Efficient Parallel Runtime Bounds Checking with the TAU Performance System

John C. Linford, Sameer Shende, Allen D. Malony
ParaTools, Inc., Eugene, OR

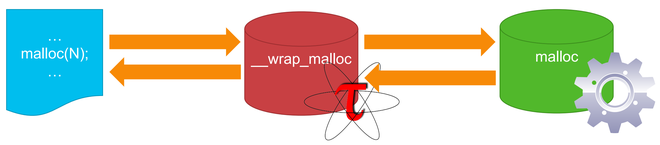
Andrew Wissink
U.S. Army Aviation Development Directorate - AFDD
NASA Ames Research Center, Moffett Field, CA

Problems Addressed

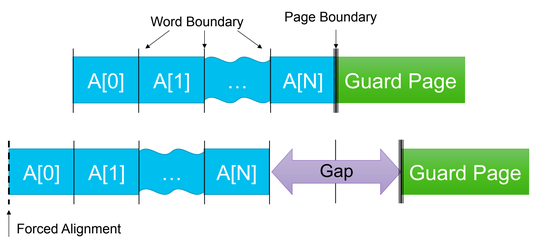
- **Debug memory errors in parallel codes written in multiple languages.**
  - When and where do memory errors occur?
  - What is the heap memory usage?
  - Was the error cause by a read or a write?
  - Which processes or threads experienced the error?
  - Were there any memory leaks in the application?
  - What were the performance characteristics prior to fault?
- **Protect sensitive data.**
  - Generate reports free of application data.
  - Open and portable debugging data file format.

Approach

1. **Instrument heap memory allocation and deallocation calls.**
   - Record heap memory use, allocation and deallocation rate.
   - Associate every allocation with its use in the source code.

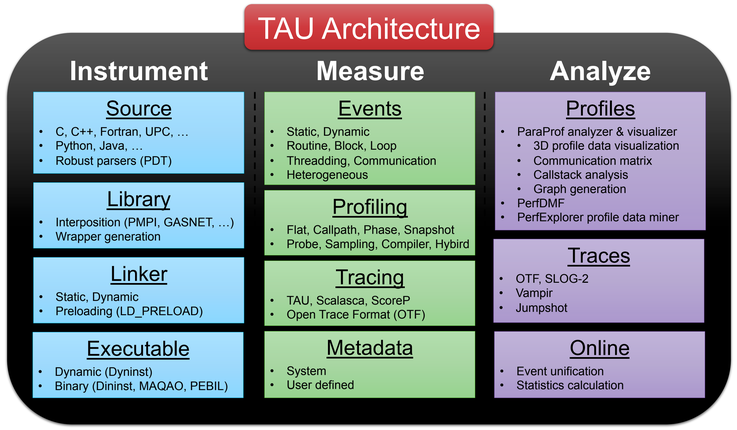


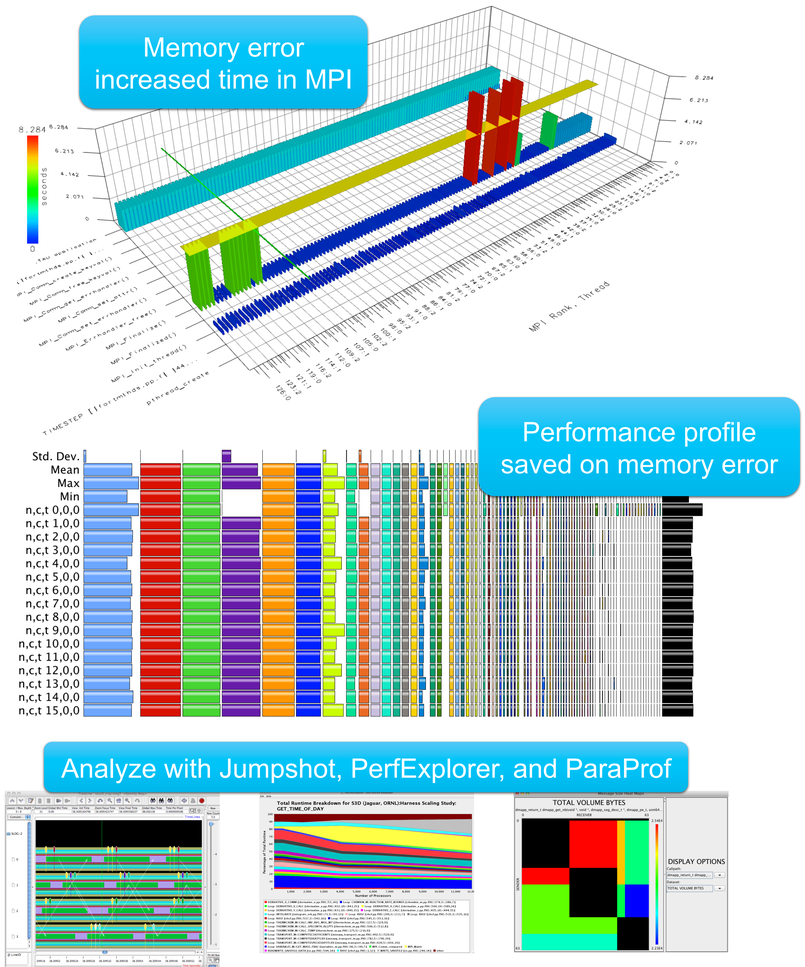
2. **Allocate guard pages before and/or after memory allocations.**
   - Use the MMU hardware to protect regions of memory from access.
   - Fast invalid memory access detection but high memory overhead.



