Computer Aided Parallelization Tools CAPTools

Richard Luczak
ASC MSRC PTES Lead
University of Tennessee

June 5, 2000
Outline

• Requirements
• History
• What is CAPTools?
• Features (main components, input codes, design, dependencies)
• Parallelization (steps, session, start CAPTools, compile/link/run)
• Examples (applications, acceleration, benefits, info)
Requirements

• DoD users need help in parallelizing codes

• Accessing/harnessing new scalable systems

• Tools to convert serial Fortran code to portable parallel source form:
  - rapidly
  - almost automatically
  - accurately
  - efficiently and scalably

• Using either OpenMP directives or MPI calls or any combination
History

- Jan 98 - Government sponsored visit from Prof. Mark Cross, Director of Research at the Univ of Greenwich, London, UK
- Aug 98 - collaborative evaluation of CAPTools defined and proposed
- Oct 98 - PET funding for tech support approved
History (cont.)

• Dec 98 - collaboration between UG and NASA Ames incorporated into ASC project through use of OpenMP directives

• Apr 99 - MPI/OpenMP beta release provided to ASC

• Evaluation status - UGC 99

• 3 workshops by UG
What is CAPTools?

A software environment to transform Fortran source code from scalar to parallel source form with message passing MPI or OpenMP directives in an interactive semi-automatic way.

INPUT | CAPTools | OUTPUT
---|---|---
Fortran77 code | Transformation to parallel code | Fortran77 code + MPI calls OR + OpenMP Directives
Features

• Main components
• Input codes
• Design
• Dependencies
Main components

- State-of-the-art interprocedural dependence analysis techniques
- In depth analysis using symbolic algebra methods
- User supplied knowledge to assist parallelization
- GUI interactive front end
- Automatic user directed array partitioning
- Automatic parallel source code generation
Input codes

- Fortran 77 (Fortran 90/95 in development)
- Multiple program files
- Include statements
- Conditional compilation directives should be preprocessed
## Design

### CAPTools generated parallel source code

<table>
<thead>
<tr>
<th>CAPLib API</th>
<th>MPI</th>
<th>PVM</th>
<th>Cray SHMEM</th>
<th>Transtech i860toolset</th>
</tr>
</thead>
</table>
Dependencies

- **True**

  \[ A(I) = \ldots = A(I) \]

- **Control**

  \[
  \text{IF} \ (A(I) \ . \text{EQ} \ 5) \ \text{THEN} \\
  A(I) = A(I) + B(J)
  \]

- **Anti**

  \[ \ldots = A(I) \]
  
  \[ A(I) = \ldots \]

- **Output**

  \[ A(I) = \ldots \]
  
  \[ A(I) = \ldots \]
Parallelization

• Steps
• Session
• Start CAPTools
• Compile/link
• Run
Steps

• Load Fortran 77 code
• Volunteer knowledge to the system
• Perform dependence analysis
Steps (cont.)

• Partition the data
• Generate control masks
• Generate communication calls
Steps (cont.)

• Generate parallel source code
• Compile/link/run the code
• Validate the results
Session
This copy of CAPTools is licenced to ASC WPAFB

Welcome to Computer Aided Parallelisation Tools (CAPTools) (c)

(c) Copyright 1992-1998 Parallel Software Products Ltd.

Rights of use
-------------
This is a Beta version of CAPTools. You should not transmit or copy this program
in any form to any other company or individual without the consent of the
owners of the code. Your right to use this Beta version of CAPTools ends on
the licence expiry date or with the first release of a commercial product, which
ever is earlier.

Using CAPTools
-------------
The user manual provides you with a full introduction to all the concepts used
by CAPTools and includes several tutorials. Use the on-line help to obtain help
in using each window.

More information on CAPTools can be obtained from:

Parallel Processing Research Group
Faculty of Science and Technology
University of Greenwich
Wellington Street
Woolwich       Tel : +44 (0)181-331-8731/8655  Web : http://captool.gre.ac.uk
London SE18 6PF  Fax : +44 (0)181-331-8665       Email : captools@gre.ac.uk

To report bugs and get user support please email: captool-support@gre.ac.uk
30 Integer Vars are READ:

