

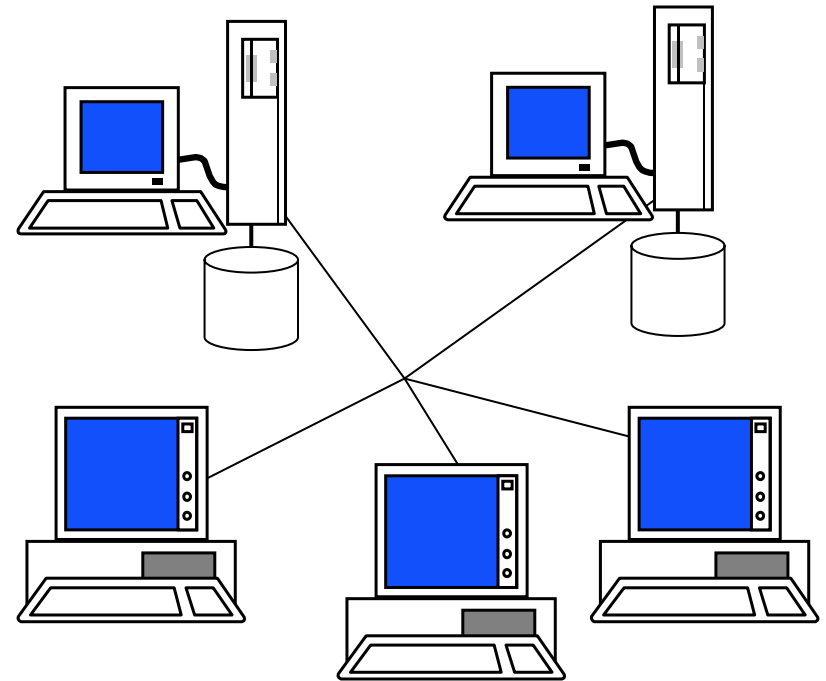
アイドル計算機を用いた分散 ODB処理の効率化の一検討

群馬大学

吉田裕介, 有次正義, 金森吉成

背景

- 分散環境の発達
 - クライアントで開発, サーバでデータの保存・管理
- 従来のODBのアーキテクチャ
 - メソッドはDBに保存しない
- データとメソッドは異なるサイト



背景

- 今までに
 - データマイグレーションとメソッドマイグレーションの組合せ
 - 効率の良い組合せを求める

という研究を行ってきた

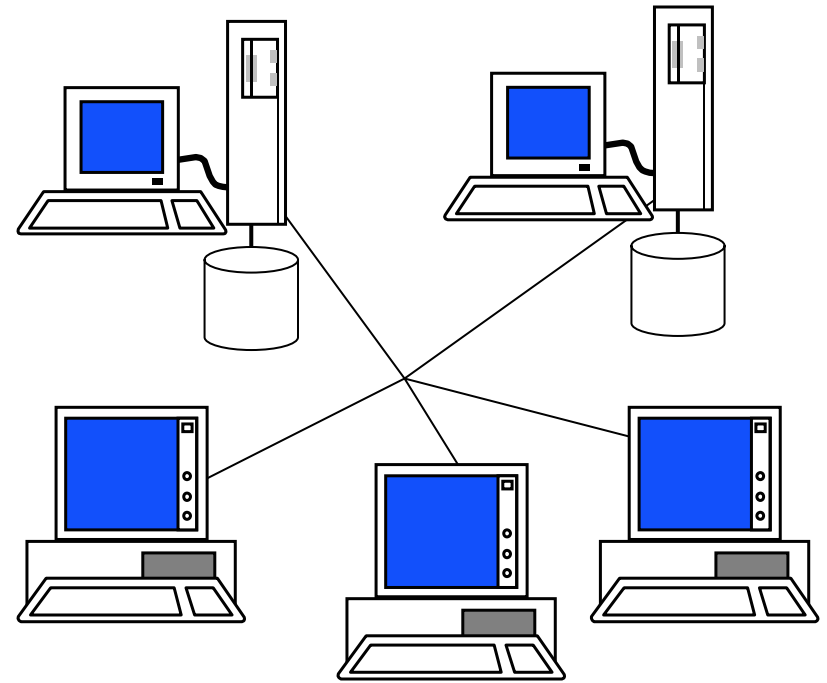
- 効率の良い組合せの決定には、サーバの負荷による影響が大きい

目的

- データマイグレーション/メソッドマイグレーションに加え、双方を移動しアイドル計算機でメソッドを実行
- アイドル計算機を有効利用して、データマイグレーションとメソッドマイグレーションのみの組合せよりも効率化する

