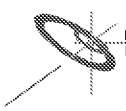
Tools for the Development of Parallel Programs

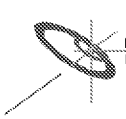
Isabel Loebich
Michael M. Resch

loebich@hlrs.de
resch@hlrs.de



Platform-independent Tools

- Program Analysis
  - FORGExplorer
- Parallelization
  - FORGE/DMP, xHPF
  - FORGE/SMP, spf
- Performance Analysis
  - VAMPIRtrace, VAMPIR
- Metacomputing
  - PACX-MPI



FORGE Tools

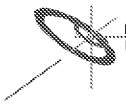
FORGExplorer: Fortran Browser

- Graphical user interface
- Global database representing the complete program
- Program analysis by
  - tracing usage of variables and calls to routines
  - searching in-context for variables
  - viewing the hierarchical tree of subprogram calls and enclosed DO loops
  - viewing COMMON block structures across routines
  - determining consistency of COMMON blocks and subprogram arguments
- Code reformatting
- Code instrumentation facility: runtime performance statistics on CPU time spent in each program unit and each DO loop

FORGE Parallelizer

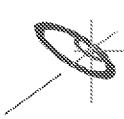
- Fortran Parallelizer
  fully automatic or semi-automatic mode
- Loop spreading interface
- Consistency checker

- Distributed memory
  - Array partitioning interface
- Shared memory
  - Array scoping interface

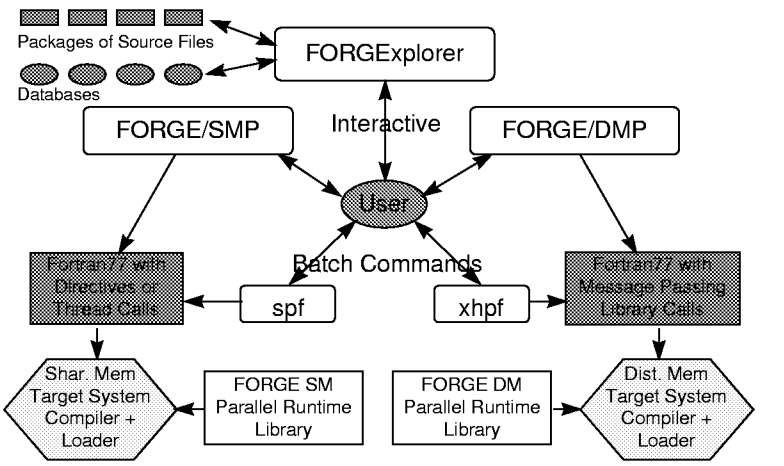


FORGE Parallelizer

- Parallel code generator
  - Message Passing Calls (DM parallel runtime library)
  - High Performance Fortran (HPF)
  - Compiler directives for shared memory
- FORGE/DMP is the interactive version of the parallelizing pre-compiler xHPF
- FORGE/SMP is the interactive version of the parallelizing pre-compiler spf



FORGE Tools

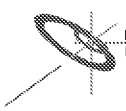


FORGE at HLRS (1)

- for workstations in /client/bin (afs, dfs)
  - forgex: interactive version with all components (browser, SMP and DMP)
  - xhpf: distributed memory parallelizer (batch)
  - spf: shared memory parallelizer (batch)
  - serial timing analysis: link instrumented code to -L/client/lib -ltsf
  - polytime: parallel performance profiler

FORGE at HLRS (2)

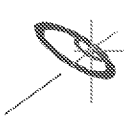
- for Paragon and CRAY T3E: set USE_FORGE
  - serial timing analysis: link instrumented code to
    -L$FORGETSFlib -ltsf
  - DM runtime library: link parallel code to
    -L$FORGEDMPlib -lmpf (standard) or
    -L$FORGEDMPlib -ltpf (parallel timing analysis)



FORGE at HLRS (3)

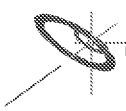
- Documentation:
  - man pages in /client/man
  - Postscript files in /client/doc/forge-2.1
  - Paragon and CRAY T3E: Postscript files in
    $FORGEdocu/TECHNOTE/