JITEM: GRID:17:read(UNIT=3) JITEM ,ktem,ltem
KITEM: GRID:17:read(UNIT=3) ktem, KTEM
LITEM: GRID:17:read(UNIT=3) JITEM, KITEM, LITEM
IOUT: GRIDGEN:18:read(UNIT=5,FMT=*) iout,iprint, IOUT
IPER: GRIDGEN:18:read(UNIT=5,FMT=*) iper, iper, iout
IPRINT: GRIDGEN:18:read(UNIT=5,FMT=*) iper, IPRINT, iout
ISTR1: GRIDGEN:18:read(UNIT=5,FMT=*) istr1, istr2
ISTR2: GRIDGEN:18:read(UNIT=5,FMT=*) istr1, istr2
NB1: GRIDGEN:14:read(UNIT=5,FMT=*) NB1 ,nb2,rf,ra,rx,xr,xmax,drad

Positive Nontrivial Default: 5
Negative Nontrivial Default: -5

User knowledge: LITEM
• Knowledge - combines knowledge entered by the user and that built up by the dependence analysis

• Disproofs - proves inequalities in dependence tests false

• Interprocedural - performs full interprocedural analysis
• Exact - performs tests to set the exact dependence attribute to dependencies

• Scalar - performs symbolic variable equality tests

• Logic - substitutes symbolic variables through the dependence graph
Choose MAXIMUM of:

Defining Statements:

Call Path For Selected Statement:

Relevant Code: JACOB

1:    REAL T(0:1001), TNEW(1000), TOL
2:    INTEGER N
3:    PRINT *, 'Enter the values of N and TOL ...'
4:    READ *, N, TOL
5:    DO 10 I=1, N, 1
6:       T(I)=0.0
7:  10    CONTINUE
8:    40    CONTINUE
9:    DO 20 I=2, N-1, 1
10:    TNEW(I)=(T(I-1)+T(I+1))/2.0
11:  20    CONTINUE
12:    TNEW(1)=T(2)/2.0
13:    TNEW(N)=(T(N-1)+100.0)/2.0
14:    DIFMAX=0.0
15:    DO 30 I=1, N, 1
16:    DIFF=ABS(TNEW(I)-T(I))
17:    IF (DIFF GT DIFMAX) THEN
18:       DIFMAX=DIFF
19:    ENDIF
Start CAPTools

- Login to SGI Origin 2000 at ASC (hpc03.asc.hpc.mil)

- Type in:
  - capo (OpenMP)
  - capttool (MPI)
Compile/link

Code with OpenMP on O2K:

- \texttt{f77 -mp file.f}
- \texttt{f90 -mp file.f}
Compile/link (cont.)

Code with MPI on O2K:
- `capf77 -msgi_64 -p mpi file.f`

Code with MPI on SP2:
- `capf77 -msp2 -p mpi file.f`

Use "`capf77 -help p`" for more info
Run

OpenMP, interactive, SGI O2K:

```
set env MP_SET_NUMTHREADS 2
a. out
```
Run (cont.)

MPI, interactive, O2K:
```
caprun -m sgi_64 -p mpi \ 
   -top pipe2 a.out
```

MPI, interactive, SP2:
```
caprun -m sp2 -p mpi \ 
   -top pipe2 a.out
```

Use "caprun -help" for details
Examples

• Applications
• Accelerating code parallelization with CAPTools
• Benefit to DoD
• More info
Applications

• N-Body
• PFEM
• R-Jet
N-Body

• Distributes points evenly on the unit sphere
• 3D n-body model
• Hand-written MPI
• CAPTools generated OpenMP
Speedup Curves for the N-Body Code
PFEM

- Solves 2D highly nonlinear BVP in CSM
- Domain decomposition (two-coloring)
- Hand-coded directive-driven
Speedup Curves for the PFEM Code
R-Jet

• Simulates vortex dynamics and breakdown in turbulent jets
• Hybrid, high-order, compact finite difference spectral method
Speedup Curves for the R-Jet Code
# Accelerating code parallelization with CAPTools

<table>
<thead>
<tr>
<th></th>
<th>Code analysis</th>
<th>Parallel strategy</th>
<th>Parallel code generation</th>
<th>Numerical validation debugging</th>
<th>Optimization</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hand</td>
<td>1 mth</td>
<td>1 mth</td>
<td>3 mths</td>
<td>6 mths</td>
<td>3 mths</td>
</tr>
<tr>
<td>CAP Tools</td>
<td>2 days</td>
<td>2 days</td>
<td>&lt; 1 hour each time</td>
<td>1 week</td>
<td>1 week</td>
</tr>
</tbody>
</table>

Parallelization process reduced from ~1 year to less than 1 month
Benefit to DoD

- Effective utilization of SMPs and DMPs now and in any combination in future
- Minimize the time required to parallelize codes
- Enable a range of parallelization strategies
- Produce scalable and efficient codes
- Provide more reliable simulation of complete warfighting systems

http://captools.gre.ac.uk

luczakr@asc.hpc.mil