bgclang: Creating an Alternative, Customizable, Toolchain for the Blue Gene/Q
Hal Finkel – Argonne National Laboratory – hfinkel@anl.gov

The Address Sanitizer project provides instrumentation-based memory-access checks. It works on non-commodity machines with realistic working sets. Because many scientific applications often contain large arrays, and speculative load calculation does not provide access to these arrays, buffer overflow errors are fairly common, often due to unattentive or formatting mistakes. Address Sanitizer is an efficient way to monitor program memory accesses and find these kinds of bugs. There are two widely used open-source tools for catching these kinds of errors: Valgrind and Address Sanitizer.

A prototype of Valgrind is currently available on the BG/Q, while an installation of Address Sanitizer on the BG/Q has contributed significantly to the PowerPC backend. However, for the sake of brevity, and to focus only on novel work well, on non-commodity machines.

The Address Sanitizer project provides instrumentation-based memory-access checks, a debugging technique essential for finding buffer-overflow errors in production-level applications with realistic working sets. Because many scientific applications often contain large arrays, and speculative load calculation does not provide access to these arrays, buffer overflow errors are fairly common, often due to unattentive or formatting mistakes. Address Sanitizer is an efficient way to monitor program memory accesses and find these kinds of bugs. There are two widely used open-source tools for catching these kinds of errors: Valgrind and Address Sanitizer.

A prototype of Valgrind is currently available on the BG/Q, while an installation of Address Sanitizer on the BG/Q has contributed significantly to the PowerPC backend. However, for the sake of brevity, and to focus only on novel work well, on non-commodity machines.

Acknowledgments

The author of this poster would like to thank Jeff Hammond, Kalyan Kumaran, Mike Papka, Katherine Riley, William Scullin, and everyone else at ALCF who has supported the development of bgclang. A great deal of credit is owed to Ramesh Srinivasan, Michael Blockens, Thomas Gooding, Michael Kincaid, and Paul Porton, Eric Schneider, and the many others who provided early and continuing testing. Finally, the author would like to thank Mike Papka and other members of the LLVM community who assisted, directly or indirectly, with this work. This research used resources of the OLCF, which is supported by DOE/SC under contract DE-AC02-06CH11357.