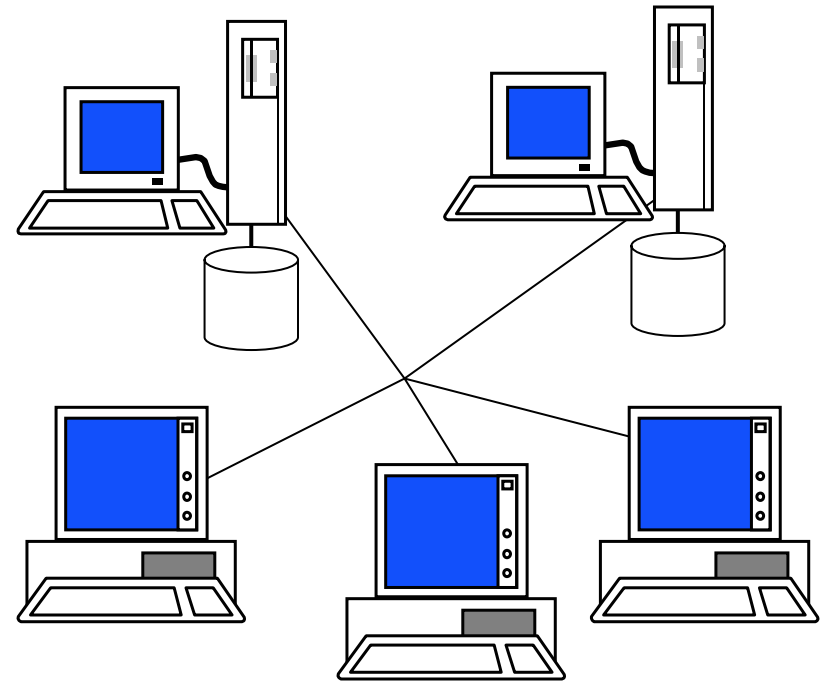
データマイグレーション - DM

- データを転送
- クライアントでメソッドを実行
- 利点
 - サーバの負荷を分散
- 欠点
 - 通信コストの増加
 - クライアントに負荷が集中



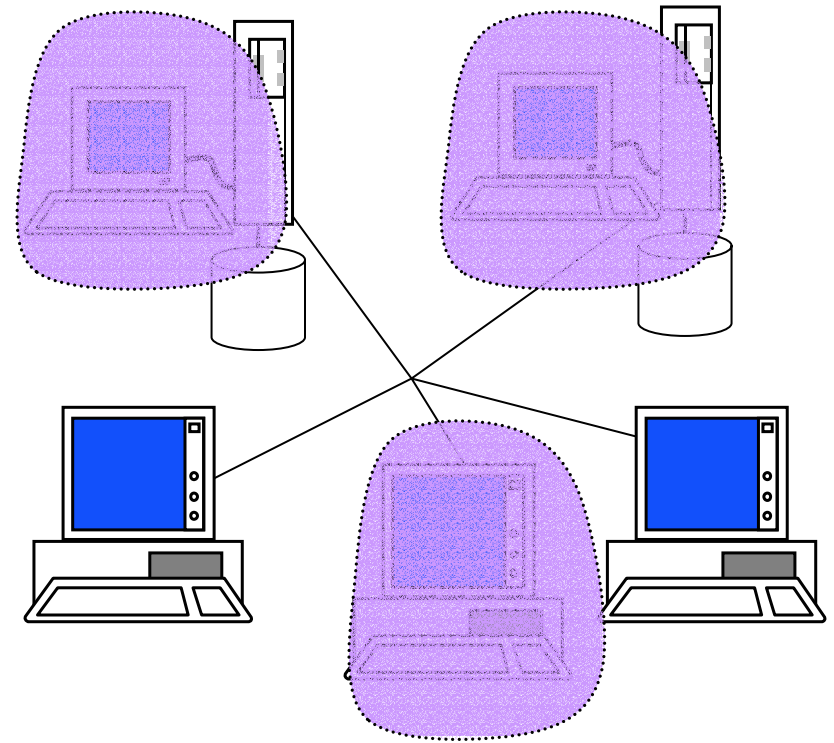
メソッドマイグレーション - MM

- メソッドを転送
- サーバでメソッドの実行
- 利点
 - 並行処理
 - 結果の転送コスト減
- 欠点
 - サーバに負荷が集中



データマイグレーションと メソッドマイグレーション

- 2つの処理手法
 - データマイグレーション
 - メソッドマイグレーション
- サーバが n 台のとき, 2^n 通りの組み合わせ
 - 例: dm, dd, md, mmの4通り
- この組み合わせの中から効率の良いものを選択

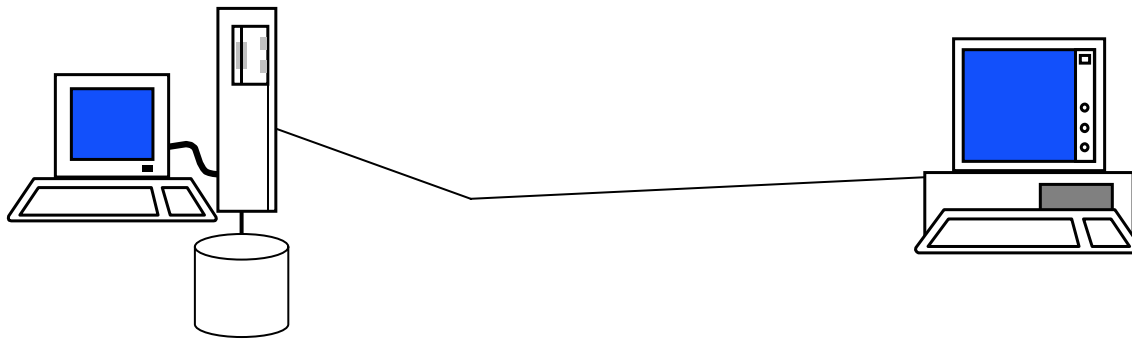


効率の良い組合せを決定する 方法

- 特定の組合せを与えたときのレスポンスタイムを見積もるコストモデル
- 2^n 通りの組合せから最も効率の良い組合せを決定するアルゴリズム
- 実験結果と比較

実験で用いた処理

- PersonSet : オブジェクトPersonの集合
- PersonSet.averageSalary(age) : age歳以下のPersonオブジェクトの平均給与を求める



データマイグレーションとメソッドマイグレーションのみの組合せ

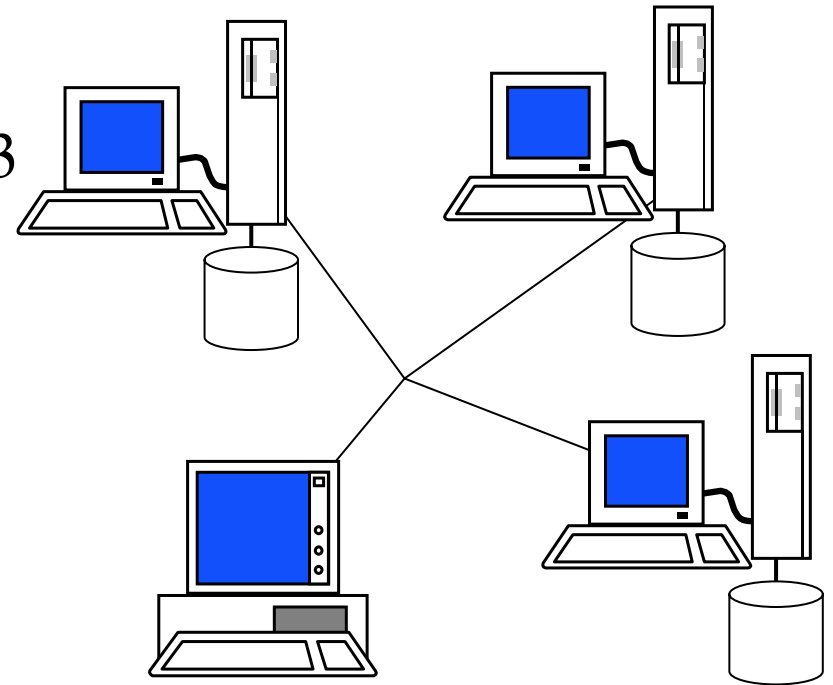
- サーバ: S_1, S_2, S_3
- サーバの負荷: ρ_1, ρ_2, ρ_3

