並列分散処理システムの研究

松下, 智

（様式 2）

氏　　名　：松下　智

論 文 名　：並列分散処理システムの研究

区　　分　：甲

論　文　内　容　の　要　旨

集積回路でのトランジスタ集積度が18ヶ月で2倍になるというムーアの法則は，50年以上経過した2022年においても継続している．2022年に入り，単なる微細化だけではなく，Flash Memory 等での三次元集積や，デスクトップCPU でのChip-let 等を用いたパッケージ内混載など手法も多様化しているが，依然として集積度向上のペースは衰えていない．1990年台には，すでに今日のようなシステムLSI であるSoC (System on Chip) 等の時代の到来が予想され「設計生産性危機」が叫ばれていた．

一方，ソフトウェアに関しては，1972年にはDijkstra らが「ソフトウェア生産性危機」を訴え，Algol やPL/I 等の言語とともに構造化プログラミングを提案した．以降，解決に向けて広く研究が行われてきた．ハードウェア性能の向上や生産・消費されるデータ量の爆発的増大もあり，引き続き大きな問題となっている．

本論文の第一の貢献は，並列プログラムの生産性向上である．まず，既存の高級言語向けの並列ライブラリにより並列プリミティブを実装し，実用プログラムを並列化した．ついで，自動並列化で短TAT (Turn Around Time) でかつ高い並列度抽出を達成した．ここでは，まず，CAD 向けの専用計算機“Cenju” に対して，並列ライブラリ，並列OS，クロスコンパイル環境を整備し汎用化した．実アプリケーションの並列化で有効性を実証し，ベクトル型スーパーコンピュータSX-2 で6～7 倍しか加速しないMHD型プラズマシミュレータを並列化し，64PE (Processing Element)で44 倍の加速を得た．次世代機“Cenju2” をシステム設計し上位互換と性能向上（単体PE 性能で7倍，最大PE 台数で4倍）を両立するシステム設計を行った．ついで，オンチップ・マルチコア型SoC を用いた自動並列化を研究した．FOPE （Fork Once Parallel Execution）実行方式による投機マルチスレッド方式の実装法を検討し，4コア(各2 issue in-order のスーパースカラ・コア）のLSI “Merlot” を試作した．Merlot の実機にてMPEG2 デコードのカーネルIDCT を自動並列してIPC (命令/clock)=1.65 を，さらに人手によるソースコードチューニングを加えてIPC=2.72, 340MIPS を実現した．FOPE 型の次世代投機マルチスレッド自動並列として“Pinot” を提案した．Pinot では，FOPE 実行をさらに改善した実行方式を採用するとともに，バイナリ(実行オブジェクト）変換による完全自動並列化を実装した．jpeg decode のコード全体にて，4コア(各5 stage single issue）で1コアの3.2 倍の速度向上を確認した．自動変換時間は3分であり，人手で行った場合の数ヶ月に対して数万分の1 に短TAT化され，自動並列化の有効性を確認した．

本論文の第二の貢献は，分散処理系構築のためのフレームワークの提案である．分散処理では経済性，拡張性，信頼性等の利点が得られるが，システム作成やデバッグは容易ではなくプラットフォームや応用も少ない．また，通信によるオーバヘッド，プライバシ漏洩，障害の発生も懸念される．分散システム構築を容易にし，かつ利点を伸ばし，欠点を補うフレームワークを提案する．耐障害性のある分散型短遅延インメモリKey Value Store（RAMCloud）に，RIFL（Reusable Infrastructure for Linearizability）とよぶ厳密一度（Exactly-Once）実行のRPC を実装し13.5μs のレイテンシを実測した．これは，RAMCloud 従来の堅牢な書き込みに対して，0.5μs の追加レイテンシに過ぎない．この上にデータベーストランザクションを実装することで，従来のH-Store の1/10 未満のトランザクション・コミット遅延を確認した．ついで，IoT（Internet of Things）において，エッジ内に閉じて処理するIn-Edge 処理の障害検出・復旧を実現するフレームワークを提案した．これを，電灯制御タイマに適用し，性能，再利用性，障害対応を実証した．さらに，MQTT を用いた従来の実装と，性能・実装性を比較した．ついで，スマートホームIoT における単一障害点を分析・優先度付けし，最重要なインターネット接続障害に対して，対策法を整理し実装評価した．

本論文の第三の貢献は，LSI 設計（EDA: Electric Design Automation）の生産性改善である．設計環境への要求を整理し，それら観点で，論理設計環境，論理検証，遅延解析（STA: Static Timing Analysis）の3項目の生産性を改善した，そして，第一の貢献で言及した自動並列型プロセッサMerlot のLSI 試作で有効性を確認した．論理設計環境では，Verilog RTL入力のフロントエンドツールを作成し，記述量を30-40% に削減した．信号名規則とツールによるブロック間自動結合を実装し，記述量を10% 未満に削減した．さらに多地点開発に対応したRTL 版管理を構築した．論理検証では，RTL 版管理からのRTL 改版情報をトリガに夜間自動検証し，IPC（命令/Cycle）やRTL タイミング結果を含む論理品質を自動グラフ化した．遅延解析（STA）では，RTL 入力のport 記述でタイミング条件を記述し，これと版管理とRTL タイミングツールを連携させ，RTL タイミングによるSlack（遅延余裕）値から論理合成制約を自動生成した．RTL タイミングにより，従来のSTA では1 週間以上かかったTAT（Turn Around Time）を30分に短縮し，最終RTL のフロアプランを用いたSTA 結果に対して2% の遅延誤差を実現した．RTL タイミングによる遅延改善は局所配線混雑や迂回の影響がなく，遅延改善の良好な収束性を確認した．これらEDA 環境の改善により，Merlot の試作LSI では，ファーストシリコンにおいて簡易OS を走行させ，ベンチマークアプリケーションの測定に成功した．