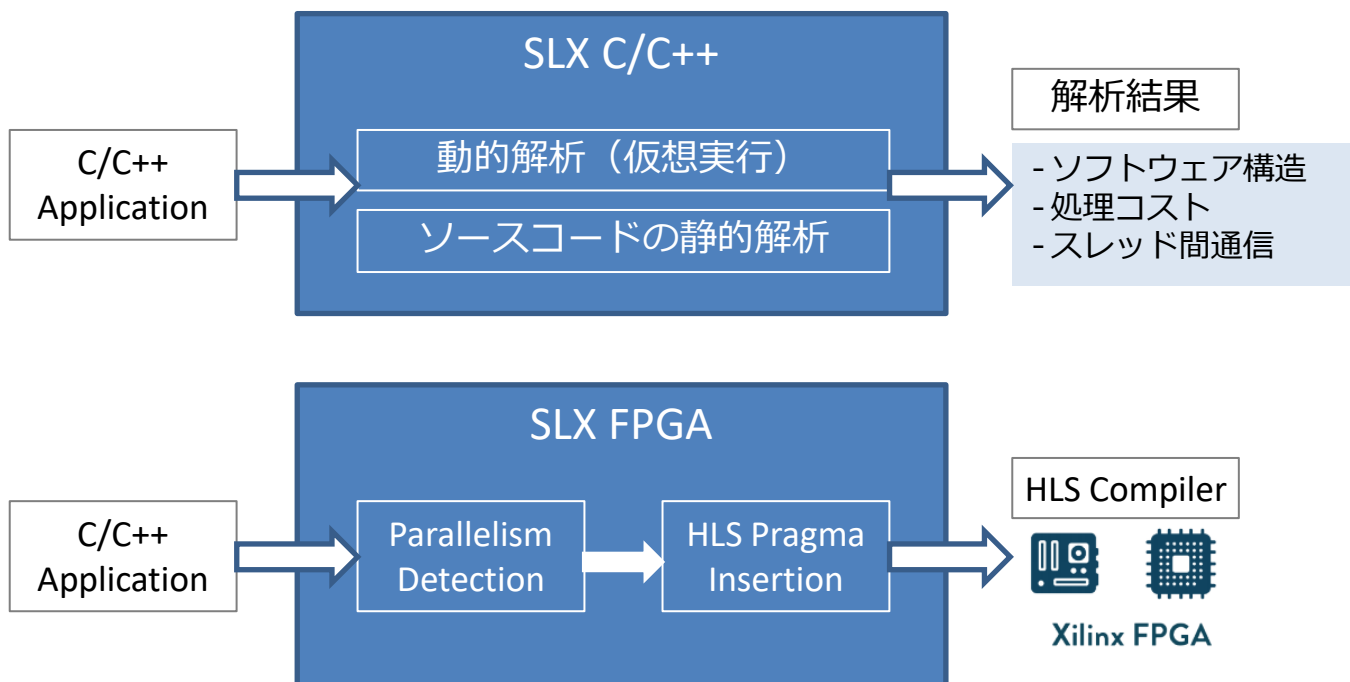


静的解析、動的解析を結合した独自のコード解析技術によって、
マルチコアC/C++アプリケーションを解析／最適化



Static, Dynamic and Semantic
Source Code Analysis



Actionable Insights for Improved
Software Performance



Data Protection and
Dependency Analysis



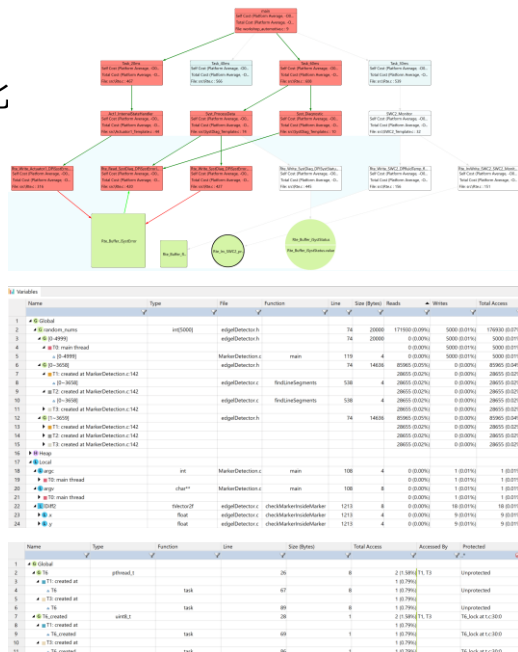
Parallelism Detection

特長・仕様

- SLX C/C++
 - ・ヘテロジニアス・マルチコアシステムにおける性能
 - ・性能向上のために並列化が可能な部分の探索
 - ・共有メモリを介したスレッド間通信を可視化
- SLX FPGA
 - ・C/C++コードの並列化の可能性を探索し、ハードウェア化が有効的な部分を抽出
 - ・HLS Pragmaの自動挿入
 - ・Xilinx Vivado HLS およびVitis 開発環境に対応
- 推奨動作環境 : Ubuntu 64bit / Windows10 64bit
- ライセンス : 年間タイムベースライセンス契約 (ノードロックおよびフローティングライセンス)

SLX C/C++

1. マルチコア・アプリケーションの解析
 - 複雑なマルチコア・アプリケーションの構造を可視化
 - CPU負荷・メモリアクセス量を定量的に示します。
2. 通信・メモリボトルネックの特定
 - 独自のコード解析技術により、変数と関数のアクセス関係、スレッド間の通信を解析。
 - ポインタ変数を介したアクセスについても、参照先の実体を解析できます。
3. スレッド間・プロセス間の共有メモリロックの追跡
 - 共有メモリのロックのメカニズムを追跡し、ロックが漏れており競合の可能性がある箇所を特定します。



SLX FPGA

1. 合成できないソースコードのリファクタリング
 - ソースコードを論理合成可能に修正するためのガイダンスを提示。
 - 合成できない標準Cライブラリの置き換え。
2. HWとSWの分割
 - ソースコードの並列実行可能性を探索し、HW化が有効な箇所を抽出。
 - HW化した際の性能向上の見積もりを提供します。
3. HLSプラグマの挿入
 - HW化する箇所を決定した後、SLX FPGAが自動的に、HLSプラグマをソースコードに挿入します。