ρ_1	ρ_2	ρ_3
0.2	[0.2, 0.8]	[0.2, 0.8]
0.5	[0.2, 0.8]	[0.2, 0.8]
0.8	[0.2, 0.8]	[0.2, 0.8]

メソッド実行結果の割合

$$f = 0.5$$

- $3 * 7 * 7 = 147$ 通り



誤差率

$$\frac{(\text{resp. time of est.}) - (\text{resp. time of expr.})}{\text{resp. time of expr.}}$$

• 効率の良い組合せの見積もりを誤った場合の損失の指標

• 例:

組合せ	mm	dm	md	dd
レスポンス タイム	10	12	11	15

コストモデルからmdが効率の良い組合せとして求めた場合

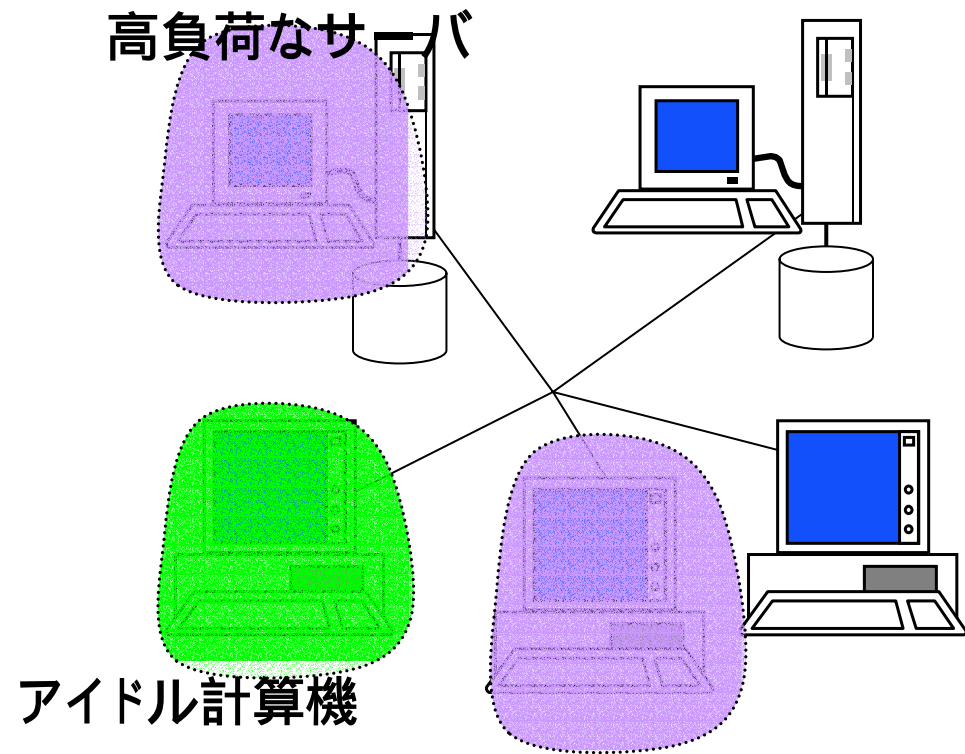
$$\text{誤差率} = (11 - 10) / 10 = 0.1$$

見積もりと実験の比較

- (1, 2, 3)の組み合わせ:147通り
 - 123通りが正しい組み合わせ:83.7%
 - 143通りが誤差率0.03以下:91.8%
- 平均誤差率:0.034
- 多くの場合で有効である
- 効率の良い組合せの決定には、サーバの負荷による影響が大きい

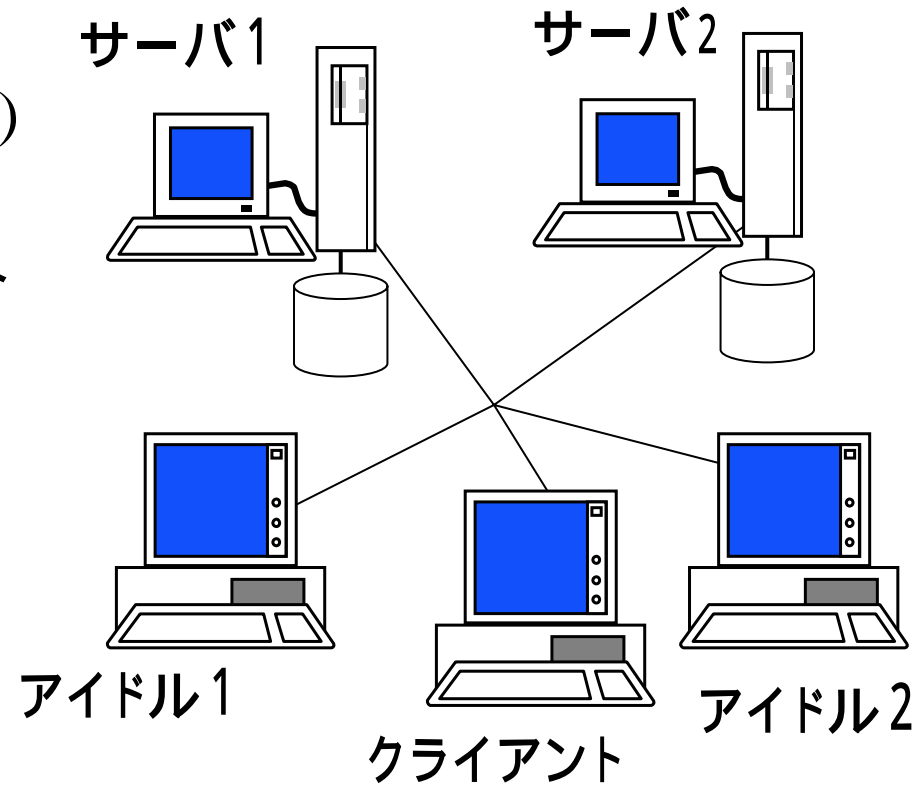
アイドル計算機の有効利用

- データとメソッドを両方移動して、アイドル計算機でメソッドを実行
- データマイグレーション/メソッドマイグレーションのみの組合せを超える効率化
- サーバの高負荷を避けられることによる効率化とデータ/メソッド転送によるオーバーヘッドとのトレードオフ



