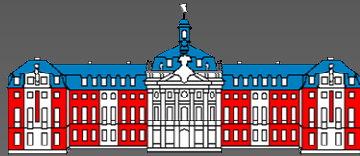


IVV Naturwissenschaften  
IV der Fachbereiche Biologie • Chemie • Physik

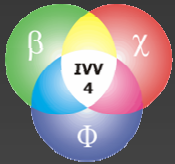


Westfälische  
Wilhelms-Universität  
Münster

# OpenMP auf der Superdome

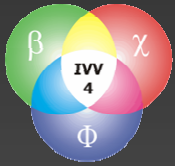
- Paralleles Rechnen
- Die Superdome
- Einloggen
- Job starten
- PBS-Skript
- OpenMP
- Beispielprogramm





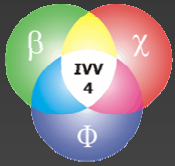
# Paralleles Rechnen

- Für welche Probleme ist Parallelisierung sinnvoll bzw. notwendig?
- Für komplexe Probleme bei denen einzelne Programmabschnitte miteinander „kommunizieren“ müssen!
- Unterschiedliche Realisierungen
  - Cluster (MPI)
  - Shared-Memory-Systeme (OpenMP)



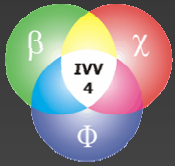
# Die Superdome

- Shared-Memory-Parallelrechner mit 24 Dual-Core-Itanium-2-Prozessoren
- 4 Partitionen:
  - Linux:
    - NWZSuperdome (4 CPUs)
      - Startpartition
    - NWZSuperbatch (16 CPUs)
      - Für umfangreiche vorher getestete Programme
    - NWZSupertest (2 CPUs)
      - Für den Testlauf
  - Windows:
    - NWZSuperWin (2 CPUs)



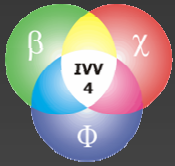
# Einloggen

- Einloggen ist nur mit einem NWZ-Account auf der NWZSuperdome per ssh möglich
- Hierbei wird diese Kennung mit zugehörigem Passwort abgefragt
- Benutzerprofil wird angebunden
- Schreiben auf `\scratch` oder `\work` möglich



# Job starten

- Kompilieren (Intel Fortran Compiler 9.1)
  - `ifort -o programm.exe programm.f90`
- Queueing-System Torque
- PBS-Skript: `job.pbs`
- Abschicken: `qsub job.pbs`
- Überwachen: `qstat`



# PBS-Skript

```
#PBS -q test (bzw. batch)
```

```
#PBS -N Projektname
```

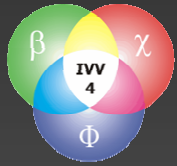
```
#PBS -e error.txt
```

```
#PBS -o output.txt
```

```
#PBS -m abe
```

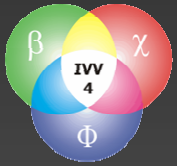
```
#PBS -M abc@uni-muenster.de
```

```
./Projektpfad/Datei.exe
```



# OpenMP

- OpenMp ist unter C/C++ und Fortran nutzbar
- In diesem Vortrag wird nur Fortran behandelt
- Konzept eines (Master-)threads
- Initiieren einer parallelen Umgebung:  
!\$OMP PARALLEL
- Beenden einer parallelen Umgebung:  
!\$OMP END PARALLEL



# OpenMP

□ Anzahl der Threads festlegen:

CALL OMP\_SET\_NUM\_THREADS(n)

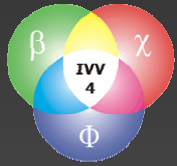
□ Interne Funktionen:

■ Anzahl der Threads innerhalb eines Teams

OMP\_GET\_NUM\_THREADS()

■ Nummer eines Threads in einem Team

OMP\_GET\_THREAD\_NUM()



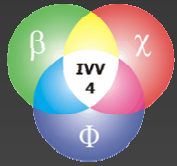
# Beispielprogramme

## Kompilierung:

- `ifort -openmp -o programm.exe programm.f90`

## Helloworld mit OpenMP

## parallelisierte Schleife



# OpenMP

## □ Schleifenkonstrukt:

```
!$OMP DO
```

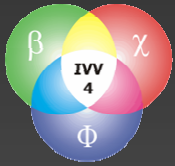
```
!$OMP END DO
```

## □ Weitere Optionen bzw. Details:

- Verbinden verschiedener OMP-Befehle
- PRIVATE, SHARED, REDUCTION Optionen
- DYNAMIC, NESTED Konstruktionen

## □ Autoparallelisierung:

- Ifort ... -parallel ...



# Quellen

## □ IVVNWZ-Homepage:

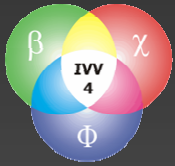
- <http://www.uni-muenster.de/IVVNWZ/computing/superdome/index.html>

## □ Offizielle Homepage

- <http://www.openmp.org/drupal/>

## □ OpenMP-Standard

- <http://www.openmp.org/drupal/mp-documents/spec25.pdf>



Vielen Dank für Ihre Aufmerksamkeit