pallas

Gesellschaft für Parallele Anwendungen und Systeme mbH

OpenMP Tools — Assure

Hans-Joachim Plum, Pallas GmbH
edited by Matthias Müller, HLRS

Pallas GmbH
Hermülheimer Straße 10
D-50321 Brühl, Germany

info@pallas.de
http://www.pallas.com

Overview: Assure and AssureView

- Assuref77/f90: OpenMP compliant, restriction in usage of OMP library (in particular: OMP_GET_THREAD_NUM)
- Use as normal compiler, but not for getting performance (small input data set)
- Multithreaded run is simulated sequentially, all memory accesses verified
- Run AssureView to visualize error breakdown. When "No Errors" are reported, multithreaded run is assured free of semantical errors as explained below, but only in the branches touched by the simulation run.

What can be detected: invoke Assure

- Compile code with yet another compiler:

```
assuref90 [options] -o myprog myprof.f90
```

- Execute (but don‘t expect performance!!)

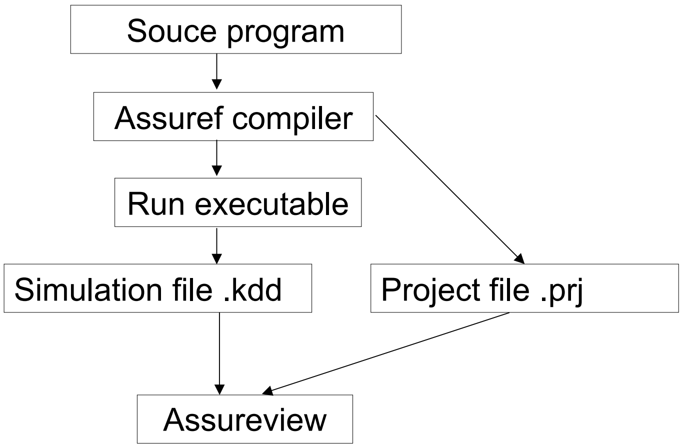
```
./myprog
```

- Visualize

```
assureview
```

Assure

The Assure process

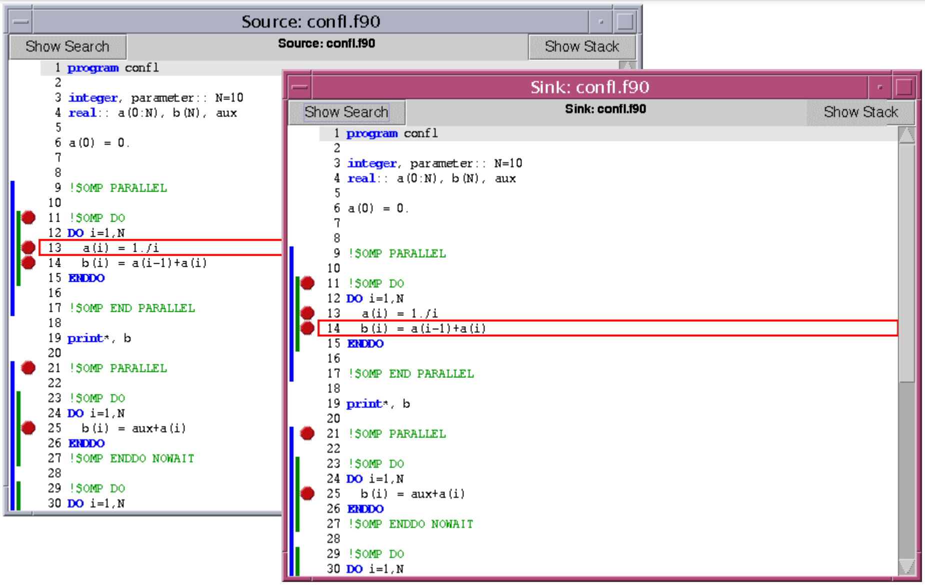


What can be detected: Conflicts

```
real:: a(0:N), b(N)
a(0) = 0.
!$OMP PARALLEL
!$OMP DO
DO i=1,N
   a(i) = 1./i
   b(i) = a(i-1)+a(i)
ENDDO
```

Whats wrong ??