実験

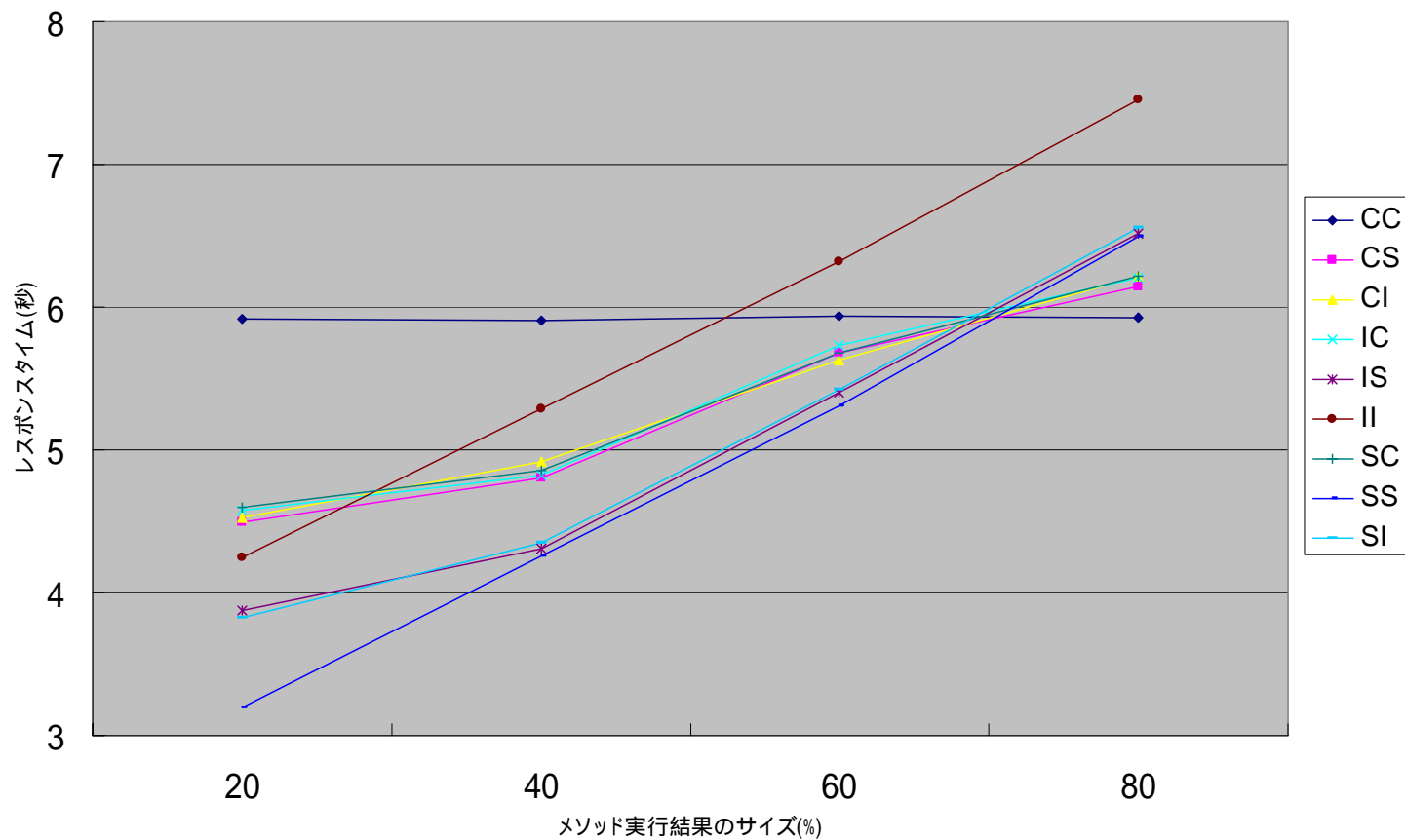
- サーバ: Ultra30(296MHz)
- クライアント: Ultra1(167MHz)
- アイドル計算機:
 - (Ultra30, Ultra1)を用いた場合
 - (Ultra30, Ultra30)を用いた場合
- PersonSet.averageSalary()を使った実験



