トップエスイー修了制作

性能モデル検査技術を用いた 性能最適化支援手法の提案と評価

長野　岳彦
Takehiko Nagano

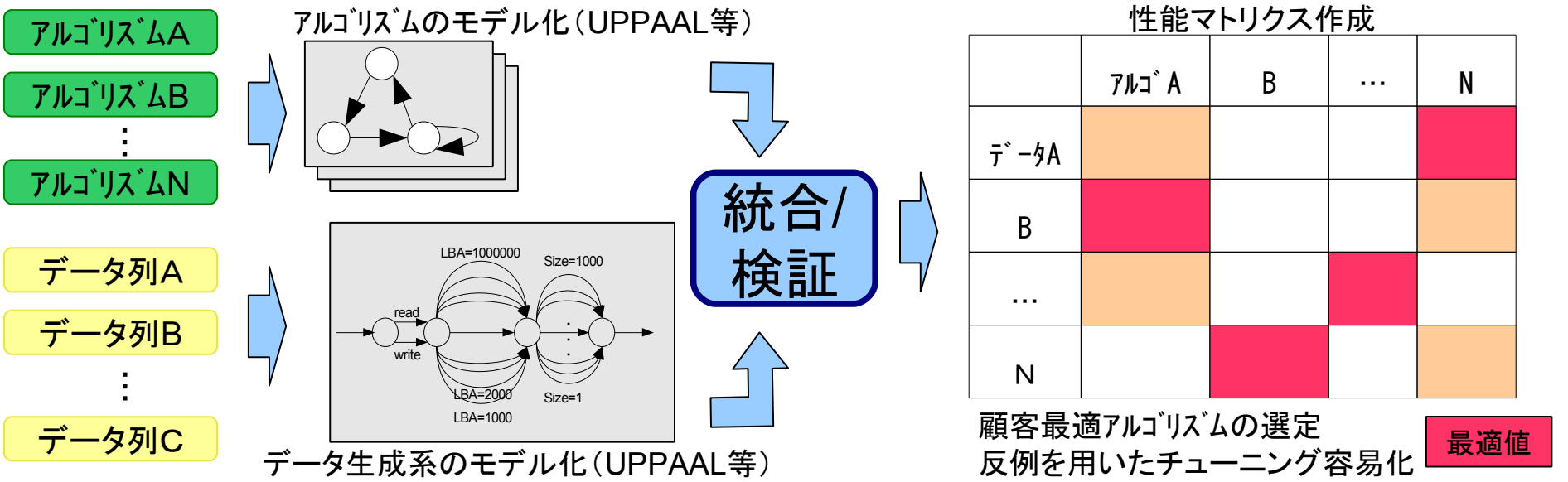
開発における問題点

現在，情報システムの性能問題が増加
⇒システム障害，工程遅延が多発
顧客毎に異なる性能実現が必要なシステムは，性能チューニングを実施し，要求性能を実現
⇒経験・想定を根拠に性能テスト実施/検証するが，網羅性が低く，出荷前後の性能問題顕在化/対策工数の発生という問題に直結

モデル検査を用いた性能検証

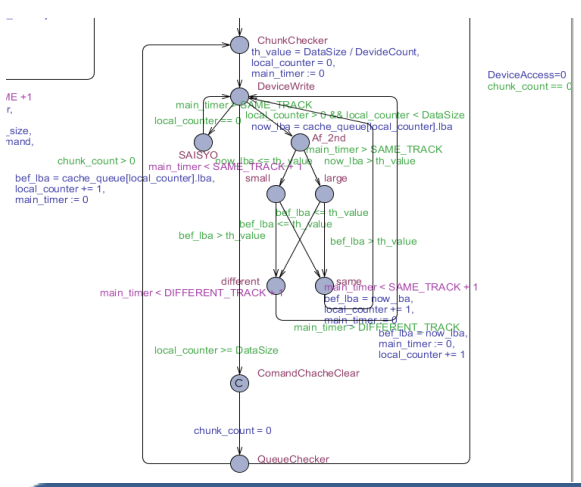
処理アルゴリズム×データパタンで実行時間の網羅検証を実施
⇒実行時間を指標としたマトリクスを作成
上記マトリクスを活用し
1)顧客毎の最適（最小実行時間）アルゴリズム決定
2)アルゴリズム選択ポリシの決定（出荷後動的最適化の実現支援）
シミュレータに対し，短時間で網羅的に検証が可能

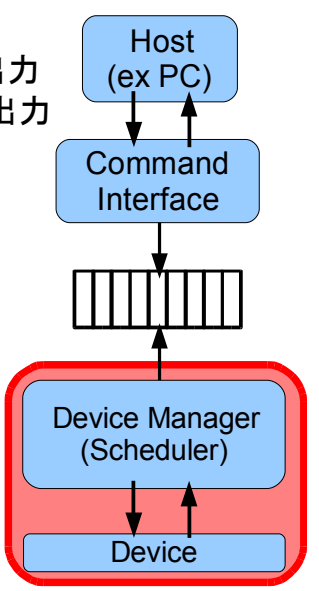
提案手法



実システム適用例

ディスクシステムの性能検証モデル
目的：ベンチマーク（ワークロード）毎の最適アルゴリズム選定
　　動的なスケジューリングポリシの決定
入出力：
1.ワークロード（データ）を入力/実行時間を出力
2.データパタンを網羅的に生成/実行時間を出力





検証結果/課題

結果

実行時間を検証する検証式
E<>(total_time <= ***minimum_execution_time***
　and state.finish)
A[] (total_time <= ***minimum_execution_time***
　and state.finish)等

性能マトリクス作成結果

	データA	データB
アルゴリズムA	1900clock	9100clock
アルゴリズムB	1990clock	1990clock

課題

簡単に**状態爆発**が発生
⇒データ生成系を完全にモデル化すると、状態爆発
UPPAALでかつJavaのヒープサイズを768Mにした場合
並列度3以上，検証時間サイズを100以上で状態数が1Mを超えて状態爆発を起こす
⇒今後状態爆発を抑える手法を検討する
例：状態遷移モデルと非状態遷移モデルを結合する