Small number of iterations per loop level

Both kernels not textbook example for GPU acceleration
Porting approach

- **In scope:**
  - Structural alterations, up to complete restructuration of
    - Data structures
    - Control flow
  - New load balancing / work aggregation

- **Out of scope:**
  - Algorithmic alterations (e.g. using high-order splines instead of table search + linear interpolation)

- **First step:** Writing kernel prototypes to assess GPU performance
  - "Prototype": extracting the kernel function from codebase, feed by artificial loop and input data

- Systems used: POWER 8 (10-core), NVIDIA K40
Agenda

About Jurassic

Performance and kernels

Prototypes: GPU-suitability

Going forward
1. Kernel: table search and linear interpolation

Implemented by Thorsten Hater

- Implemented two prototypes:
  - „basic“: just search
  - „mockup“: approximation of complete kernel function (about 75 lines of code)

- Result:
  - speedup against best CPU runtime for sufficiently many parallel searches
  - GPU-kernel dominated by launch overhead (80%)
  - Speedup independent of table size
2. Kernel: Raytracing

- Rather complex function (~500 lines of code)
- Prototype: recompilation of unaltered host function
  - Performance limitors / uncertainties:
    - data access is not optimized for GPUs (structs)
    - branching might not be representative
  - No optimizations here due to small contribution to total runtime
    - GPU port needed for data locality
- speed comparable or faster as full CPU socket for reasonable to slightly large ray numbers
Agenda

About Jurassic

Performance and kernels

Prototypes: GPU-suitability

Going forward
Going forward: concept for further work

- Wrap-up:
  - Code not intuitively portable to GPU
    - Parallelism shadowed by loop hierarchy
    - Not many FLOP/s needed
  - But: prototypes have astonishingly good performance
    - Even though no optimization towards GPU

- Conclusion: good candidate for #pragma-based approach

- Plan (work in progress)
  - Port with OpenACC (PGI-compiler) and OpenMP 4.0 (IBM LLVM-compiler)
  - Direct comparison possible
  - But: Performance might not be representative (alpha/beta-compilers)