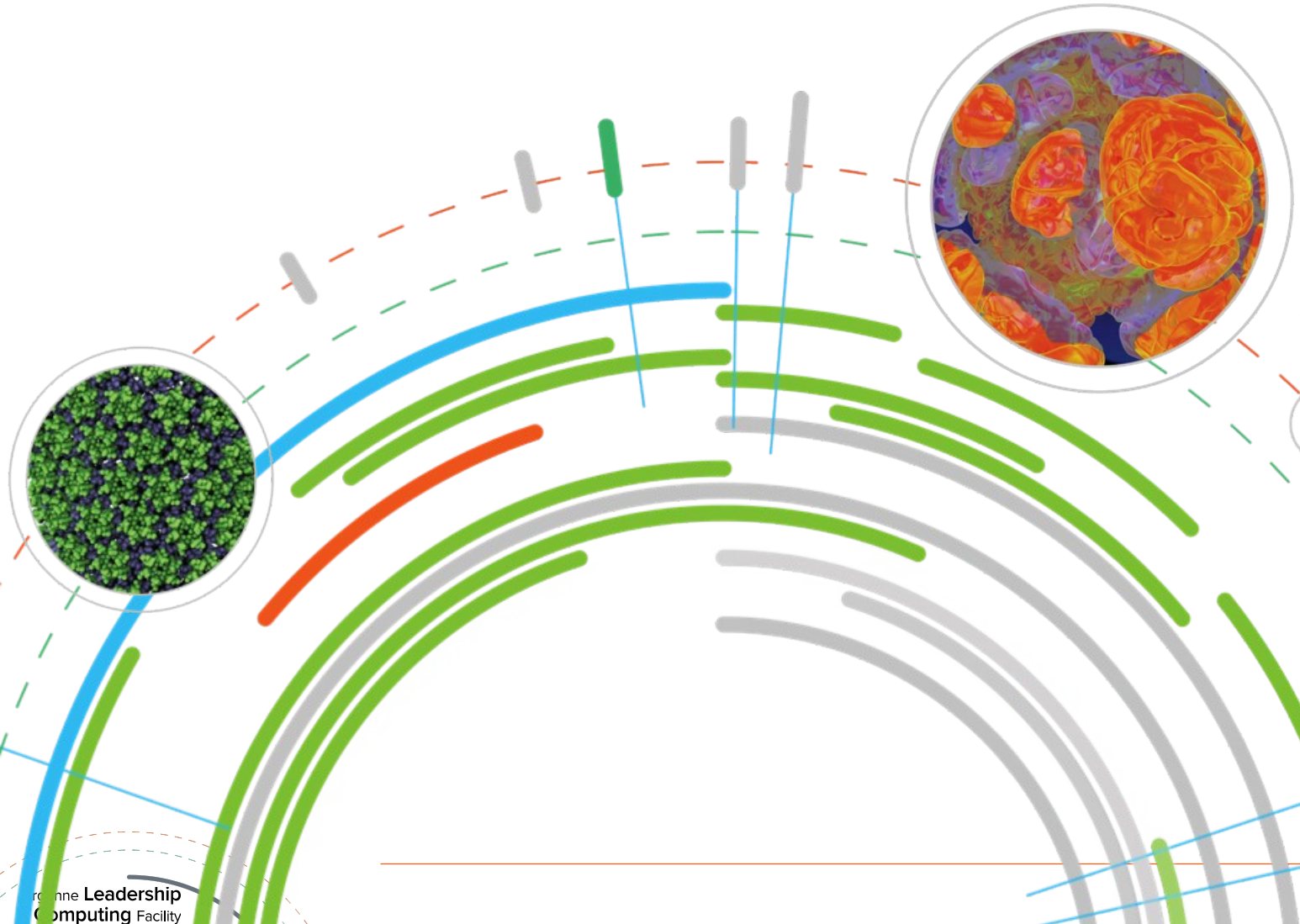
bgclang – OpenMP in LLVM/Clang for the BG/Q