- http://www.hlrs.de/support/tools/browser
- http://www.hlrs.de/support/tools/parallelizer



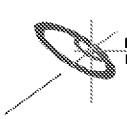
Vampir

Visualisation and Analysis of MPI Resources



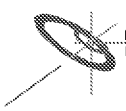
VAMPIR / Performance Analysis

- for MPI programs
- displays post-mortem trace files
- Trace data collection:
  - VAMPIRtrace
  - converters for other trace file formats
- Performance analysis on a workstation



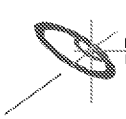
VAMPIRtrace

- MPI profiling library that generates VAMPIR traces
- based on MPI profiling interface
- records calls to MPI functions
- tracing can be enabled / disabled during runtime
- supports Fortran 77, Fortran 90, C, and C++ applications
- manual instrumentation
  - subroutine entry / exit events
  - enable / disable tracing, set trace options



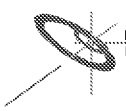
VAMPIR / Views

- Timeline diagram (global and process local)
- Statistics display (global and process local)
- Communication matrix (global)
- System activity snapshot (global)
- System activity animation (global)
- Zooming and Filtering
  - arbitrary time intervals
  - mouse-based zooming
  - arbitrary process sets
  - selection of messages
  - arbitrary selection and grouping of subroutines



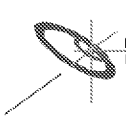
VAMPIR / Global Timeline Diagram

- one timeline per process
- color coded subroutine class
- subroutine name or number depending on screen space
- message lines connecting sender and receiver
- additional message information on request
- acts as master display for statistics displays



VAMPIR / Statistics Display

- all or single processes
- pie chart, histogram or table
- linear or logarithmic scale
- hide dominating parts
- number of calls or execution time (absolute / relative)
- selection of arbitrary time interval using a timeline diagram

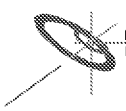


VAMPIR / Communication Matrix

for each sender / receiver pair:

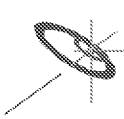
- message lengths
  - minimum
  - maximum
  - sum
  - average
- transmission rates
  - minimum
  - maximum
  - average

selection of arbitrary time interval using a timeline diagram



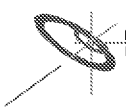
VAMPIR / System Activity

- current subroutine class and subroutine
- processor utilization for a subroutine class
- snapshot or animation
- selection of arbitrary time interval using a timeline diagram

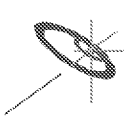


VAMPIR at HLRS

- for workstations in /client/bin (afs, dfs):
  vampir
- for Paragon, NEC SX-4, and CRAY T3E:
  set USE_VAMPIR, link to:
  -L$VAMPIRpath -lVT -lmpi (Paragon, NEC SX-4)
  -L$VAMPIRpath -lVT -lpmpi (CRAY T3E)
- Documentation:
  - VAMPIR man pages in /client/man
  - Postscript files in /client/doc/vampir-1.0
  - Paragon, NEC SX-4, and CRAY T3E:
    $VAMPIRdocu/userguide.ps

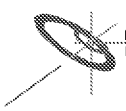


Vampir Demo

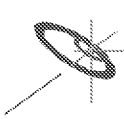


Demo Example

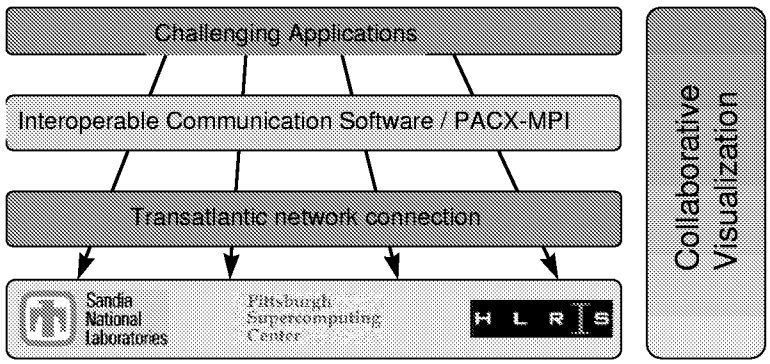
- The Parallel CFD Test Case
  - Euler equation: df/dt = del2(f)
  - Explicit scheme
  - Cube with unit dimension
  - Initial condition: f = 0
  - boundary condition: f=x