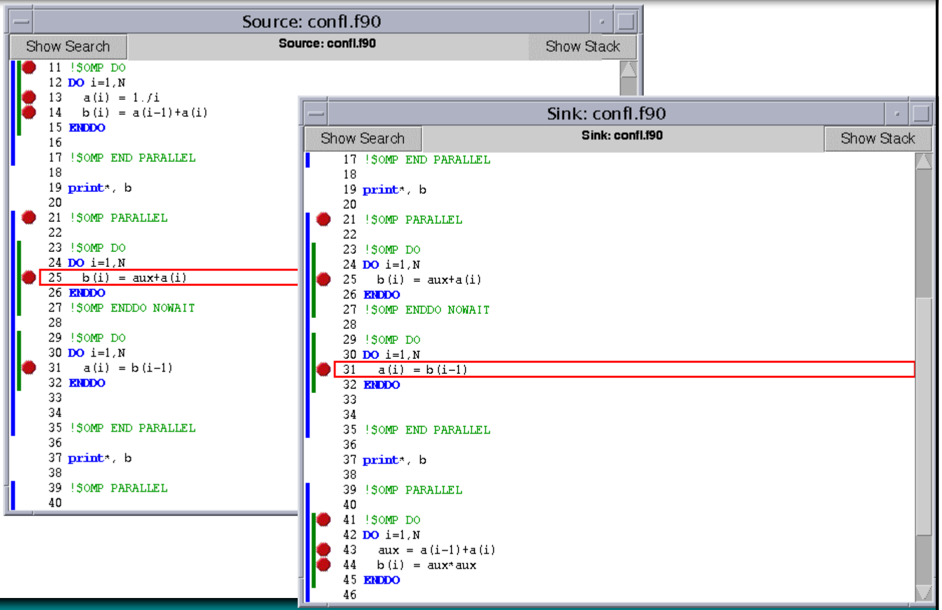
What can be detected: Conflicts



```
program confl

integer, parameter:: N=10
real:: a(0:N), b(N), aux

a(0) = 0.


!$OMP PARALLEL

!$OMP DO
DO i=1,N
  a(i) = 1./i
  b(i) = a(i-1)+a(i)
ENDDO

!$OMP END PARALLEL

print*, b

!$OMP PARALLEL

!$OMP DO
DO i=1,N
  b(i) = aux+a(i)
ENDDO
!$OMP ENDDO NOWAIT

!$OMP DO
DO i=1,N
```

```
program confl

integer, parameter:: N=10
real:: a(0:N), b(N), aux

a(0) = 0.


!$OMP PARALLEL

!$OMP DO
DO i=1,N
  a(i) = 1./i
  b(i) = a(i-1)+a(i)
ENDDO

!$OMP END PARALLEL

print*, b

!$OMP PARALLEL

!$OMP DO
DO i=1,N
  b(i) = aux+a(i)
ENDDO
!$OMP ENDDO NOWAIT

!$OMP DO
DO i=1,N
```

What can be detected: Conflicts



```
!$OMP DO
DO i=1,N
  a(i) = 1./i
  b(i) = a(i-1)+a(i)
ENDDO

!$OMP END PARALLEL

print*, b

!$OMP PARALLEL

!$OMP DO
DO i=1,N
  b(i) = aux+a(i)
ENDDO
!$OMP ENDDO NOWAIT

!$OMP DO
DO i=1,N
  a(i) = b(i-1)
ENDDO


!$OMP END PARALLEL

print*, b

!$OMP PARALLEL

```

```
!$OMP END PARALLEL

print*, b

!$OMP PARALLEL

!$OMP DO
DO i=1,N
  b(i) = aux+a(i)
ENDDO
!$OMP ENDDO NOWAIT

!$OMP DO
DO i=1,N
  a(i) = b(i-1)
ENDDO


!$OMP END PARALLEL

print*, b

!$OMP PARALLEL

!$OMP DO
DO i=1,N
  aux = a(i-1)+a(i)
  b(i) = aux*aux
ENDDO

```

Assure: Error Types

Write-Read conflicts

```
!$OMP PARALLEL DO
DO i=1,N
      a = b+c(i)
      d(i) = a+e(i)
```

The 2 statements inside the loop have to be executed in that (Write-Read) order, which is not guaranteed in a multithreaded run (a is shared by default).