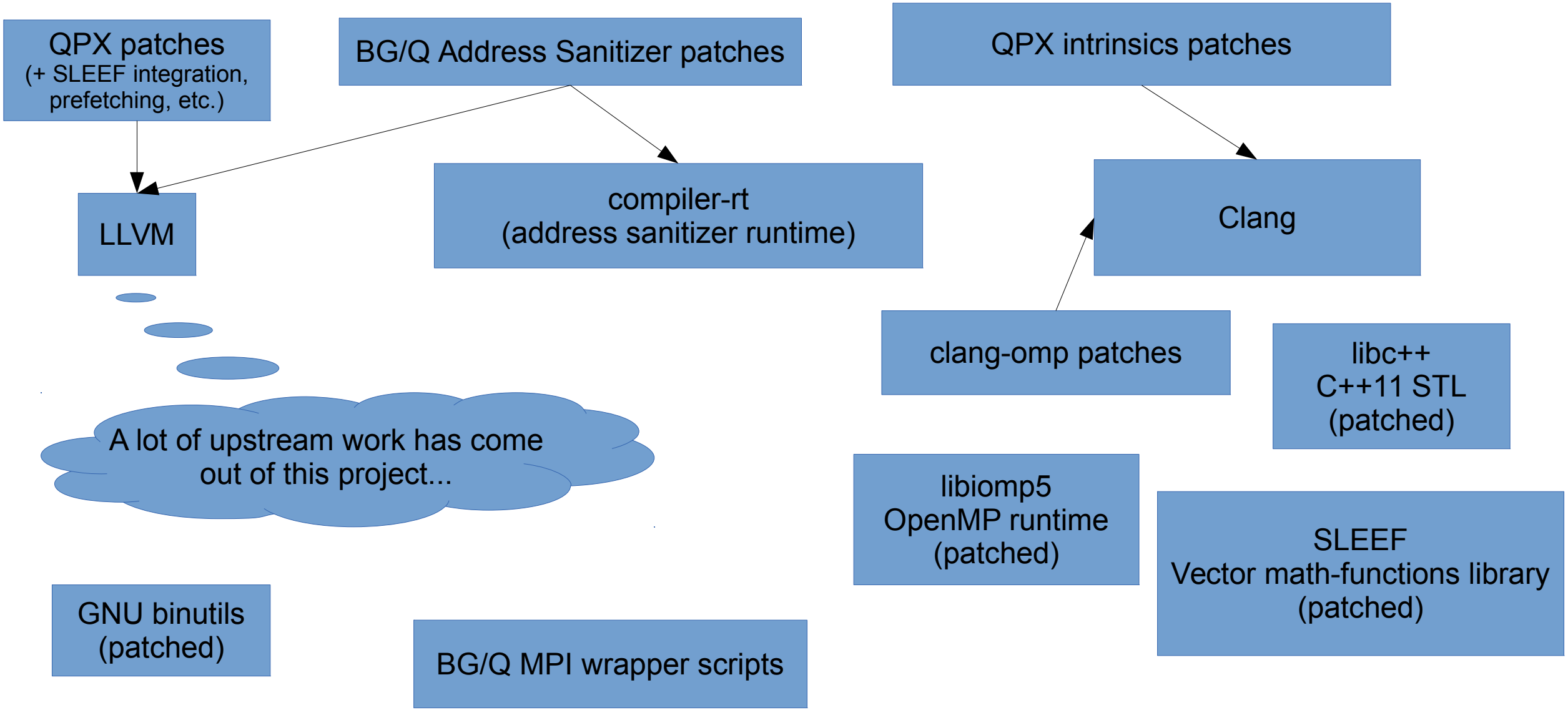


Also, see
the bgclang poster!

Argonne **Leadership**
Computing Facility

Argonne
NATIONAL LABORATORY

bgclang Components:



OpenMP

Enable OpenMP support with: -fopenmp

- Full OpenMP 3.1 support with some OpenMP 4 features
- Derived from Intel's clang-omp project and uses Intel's libiomp5 runtime library (patched).
- The runtime library has not yet been optimized for the BG/Q

```
#pragma omp simd
for (int i = 0; i < n; ++i) {
  ...
}
```

OpenMP 4 SIMD directives are included!

The Intel OpenMP runtime on PowerPC, really?

Porting Intel's runtime to PowerPC was not difficult:

- Intel already had memory barrier macros in the code, we just had to use them:

```
#if KMP_ARCH_PPC64
# define KMP_MB()       __sync_synchronize()
#endif

#ifndef KMP_MB
# define KMP_MB()       /* nothing to do */
#endif
```

- There were some little-Endian dependencies in the affinity code, but that can be disabled.
- Most of the code just uses standard POSIX interfaces, and thus was portable.
- The little bit of assembly code in the microtask dispatch was easy to replace with C code based on Clang's usage.

OpenMP in Clang?

bgclang uses the Clang OpenMP implementation developed by Intel:

- Intel develops and maintains a latest-release-based patchset for OpenMP 4 (http://clang-omp.github.io/)
- ALCF maintains the trunk-based patchset for OpenMP 4 (https://github.com/clang-omp/clang_trunk) [with an associated llvm_trunk repository]
- Intel is rewriting the OpenMP patches based on upstream code review and support is appearing in upstream Clang! ALCF, IBM and others are assisting with this process.

Does it work? Yes. (No performance tuning yet, however...)

For example, some results from the (unofficial) C OpenMP version of the NAS parallel benchmarks (2.3) class B:

(these were run with 64 threads, compiled with -O3 -fopenmp with bgclang and -O3 -qsmp=omp -qstrict with bgxlc_r)

Benchmark	bgclang (s)	bgxlc_r (s)
EP	29.96	11.63
BT	163.70	107.12
FT	32.56	34.26
LU	71.33	50.31
MG	5.19	2.32
SP	1244.55	214.78

Acknowledgments

- Jeff Hammond, Kalyan Kumaran, Mike Papka, Katherine Riley, William Scullin, and everyone else at ALCF who has supported the development of bgclang
- Ramesh Balakrishnan, Michael Blocksome, Thomas Gooding, Michael Kruse, Jack Poulson, Erik Schnetter, and the many others who provided early and continuing testing
- Chandler Carruth, Tobi Grosser, Chris Lattner, Jakob Olesen, Nadav Rotem, Bill Schmidt, Arnold Schwaighofer, Andy Trick, and all other members of the LLVM community who assisted, directly or indirectly, with this work
- ALCF is supported by DOE/SC under contract DE-AC02-06CH11357