

OpenMP Runtime Error Detection with ARCHER

At the 23rd VI-HPS Tuning Workshop

Joachim Protze, Simone Atzeni

RWTH Aachen University, University of Utah

Dong H. Ahn, Ignacio Laguna, Greg Lee, Chris Earl, Martin Schulz

LLNL

July 2016



+

Data race example in OpenMP

```
static double farg1,farg2;  
#define FMAX(a,b) (farg1=(a),farg2=(b),farg1>farg2?farg1:farg2)
```

```
1619: #pragma omp parallel for shared(bar, foo, THRESH)  
1620: for (x=0; x<1000; x++)  
1621:     T = FMAX(0.1111*foo*bar[x],THRESH);
```

Tool flags a write-write race in line 1621

Threaded Defects



Definition: data race

Data race

- Two threads access the same shared variable
 - at least one thread modifies the variable
 - the accesses are concurrent, i.e. unsynchronized
- Leads to non-deterministic behavior
- Hard to find with traditional debugging tools

Archer

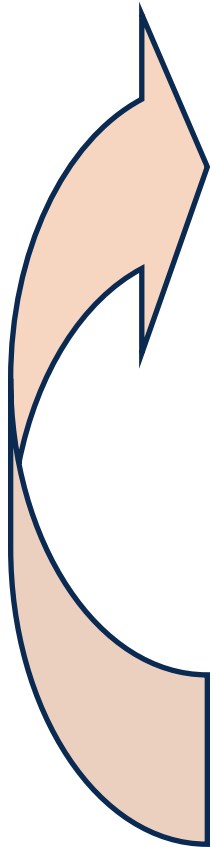
- Error checking tool for
 - Memory errors
 - **Threading errors**
(OpenMP, Pthreads)
- Based on ThreadSanitizer (runtime check)
- Available for Linux, Windows and Mac
- Supports C, C++ (Fortran in work)
- Modified OpenMP runtime improved for data race detection
- More info: <https://github.com/PRUNER/archer>



Archer – Background

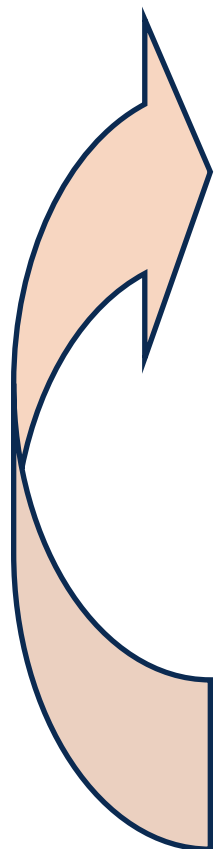
- Static Analysis
 - Only for OpenMP programs
 - Exclude race free regions and sequential code from runtime analysis to reduce overhead
- Runtime check
 - Error detection only in software branches that are executed
- Low runtime overhead
 - Roughly 2x - 20x
 - Detect races in large OpenMP applications
 - No false positives
- Compiler instrumentation
 - Slower compilation process (apply different passes on the source code to identify race free regions of code, instruments only the rest)

Archer – Usage on LC systems



- Load package:
 - use clang-3.8.1
- Compile the program with the `-g` compiler flag
 - `clang -g -fsanitize=thread -fopenmp myprog.c -o myprog`
- Run the program under control of ARCHER Runtime
 - `export OMP_NUM_THREADS=...`
`./myprog`
 - Detects problems only in software branches that are executed
- Understand and correct the threading errors detected
- Edit the source code
- Repeat until no errors reported

Archer – Usage with static analysis (not yet on LC systems)



- Compile the program with the Archer compiler wrapper
 - `clang-archer myprog.c -o myprog`
- Run the program under control of ARCHER Runtime
 - `export OMP_NUM_THREADS=...`
`./myprog`
 - Detects problems only in software branches that are executed
- Understand and correct the threading errors detected
- Edit the source code
- Repeat until no errors reported

Archer – Result Summary

```
1  #include <stdio.h>
2
3  int main(int argc, char **argv) {
4      int a = 0;
5      #pragma omp parallel
6      {
7          if (a < 100) {
8              #pragma omp critical
9              a++;
10         }
11     }
12 }
```

WARNING: ThreadSanitizer: data race

Read of size 4 at 0x7fffffffddcd by thread T2:

- #0 .omp_outlined. race.c:7
(race+0x0000004a6dce)
- #1 __kmp_invoke_microtask <null>
(libomp_tsan.so)

Previous write of size 4 at 0x7fffffffddcd by main thread:

- #0 .omp_outlined. race.c:9
(race+0x0000004a6e2c)
- #1 __kmp_invoke_microtask <null>
(libomp_tsan.so)

Hands-on

```
$ use clang-3.8.1
```

```
$ cp -r /g/g90/protze1/archer-examples archer-examples
```

```
$ cd archer-examples
```

```
$ clang -fopenmp -g prime_omp.c -lm
```

Try:

```
$ OMP_NUM_THREADS=2 ./a.out
```

```
$ OMP_NUM_THREADS=4 ./a.out
```

```
$ OMP_NUM_THREADS=8 ./a.out
```

Hands-on 2

- Now compile with data race detection:

```
$ clang -fopenmp -fsanitize=thread -g prime_omp.c -lm
```

```
$ OMP_NUM_THREADS=2 ./a.out
```

Fix the issues, recompile, test again

Better: do this using the batch system

Conclusions

- Races:
 - Often hard to detect, in many cases only visible from time to time
 - Races manifesting only at large scale are often detectable by ARCHER at small scale
 - Fortran support is on the way
- Use tools to detect defects as early as possible:
 - During development + unit testing
 - Development of ARCHER is ongoing effort, also porting to more architectures (power8) and OpenMP runtimes (like lomp/gomp).
- More information:
<https://lc.llnl.gov/confluence/display/LCTOOLS/ThreadSanitizer+with+Archer>

Thank You