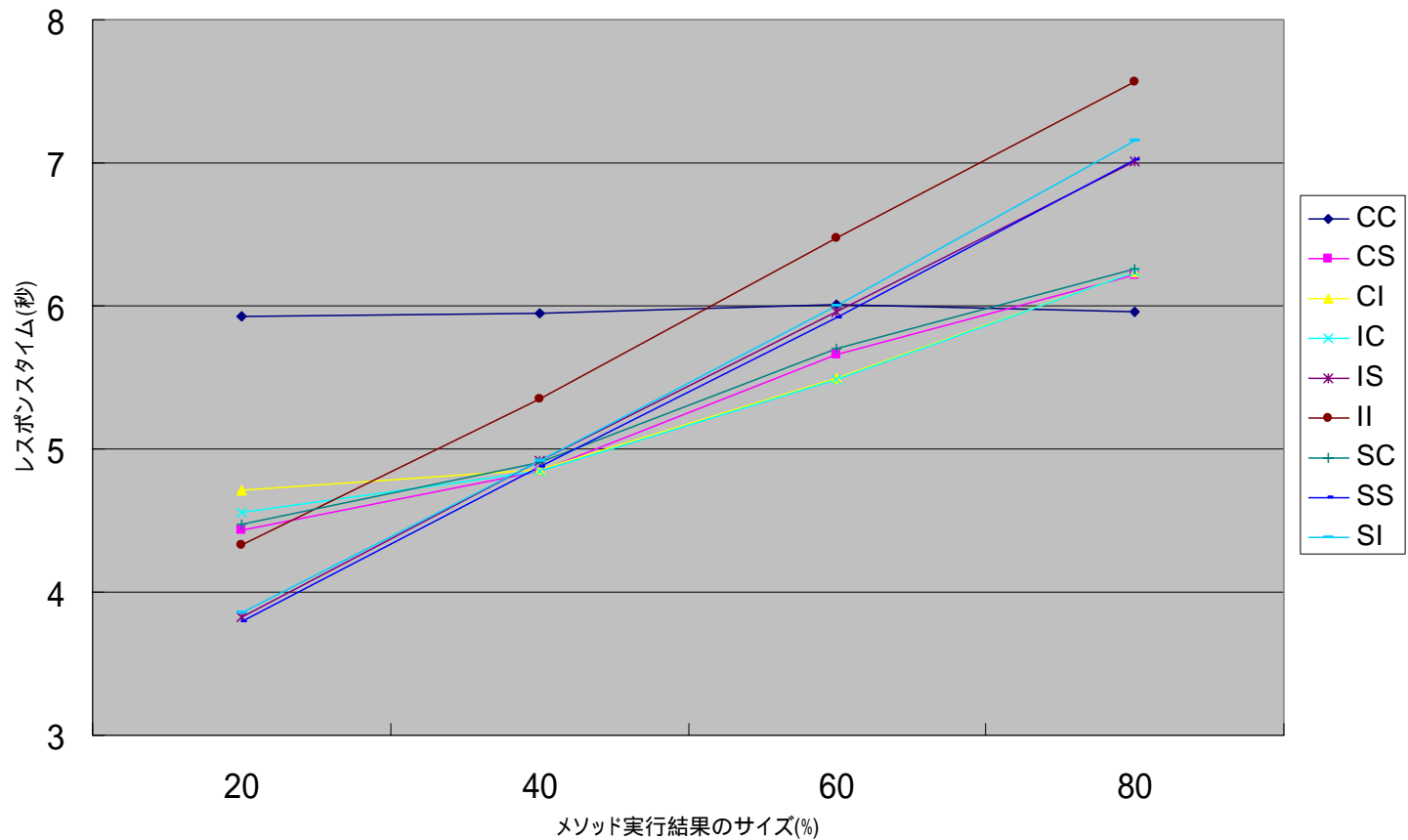
(Ultra30, Ultra1)を用いた場合

- クライアントと同じ処理能力を持つ
- アイドル計算機にデータを移動するためのオーバーヘッド

