

# Astraroath Code



# Astraroth Code



- One of the most scientific of medieval demons. A great and strong duke who rules 40 legions.
- Also an experimental (M)HD code in development, which uses GPU:s for accelerating computation.

# GPU basics

- GPU computation offers potentially high efficiency in computation utilizing many floating point operations.
- There are two frameworks, NVidia CUDA and OpenCL, in operation. CUDA is more efficient for science purposes, at the moment.
- Also a directive based OpenACC has already some limited functionality.
- C/C++ better supported than Fortran.

# How it works

- CPU *host* and GPU *device* are handled separately.
- Host directs the general functioning of the program.
- Device performs specific subroutines called *kernels*.
- Computation within kernels are calculated with parallel threads.

# Challenges: Communication

- Transferring data between host and device is slow.
- Amount of data transfers should therefore be minimized.
- In Astaroth code:
  - Not a big issue with a single GPU, as whole grids are need to be copied only for reading / writing.
  - Requires more consideration, if MPI will be implemented.

# Challenges: Memory management

- CUDA devices have different layers of memory (i.e. global memory, shared memory).
- Optimal performance requires optimal use of memory.
- In Astaroth Code:
  - The most important question at the moment. Memory use makes huge difference at single CPU level.

# Challenges: Code complexity

- Separate code need to be written for GPUs.
- OpenACC is still not yet very efficient and requires a lot of unintuitive tricks.
- If you want to accelerate a function with GPUs, you need to write a corresponding kernel for CPU.
- Requires a lot of work.



# Challenges: Different ways of thinking

- You need to think in terms of multiple threads.  
→ Many for-loops will disappear.
- From learning point of view, requires some new concepts

// Normal CPU function:

```
void calculation(float* A, float* B, float* C) {  
    for (int i=0; i < N; i++) {  
        C[i] = A[i]*B[i];  
    }  
}
```

// A simple CUDA kernel:

```
__device__ void calculation(float* A, float* B, float* C) {  
    int i;  
    i = blockDim.x*blockIdx.x + threadIdx.x;  
    C[i] = A[i]*B[i];  
}
```



# Astaroth Code so far

- The Astraroth Code is at the moment under heavy development this summer.
- Basic functionality (isothermal hydrodynamics) with GPU already exist, but new physics is added at the moment (e.g. shearing).
- The very initial tests have shown  $\sim 30$  times speed-up compared to single CPU version of same operations.
- However, the version tested was not very optimized yet.

# Conclusions

- GPUs might offer significant speed-up to computation.
- Implementation requires a lot of work. Most optimal methods have possibly not even been invented yet.
- More results coming until the end of the year.

Thank you!