C2SM Center for Climate Systems Modeling

ETH Eidgenössische Technische Hochschule Zürich Swiss Federal Institute of Technology Zurich

Porting the COSMO v5 Fortran code to GPU

Stefan Rüdisühli

Swiss COSMO User Workshop

3 December 2014

Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra

Eidgenössisches Departement des Innern EDI
Bundesamt für Meteorologie und Klimatologie MeteoSchweiz

CSCS Swiss National Supercomputing Centre

COSMO goes GPU: Overview

- Different approaches:
  - Dynamical core: complete re-write (C++)
  - Physics/assimilation/etc.: extend Fortran code (OpenACC)

- Reference: OPCODE (GPU prototype based on COSMO v4.X)
  - Started over with COSMO v5.0

- My areas of work:
  - radiation
  - assimilation
  - output
  - single precision

C2SM Center for Climate Systems Modeling

OpenACC

- Directives: special comments, ignored on CPU
- Original code largely unaltered
- Separate CPU and GPU memory
- Relatively new technique

OpenACC

- Directives: special comments, ignored on CPU
- Original code largely unaltered
- Separate CPU and GPU memory
- Relatively new technique

```
DO k=1,ke
  DO j=1,je
    DO i=1,ie
      p(i,j,k) = p0(k)+pp(i,j,k)
    ENDDO
  ENDDO
ENDDO
PRINT*, MINVAL(p(:,:,ke))
```

OpenACC

- Directives: special comments, ignored on CPU
- Original code largely unaltered
- Separate CPU and GPU memory
- Relatively new technique

```
DO k=1,ke
  DO j=1,je
    DO i=1,ie
      p(i,j,k) = p0(k)+pp(i,j,k)
    ENDDO
  ENDDO
ENDDO
PRINT*, MINVAL(p(:,:,ke))
```

```
!$acc data present (p,p0,pp)
…
!$acc parallel
DO k=1,ke
  !$acc loop gang
  DO j=1,je
    !$acc loop vector
    DO i=1,ie
      p(i,j,k) = p0(k)+pp(i,j,k)
    ENDDO
  ENDDO
ENDDO
!$acc end parallel
…
!$acc update host (p(:,:,ke))
PRINT*, MINVAL(p(:,:,ke))
…
!$acc end data
```

OpenACC

- Directives: special comments, ignored on CPU
- Original code largely unaltered
- Separate CPU a
- Relatively new t

```
DO k=1,ke
  DO j=1,je
    DO i=1,ie
      p(i,j,k) = p0(k)+pp(i,j,k)
    ENDDO
  ENDDO
ENDDO
PRINT*, MINVAL(p(:,:
```

start data region

start parallel region

parallelization detail

parallelization detail

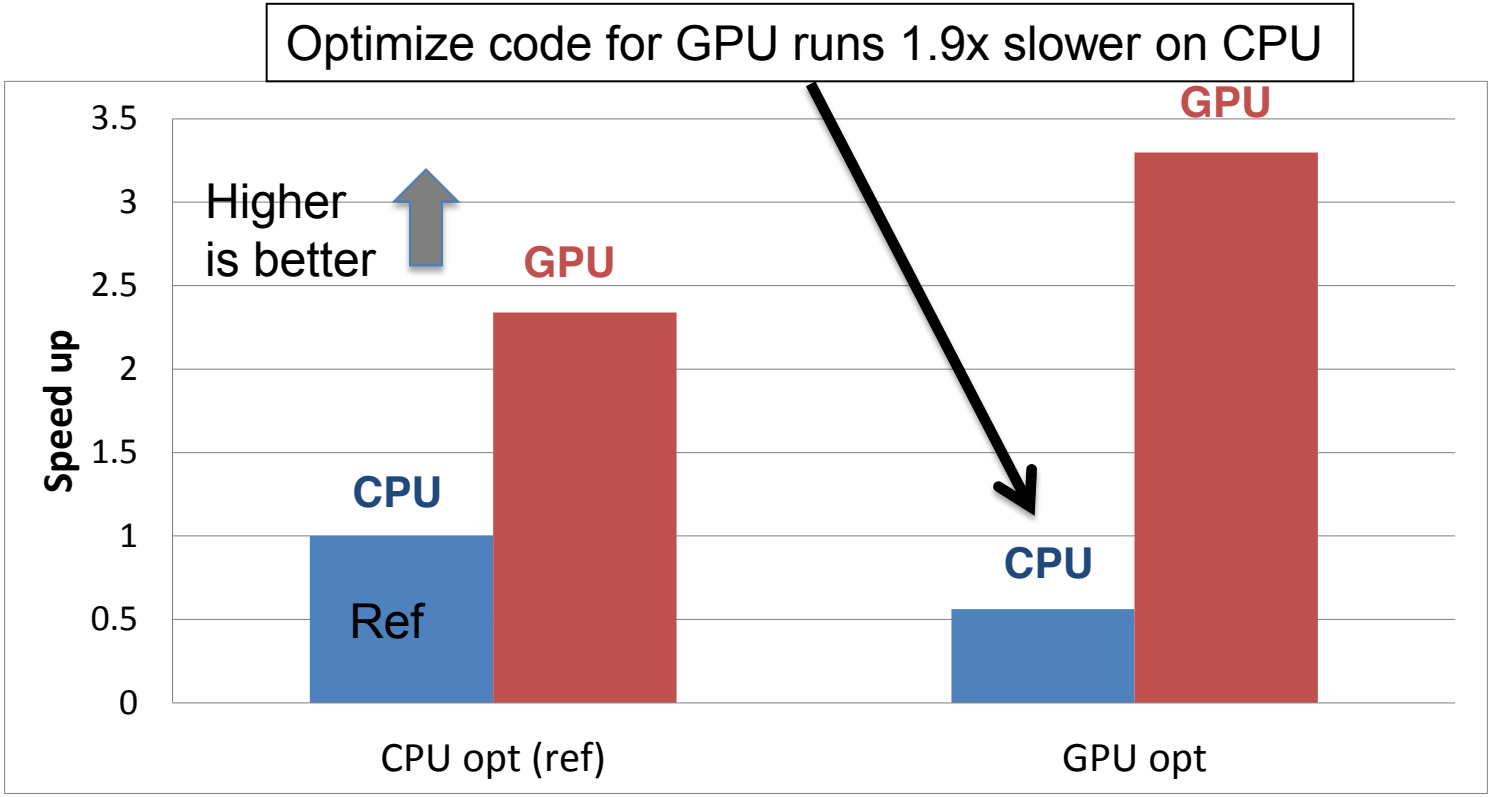
end parallel region

update CPU array

end data region

```
!$acc data present (p,p0,pp)
…
!$acc parallel
DO k=1,ke
  !$acc loop gang
  DO j=1,je
    !$acc loop vector
    DO i=1,ie
      p(i,j,k) = p0(k)+pp(i,j,k)
    ENDDO
  ENDDO
ENDDO
!$acc end parallel
…
!$acc update host (p(:,:,ke))
PRINT*, MINVAL(p(:,:,ke))
…
!$acc end data
```

Radiation performance



- Key kernel of radiation: Speed-up with respect to reference code on CPU
- Radiation is the strongest example, on average in the physics, code optimized for GPU is 1.3x times slower when run on CPU

[Xavier Lapillonne]

Thanks for your attention