

Software Decomposition Ankit Jain, Ravi Shankar

Technology Transfer:

Multicore Architectures cannot be fully and effectively utilized with sequential model based software. Software cannot be re-written entirely in order to utilize next generation multicore architectures. We have developed a 10-Step methodology which effectively reverse engineers existing (Legacy) software to convert it into concurrent model based software. The aim is to solve embedded software and real-time concurrency issues while partitioning code over multiple concurrent architectures. We utilize top-down representation, bottom-up annotation, and middle out analysis to effectively implement this methodology. We perform performance annotation by analyzing computation, communication and concurrency cost. This addresses three technology transfer issues: Software Decomposition, Concurrency Modeling, and Performance Annotation.

Demo:

We implement 10-Step Methodology as follows:

1. Analyze Software Code Bottom-Up for Computation and Communication Analysis as per the Set of Rules developed.
2. We use Klocwork inSpect to perform complexity analysis using coupling cohesion metrics. Loops in the code are unrolled for accurate use-case based static analysis of the software.
3. Control and Data Flow Structure analysis performed manually without aid of any tool.
4. Klocwork inSight is used graphically analyze relationship between modules in the software architecture. Modules of interest determined in step 1-2 are analyzed here.
5. We group tightly coupled modules which execute sequentially into sub-modular systems.
6. We decompose high cost modules to exploit naïve parallelism into multiple sub-modular systems. [Step 5 & 6 can generate multiple options].
7. We use LTSA tool to model and test concurrent multi-core architecture.
8. Develop multi-core (4-Core) architecture simulation model (as per decomposition option) using MLDesigner using discrete event model of computation.
9. Estimate concurrency cost by mapping decomposed software onto multi-core architecture. Multi-core
10. Decomposition option and multi-core architecture solution is chosen based on Quality-of-Service requirements.