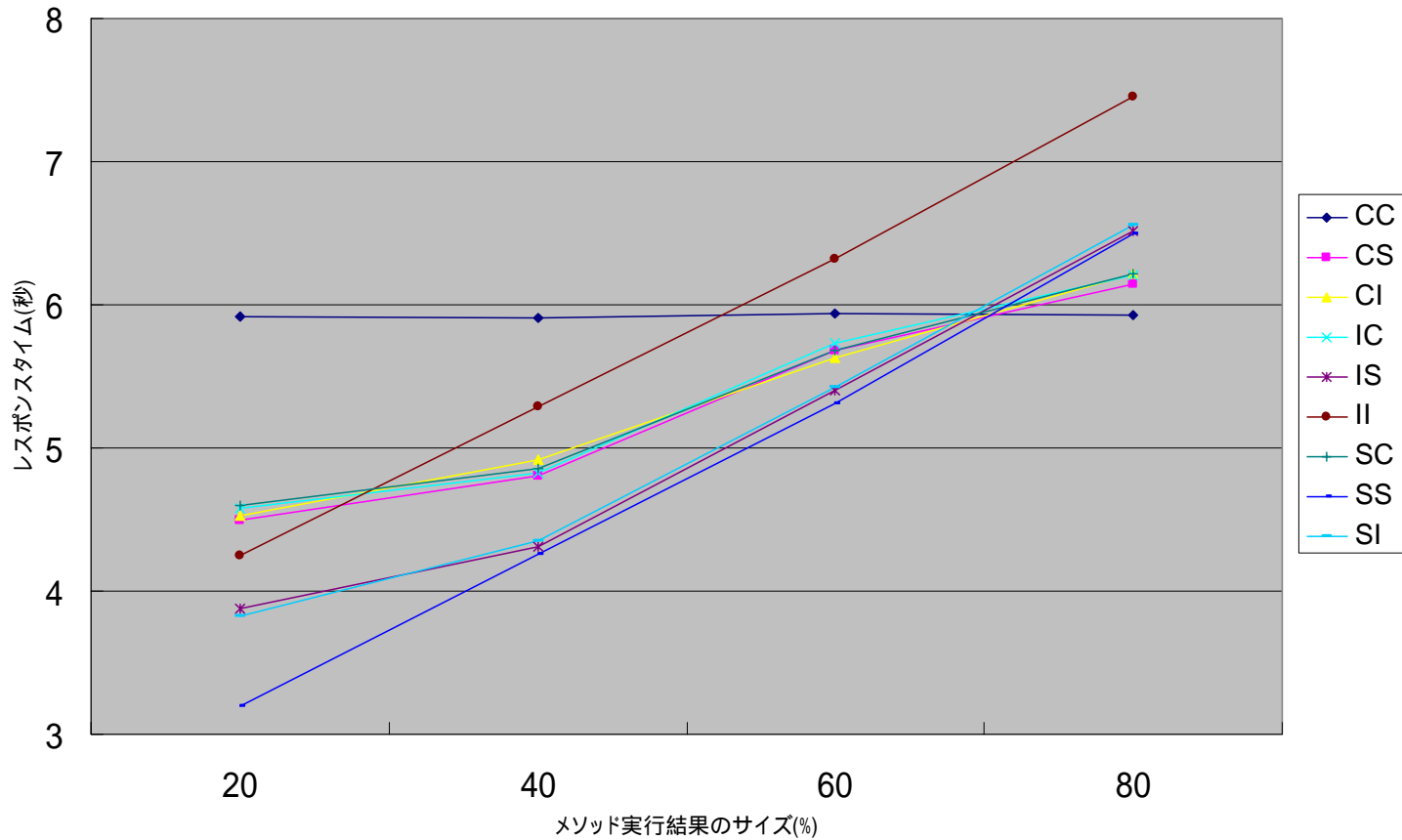
実験結果：負荷なし



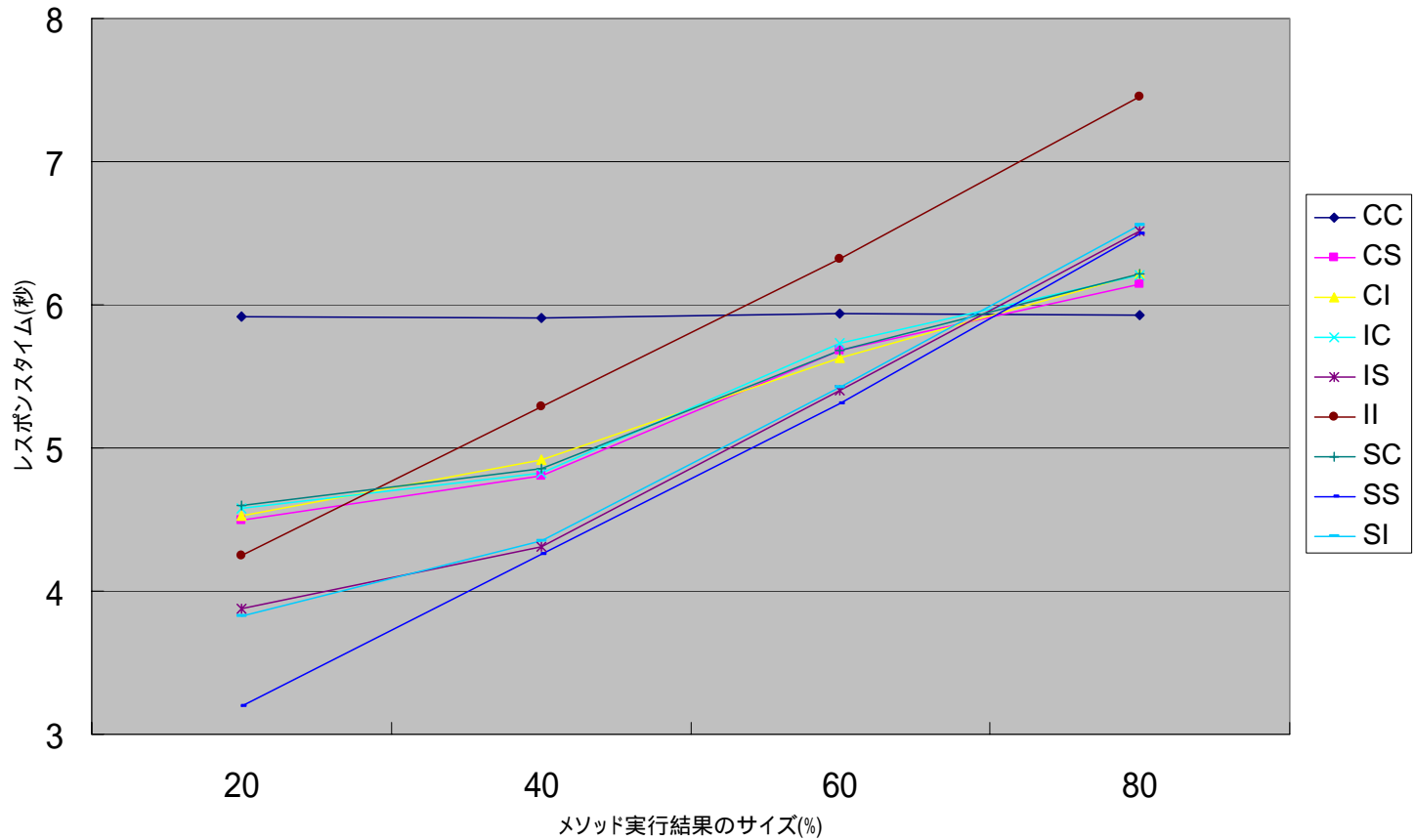
実験結果 : 負荷80%



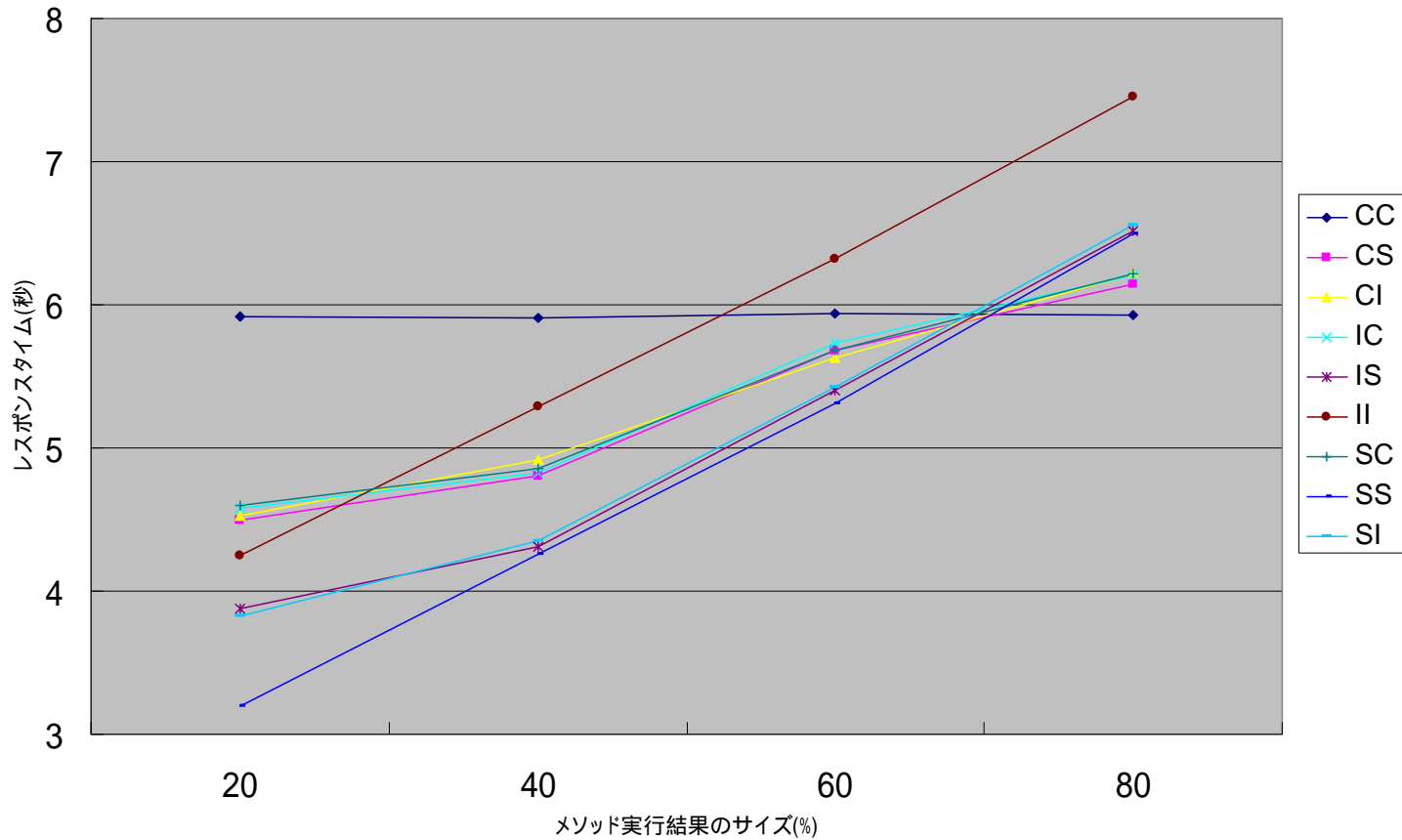