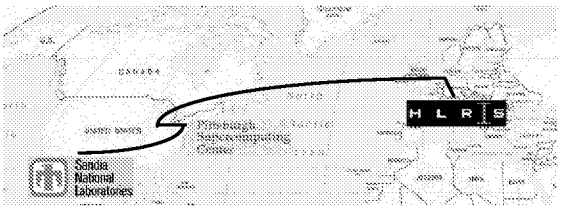
PACX-MPI



G-WAAT
Global Wide Area Application Testbed



Transatlantic network connection



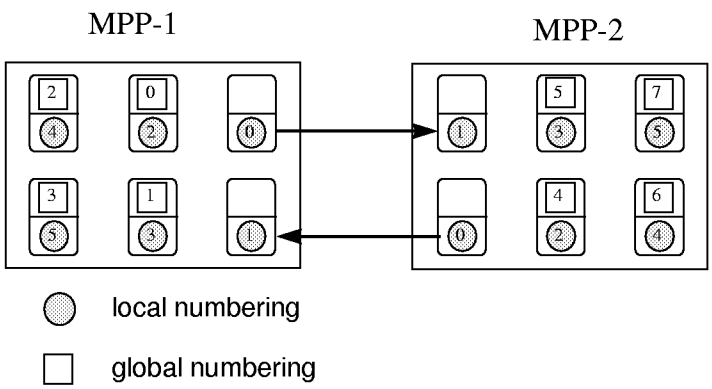
vBNS
STARTAP
TELEGLOBE

Transatlantic network connection

Goals of PACX-MPI

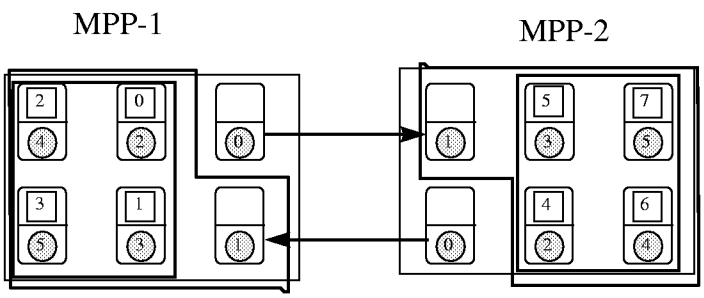
- Coupling of MPPs
- No change in code
- No extensions to MPI
- Usage of vendor implemented MPI for internal communication
- Usage of standard protocol for external communication
- Implementation according to applications' needs
  no limitation in problem size or machine size

Concept of PACX-MPI



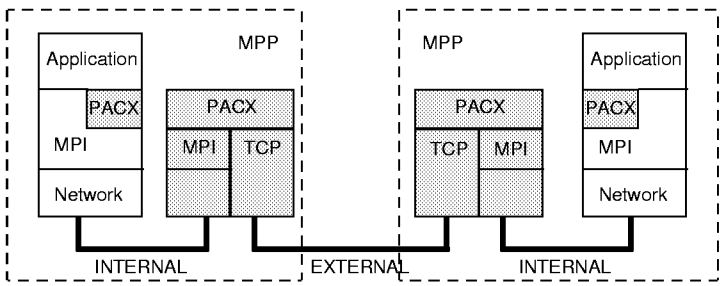
Concept of PACX - MPI

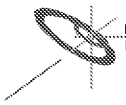
On each machine there are two communicators



Concept of PACX-MPI

Layered approach



MPI calls available

- Initialization and environment control:
  - MPI_Init,MPI_Finalize, MPI_Abort, MPI_Comm_rank, MPI_Comm_size
- Point-to-point communication:
  - MPI_Send, MPI_Recv, MPI_Bsend
  - MPI_Isend, MPI_Irecv
- Collective operations:
  - MPI_Barrier, MPI_Bcast, MPI_Reduce, MPI_Allreduce