User-level thread implementation based on Boost context library

Currently, Converse has several user-level implementations.
1. QuickThread
2. Ucontext_t based
3. uJcontext(setjmp/longjmp based)
4. pthread based
5. stack copy

Ucontext_t is the default if it is available and uJcontext is used for Mac and Quickthread is used for ARM.
Ucontext_t is deprecated and has not been maintained by the Unix and Posix standard. The behavior of this sys calls is not defined on different environments.

For example, ucontext_t can allow migration of user-level context across threads in Linux but it doesn't allow the migration on Mac. (Different implementation)
UJcontext based on setjmp/longjmp cannot be used for this kind of migration. Setjmp/longjmp is designed for signal handling and nonlocal goto. So, the context should be resumed on the thread where the context is suspended previously. Quickthread also doesn't show the same behavior on different environments as ucontext_t.

Now, Boost context is the only available user-level context for most of environments we are targeting.
1. Supported OS: WINDOWS, Mac, LINUX/UNIX, IOS
2. Supported Architecture: Details in the following link (most of architectures on each supported OS)
   (http://www.boost.org/doc/libs/1_61_0/libs/context/doc/html/context/architectures.html)

The license of the boost library is similar to BSD and MIT license, which means we don't have liability to disclose our codes using Boost.

In addition, Boost provides assembly codes(machine executable object code) for this context implementation. (Don't need to compile their codes. We can use their implementation only with these assembly codes)
(http://www.boost.org/users/license.html)

Argobot also adopts fcontext_t and using assembly codes for their user-level threads.
This assembly code doesn't include the support of Windows. This will be added in a separate patch.

#5 - 08/03/2017 12:36 AM - Seonmyeong Bak
With this library, the OpenMP integration works well on MacOSX with GCC and barrier related directives also works on Linux and Mac OS X.

Boost context doesn't guarantee thread-local variables so I turned off TLS based CvPv variable option (CMK_NOT_USE_TLS_THREAD 1)

#6 - 08/03/2017 09:28 AM - Sam White
- Target version set to 6.9.0