実験結果 : 負荷50%



実験結果 : 負荷なし



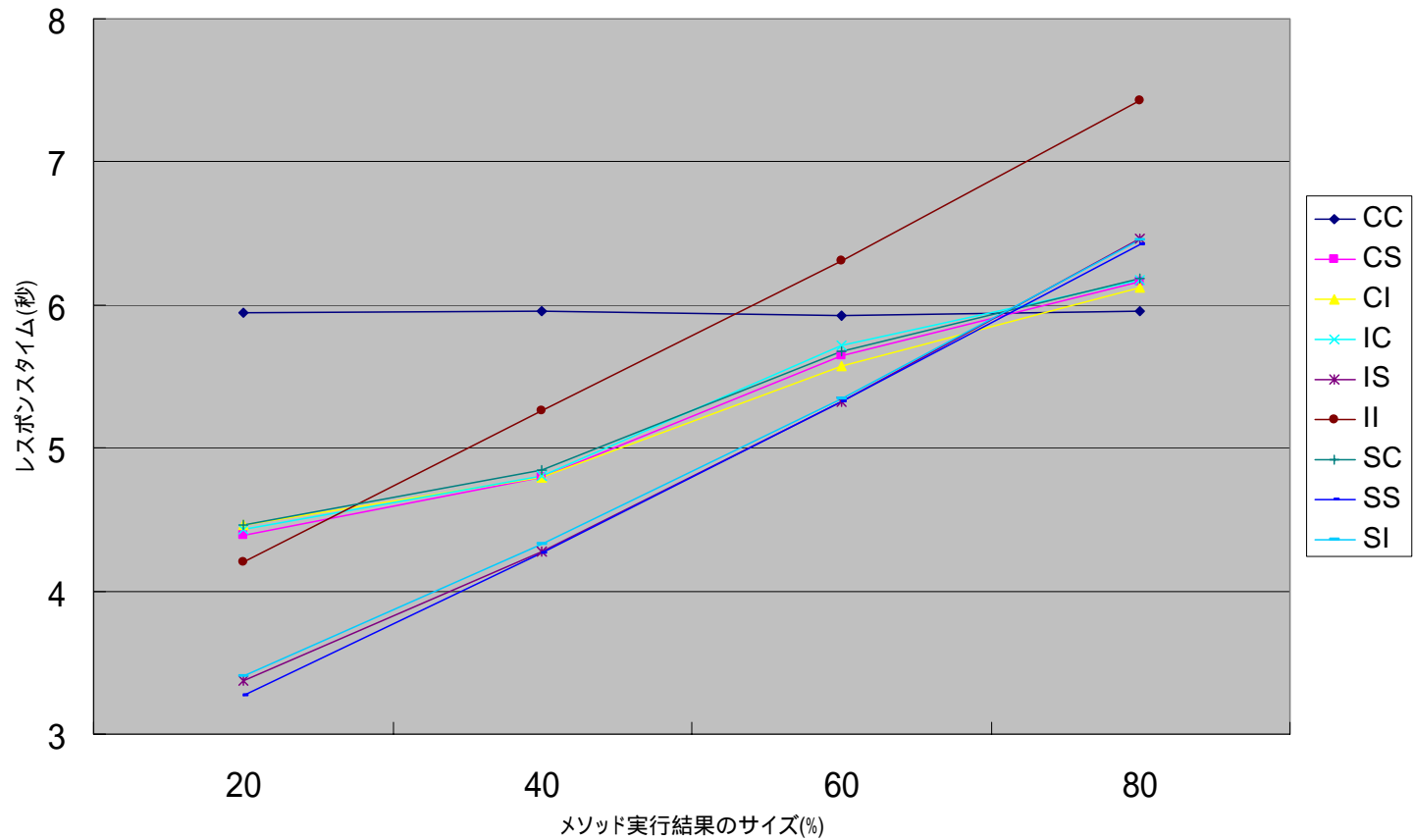
実験結果 : 負荷80%



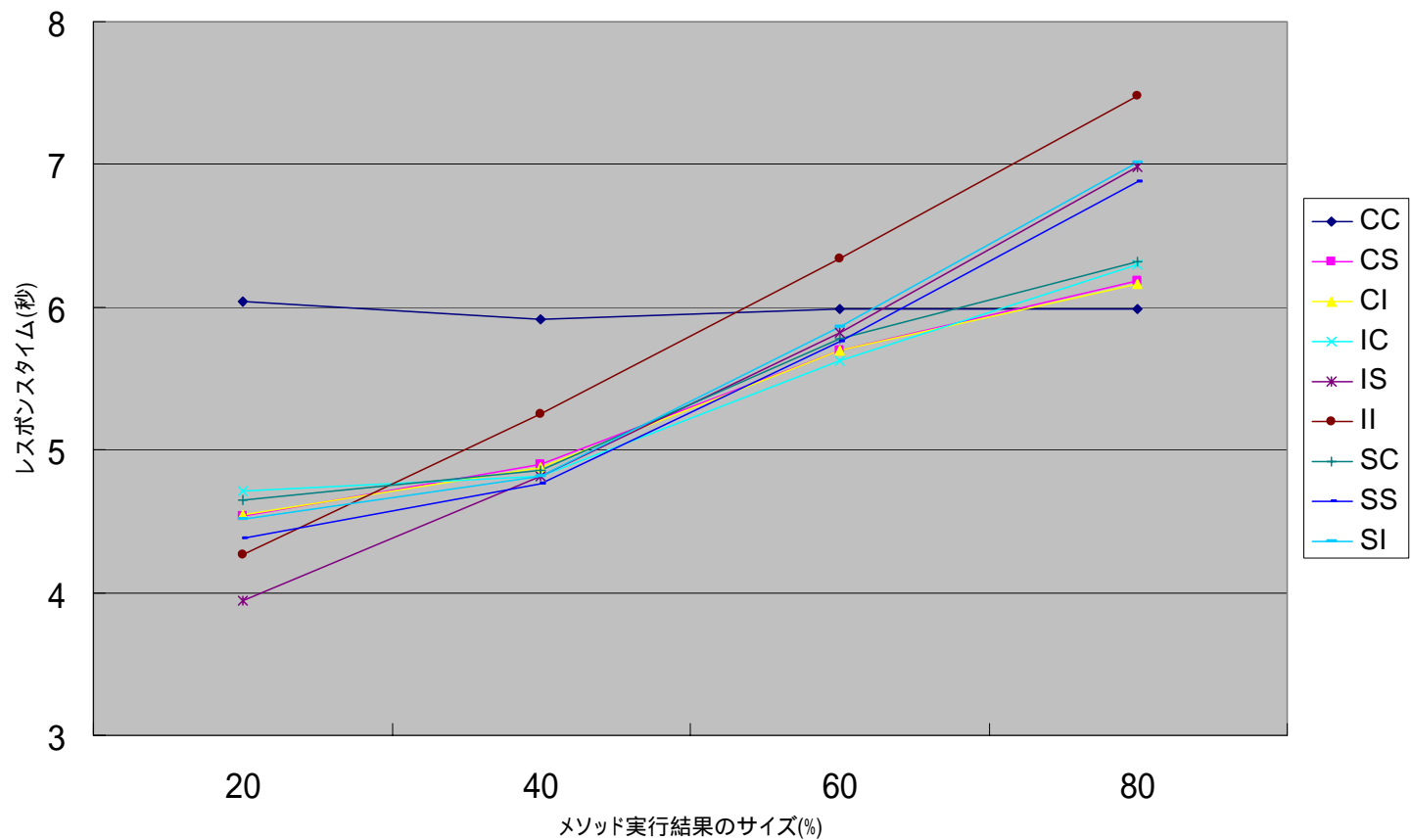
