

ARCHER: Effectively Spotting Data Races in Large OpenMP Applications

Ignacio Laguna, D. H. Ahn, G. L. Lee, M. Schulz (LLNL)
S. Atzeni, G. Gopalakrishnan, Z. Rakamarić (Univ. of Utah)
J. Protze (RWTH Aachen)

March 17-18, 2016



OpenMP is Widely Used in HPC Applications

The OpenMP logo is displayed in a white box with a teal border. It features the text "OpenMP" in a teal, sans-serif font, with a registered trademark symbol (®) to the right. The background of the slide is a photograph of a server room with rows of server racks and orange cables.

```
void simple(int n, float *a, float *b)
{
    int i;
    #pragma omp parallel for
    for (i=1; i<n; i++)
        b[i] = (a[i] + a[i-1]) / 2.0;
}
```

- Standard to express parallelism in multi-threaded code
- Parallel code can be executed in CPUs or accelerators (e.g., GPUs)
- As in all multi-threaded programming models, data races can occur

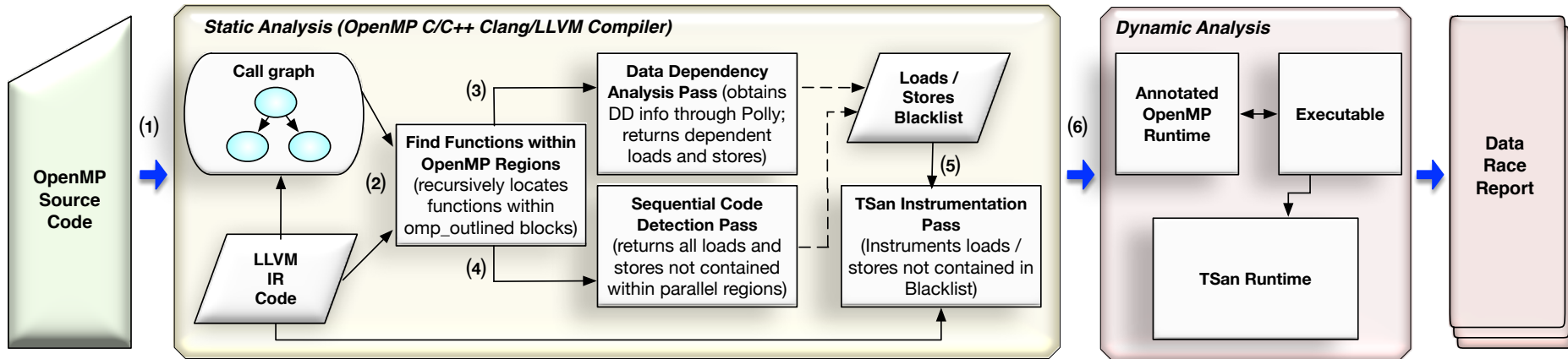
Gap in Data Race Detection Tools for HPC

- Identifying data races in large OpenMP applications is challenging
 - Scalability is key
 - Accurate and precise detection is very important
 - Low overhead and portability allow adoption in practice

Tool	Technology	Accuracy/Precision		Overhead	Portability
		Pthread	OpenMP		
Intel SSA	Static	●	●	●	●
Intel Inspector	Dynamic	●	●	●	●
Helgrind	Dynamic	●	●	●	●
ThreadSanitizer	Dynamic	●	●	●	●

Archer Data Race Detector

Accurately detects OpenMP data races with low overhead



- Combines static and dynamic techniques in a single tool
- Build on top of:
 - Polly for static dependence analysis
 - ThreadSanitizer – dynamic happens-before analysis
 - LLVM/Clang OpenMP runtime

Approach – Static Analysis Phase

Target instrumentation on a simple OpenMP program

```
1 main() {
2  // Serial code
3  setup();
4  sort();
5
6  #pragma omp parallel for
7  for(int i = 0; i < N; ++i) {
8      a[i] = a[i] + 1;
9  }
10
11 #pragma omp parallel for
12 for(int i = 0; i < N - 1; ++i) {
13     a[i] = a[i + 1];
14 }
15
16 #pragma omp parallel
17 {
18     sort();
19 }
20
21 // Serial code
22 printResults();
23 }
```

Serial code blacklisted

Used in serial and parallel code

No data dependency code blacklisted

Potentially racy code instrumented

Potentially racy code instrumented

Serial code blacklisted

Evaluation: OmpSCR Benchmarks

Application	Inspector - Default	Inspector - ExtremeScope	Inspector - MaxResources	Archer (no SA)	Archer
Slowdown Mean	29.5	30.3	122.4	19.6	18.4
Geometric Mean	18.3	20.2	71.5	10	8.8
Detected Races	6	6	9	12	12
False Alarms	2	3	2	0	0

The total number of races in the benchmarks is 12.

