Dieter an Mey
Center for Computing and Communication, RWTH Aachen University, Germany
anMey@rz.rwth-aachen.de
Overview

• Setting the Scene
• The Time Scale
• Is OpenMP easy?
• How about NUMA, clusters, memory hierarchies?
• Does OpenMP scale?
• Summary
Setting the scene

• My perspective:  
  HPC support in a technical university  
  (*engineering, natural sciences ...*)

• 4 ... 144-way compute servers  
  1000s jobs per day  
  8-16 processors per job on the average

• MPI - OpenMP – Autoparallel – Hybrid
The Time Scale

today OpenMP running on all big SMPs:
  Cray X, Fujitsu Primepower, HP Superdome, IBM Regatta,
  NEC SX, SGI Altix, Sun Fire => OpenMP is expensive
2003  dual processors Intel boxes with hyperthreading
       for 4 threads, not scalable, but cheap
2004  4-way Opteron machines attractive (NUMA!!)
2005  16-way Opteron boxes at a very competitive price
       with many OpenMP compilers on Lin/Win/Sol available
2006  dual core processors everywhere (incl. laptops)
2007  low latency networks are getting commodity
       OpenMP on Infiniband-Clusters
2008  4-8-core processors
       => 8-32 – way systems affordable
2009  OpenMP V3.0 available for the end user

Are we keeping pace?
Is OpenMP easy?

- Yes, you can easily run into data races.
- We need data race detection / prevention tools in the development cycle.
- The default shared strategy for global data may be inadequate

```plaintext
!$OMP DEFAULT(SHARED|PRIVATE|NONE)
!$OMP THREADSHARED(list)
!$OMP THREADPRIVATE(list)
```

requires a compiler switch? (Fujitsu: frt –Kprivate …)

- autoscopying = cooperating with the compiler

```plaintext
!$OMP PARALLEL DEFAULT(AUTO)
```
Automatic Scoping – Sometimes it would really help!

```c
!$omp parallel default(__auto) do
   do i = is,ie
      ---- 1600 lines omitted ----
   end do
   do i = is,ie
      ---- 1600 lines omitted ----
   end do
```

---
How about NUMA, clusters, memory hierarchies?

• Frequently first-touch is not the way to go! If data is initialized by the master process (e.g. by reading from files), data needs to be migrated (automatically or manually)

• !$OMP NEXTTOUCH(*|var_list)
  is easy to understand, nothing breaks, if it is not supported
  => we’ll have to wait for the migration

• How about
  !$OMP PREFETCH(var_list, [urgency])

Stef Salvini, EWOMP 2003
http://www.rz.rwth-aachen.de/ewomp03/omptalks/Tuesday/Session8/ewomp_salvini.mpg
NEXTTOUCH works

Stream saxpying in C/C++ on a 4-way Opteron system running Solaris
Does OpenMP scale well?

• Well, occasionally ... rarely
• How are we going to use all these nice 8..32-way boxes in the near future?
• Parallelizing while loops
  The proposal is on the table since the very beginning:
  #pragma omp taskq
  (KAI guidec/guidec++, Intel icc)
• Better support of nested parallelism.
  Can we efficiently use OpenMP encapsulated in libraries?
  User code calling a
  Library function (e.g. Newton alg.) calling
  User function
OpenMP Nested - orientation

!$OMP PARALLEL (name)
omp_get_thread_num(name)

level = omp_get_parallel_level()
omp_get_thread_num(level)

! Get thread id of ancestors
do level = 0, omp_get_parallel_level()
    print *, omp_get_thread_num(level)
end do
OpenMP Nested - threadprivate

- The **threadprivate** directive specifies that named global-lifetime objects are replicated, with each thread having its own copy.
- There needs to be a way to suppress additional copying in a lower level of nested parallelism.

```fortran
SUBROUTINE SUB()
  COMMON /T/ A(1000000000)
  !$OMP THREADPRIVATE(/T/)
  !$OMP PARALLEL COPYIN(/T/)
  ... 
  !$OMP PARALLEL WORKSHARE SHARED(/T/)
  A = ... 
  !$OMP END PARALLEL WORKSHARE
  ... 
  !$OMP END PARALLEL
END SUBROUTINE SUB
```
Summary

• We need to hurry …
• Data race detection / prevention tools
• Autoscopying
• Task queues
• Nexttouch / prefetch
• Better nested support
  – Orientation
  – threadprivate