

Software Development Tools for Petascale Computing Workshop

Correctness Tools Working Group Report

2007/8/2

Co-chairs: Susan Coghlan & Curtis Janssen

Process

- Identified subareas and scope for WG charter.
- Discussed issues and identified gaps within each subarea.
- Individually rated each gap's priority and difficulty
 - 70 points could be distributed among gaps
 - Each area could be given high, medium, or low difficulty
- Averages were taken for each gap
 - Difficulties that were not assessed by an individual did not count towards the average. Priorities that were not assigned were included as zero.
- Gaps in blue were added after our session met. Not all participants had the chance to prioritize these.

Umbrella Issues

Gap	Priority	Difficulty
Open source support model needs to be elucidated, as well as the role of vendors.	2	Medium
Common open infrastructure with performance expectations that are documented is missing.	3	Med-High
Operating system support (overriding issue for pretty much everything) remains an issue for many tools.	2	Med-High
Ease of use often missing.	2	Med-High
Education lacking.	3	Med-Low
Impossible to develop tools for scale without access.	0	Medium

Traditional Debuggers (eg. TotalView, DDT)

Gap	Priority	Difficulty
Scalability (traditional debuggers being able to do the same things on more nodes) is too limited.	10	Med-High
Extensibility of the tools for more user-driven analysis capability is needed.	2	Med-High
New ways to represent the output of the debuggers, pre-analyzed for the users are needed.	2	Med-High
Interface between traditional debugger to light-weight debugging tools (tool collaboration interface) is needed provide to a smooth transition from super light-weight to super heavy tools.	5	Med-High

Light-weight Debugging Tools

Gap	Priority	Difficulty
Missing tools to debug core files at large scale.	4	Medium
Not enough extreme scale lightweight debugging tools	3	Med-High
Lightweight tools to point to root cause	3	High

Memory Usage Correctness

Gap	Priority	Difficulty
Memory leak/high water mark tool to instrument libraries is needed.	8	Medium

Thread correctness (itc)

Gap	Priority	Difficulty
Existing tools are neither multi-platform nor good for parallel environments.	6	High

MPI Usage Correctness (eg. Umpire)

Gap	Priority	Difficulty
Tools that run at small scale could be (but are not) used to predict issues that could occur at large scale.	2	Medium

Static Analysis Tools (eg. lint)

Gap	Priority	Difficulty
Compiler infrastructure that is needed to build the tools is missing.	2	Med-High
Many tools still needed, particularly for Fortran.	2	Med-Low

Correct Fault Handling

Gap	Priority	Difficulty
Missing RAS systems that applications and system software can use to learn about hardware causes for program faults and allow the fault to be handled.	3	Medium

Test Harness for Regression Testing, Verification

No standardized tool for regression testing exists.	1	Medium
Hardware/system software coverage test suites are not exhaustive.	2	Medium

Formal Verification (mainly used for hardware now)

Gap

Tools that use formal verification methods to identify deadlocks, livelocks, race conditions, and other errors in parallel software are missing.

Priority Difficulty

2

High

Major Issues

- Traditional debugger scalability.
- Understanding memory usage
- Thread correctness
- Filling gap between lightweight and heavyweight debuggers

Numerous other issues important, too.