Submit New Art

Intel ArBB FAQs

Frequently Asked Questions about Intel® Array Building Blocks (Intel® ArBB):

Q1: What are the benefits of Intel ArBB?
A1: Intel ArBB's data-parallel capabilities provide an array of benefits to developers:
o Forward-scaling: Intel ArBB lets applications span multi-core and many-core processors without requiring developers to rewrite programs over and over. The benefits of only writing and debugging code once are substa
o Safety: Intel ArBB helps prevent parallel programming bugs such as data races and deadlocks. Intel ArBB gu against these problems enabling developers to code their algorithms at a high level and abstracting details pertai to low-level optimization. Intel ArBB optimizes the application across all available cores, while preventing the developer from introducing parallelism errors (such as data races and deadlocks).
o Ease-of-use: Intel ArBB extends C++ and is compatible with standard C++ compilers.

Q2: Will I have to learn a new language?
A2: No. Intel ArBB is not a new language. Intel ArBB is a programming model that introduces implicitly paralle operators on new aggregate data types. Intel ArBB is implemented in standard C++, and is backed by a runtime library which generates code that simultaneously takes advantage of both SIMD and threaded parallelism.

Q3: If I use Intel ArBB, am I limited to C or C++ to develop my application?
A3: Intel ArBB support for C++ is our first priority after which we plan to make Intel ArBB available for other languages.

Q4: Will I have to rewrite my whole application to use ArBB?
A4: Developers target only the computationally intensive kernels in their application with Intel ArBB.

Q5: Will I need to rewrite portions of my application for each target platform?
A5: No. Intel ArBB provides forward scaling via its dynamic, platform-aware runtime allowing the application run on a variety of machines with different core counts, different cache sizes, different memory models, and ev different SIMD widths in a heterogeneous environment.

Q6: How can I ensure that my application thread pool and the Intel ArBB thread pool are not competin resources?
A6: Developers can specify the maximum number of threads that Intel ArBB uses. Unlike other parallelism solutions, Intel ArBB dynamically adapts to changes in the availability of underlying thread resources. Intel Ar uses the Intel TBB runtime as its underlying threading runtime, ensuring Intel TBB and Intel Cilk Plus interoperability. The Intel TBB scheduler ensures that resources are not oversubscribed.

Q7: Will Intel ArBB have Mac OS X support?
A7: We are not disclosing future plans for Mac OS at this time.

Q8: Will Intel ArBB be open-sourced like Intel® Threading Building Blocks (Intel® TBB) ?
A8: We are exploring open source options but have no specific plans to announce at this time. We welcome

customer input and are committed to providing an open solution in some form.

Q9: How does Intel ArBB compare with OpenCL?
A9: Intel ArBB provides the same cross-architecture advantages as OpenCL while also providing more abstract programming constructs than OpenCL does and the additional scalability and application behavior benefits of dynamic compilation.

Q10: Will Intel ArBB support the Intel Many Integrated Core (Intel MIC) architecture?
A10: A future release of Intel ArBB will support the Intel MIC architecture. The first release of our Intel ArBB product in 2011 will be focused on multicore architectures.

Do you need more help?

Click tags links for related articles
Search Knowledge Base
Visit User Forums
Get other Support options

This article applies to: Intel® Array Building Blocks Knowledge Base