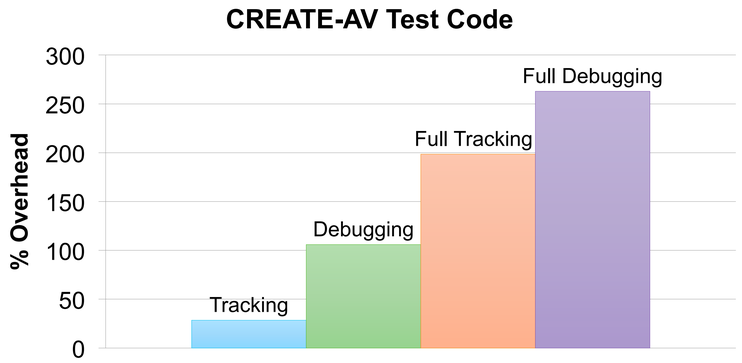
3. **Intercept error signal if error occurs.**
   - Save backtrace in the application profile.
   - Write application profile to disk and gracefully shut down.
4. **Send application profile to developers for analysis.**
   - Profile contains no application input or memory content information.

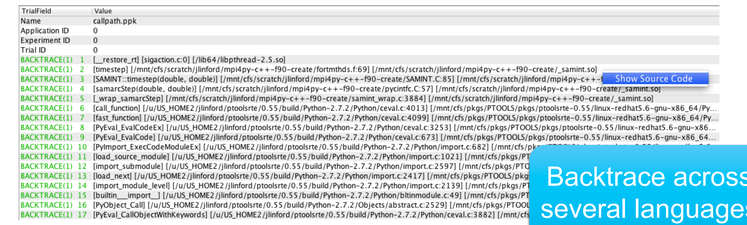
The TAU Performance System

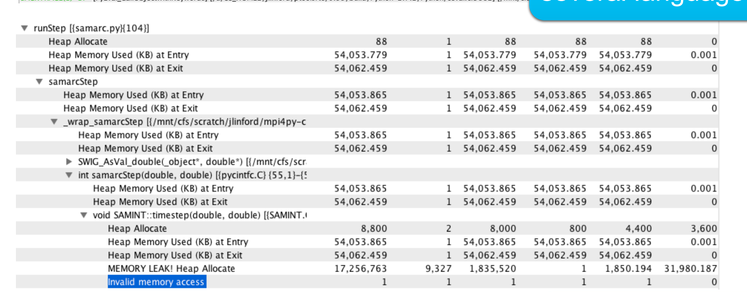




Results







Related Work

- H. Aygün, D.U.M.A – Detect Unintended Memory Access, http://duma.sourceforge.net/, May 2013.
- Apple Corp., “Guard Malloc Manual Page,” https://developer.apple.com/, March 2009.
- B. Perens, “efence – Electric Fence Malloc Debugger,” http://linux.die.net/man/3/efence, 1999.
- J. Seward and N. Nethercote, “Using Valgrind to detect unintended value errors with bit-precision,” in *USENIX ATC*, Anaheim, CA, USA, April 2005.
- K. Serebryany, D. Bruening, A. Potapenko, and D. Vyukov, “Address-sanitizer: A fast address sanity checker,” in *USENIX ATC*, 2012.
- Microsoft Corp., “Gflags and PageHeap,” http://msdn.microsoft.com/, July 2013.
- S. Shende, A. D. Malony, J. Linford, A. Wissink, and S. Adamec, “Isolating Runtime Faults with Callstack Debugging using TAU,” in *Proceedings of the IEEE HPEC 2012 Conference*. IEEE Computer Society, 2012.