Repair: private(a)

Assure: Error Types

Read-Write conflicts

```
!$OMP PARALLEL DO
DO i=1,N
      d(i) = a+e(i)
      a = b+c(i)
```

Repair: private(a)

Assure: Error Types

Write-Write conflicts

```
!$OMP PARALLEL DO
DO i=1,N
      a = b+c(i)
```

Repair: private(a)

Assure: Error Types

Private symbol, used outside loop

```
!$OMP parallel do private(a)
DO i=1,N
      a=c(i)
      d(i) = a*a
ENDDO
PRINT*, a
```

Repair: lastprivate(a)

Assure: Error Types

Uninitialized private

```
firstiter = .TRUE.

!$OMP parallel do private(firstiter)
DO i=1,N
  IF( firstiter ) THEN ...
ENDDO
```

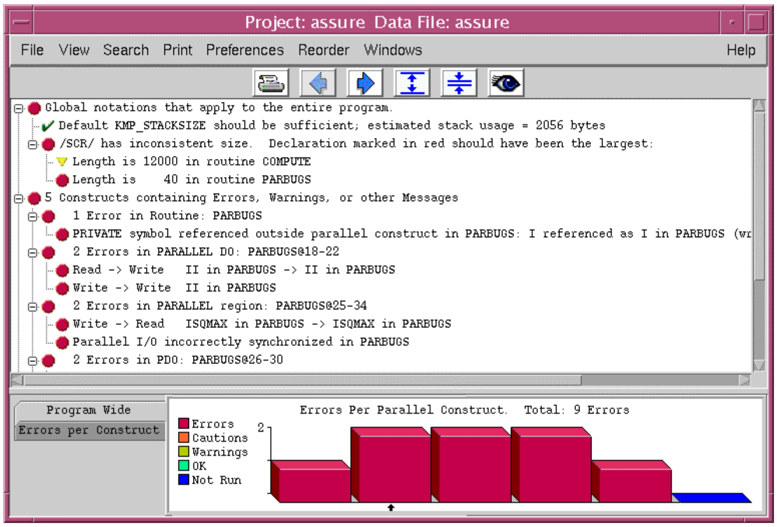
Repair: firstprivate(firstiter)

Assureview: Displays

Main

- Main error list
  - Clickable button for each error
    Click to get precise diagnostics
  - Overview chart showing statistics of bugs, different severities
- Call Tree

Assureview: Displays


Project: assure Data File: assure
File View Search Print Preferences Reorder Windows Help
Program Wide
Errors per Construct

```
⊟ ● Global notations that apply to the entire program.
   ✓ Default KMP_STACKSIZE should be sufficient; estimated stack usage = 2056 bytes
  ⊟ ● /SCR/ has inconsistent size.  Declaration marked in red should have been the largest:
      ▼ Length is 12000 in routine COMPUTE
      ● Length is    40 in routine PARBUGS
⊟ ● 5 Constructs containing Errors, Warnings, or other Messages
  ⊟ ●  1 Error in Routine: PARBUGS
      ● PRIVATE symbol referenced outside parallel construct in PARBUGS: I referenced as I in PARBUGS (wr
  ⊟ ●  2 Errors in PARALLEL DO: PARBUGS@18-22
      ● Read -> Write   II in PARBUGS -> II in PARBUGS
      ● Write -> Write   II in PARBUGS
  ⊟ ●  2 Errors in PARALLEL region: PARBUGS@25-34
      ● Write -> Read   ISQMAX in PARBUGS -> ISQMAX in PARBUGS
      ● Parallel I/O incorrectly synchronized in PARBUGS
  ⊟ ●  2 Errors in PDO: PARBUGS@26-30
```

Errors Per Parallel Construct. Total: 9 Errors

- 🟥 Errors
- 🟧 Cautions
- 🟨 Warnings
- 🟩 OK
- 🟦 Not Run

Assureview: Displays

Reading the diagnostics

Click the "+" buttons to get into the diagnostics

Finally the code sections are shown containing the error locations, (source and sink), both clearly marked.

AssureView: Displays

Inside code windows

- Show Search: normal string search menu
- Show Stack: show the calling sequence for arriving at the location.

Assureview: Displays

Other buttons

View

Select display of the error list

Search

Normal search menu, inside error list

Print

Self explaining

Assureview: Displays

Preferences

Miscellaneous settings. In particular:
source code locations ("finding files")

Reorder

.. error list by different criteria

Thanks for your attention!

pallas

Pallas GmbH
Hermülheimer Straße 10
D-50321 Brühl, Germany

info@pallas.de
http://www.pallas.com