(Ultra30, Ultra30)を用いた場合

- アイドル計算機とサーバが同じ処理能力を持っている

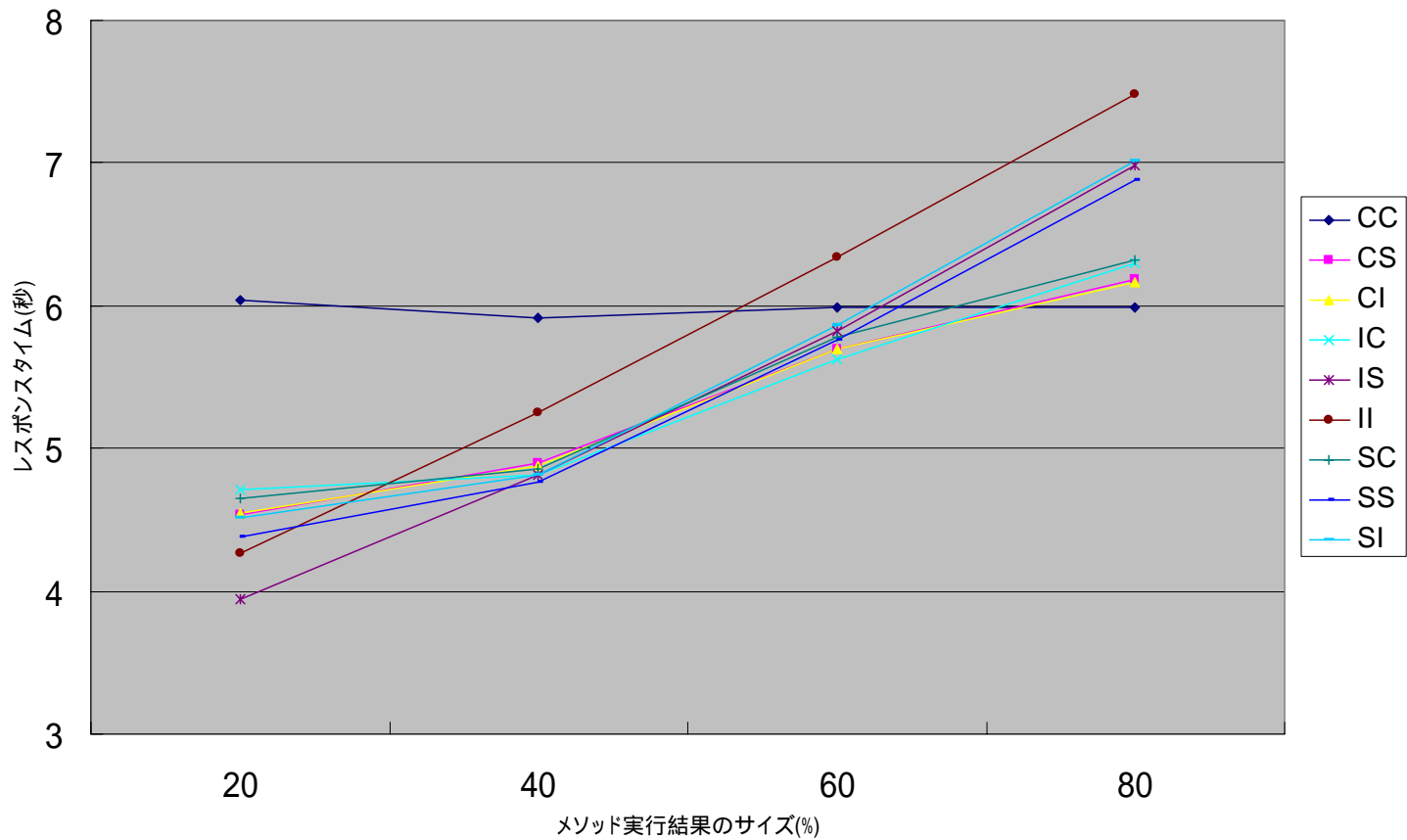
実験結果: 負荷なし



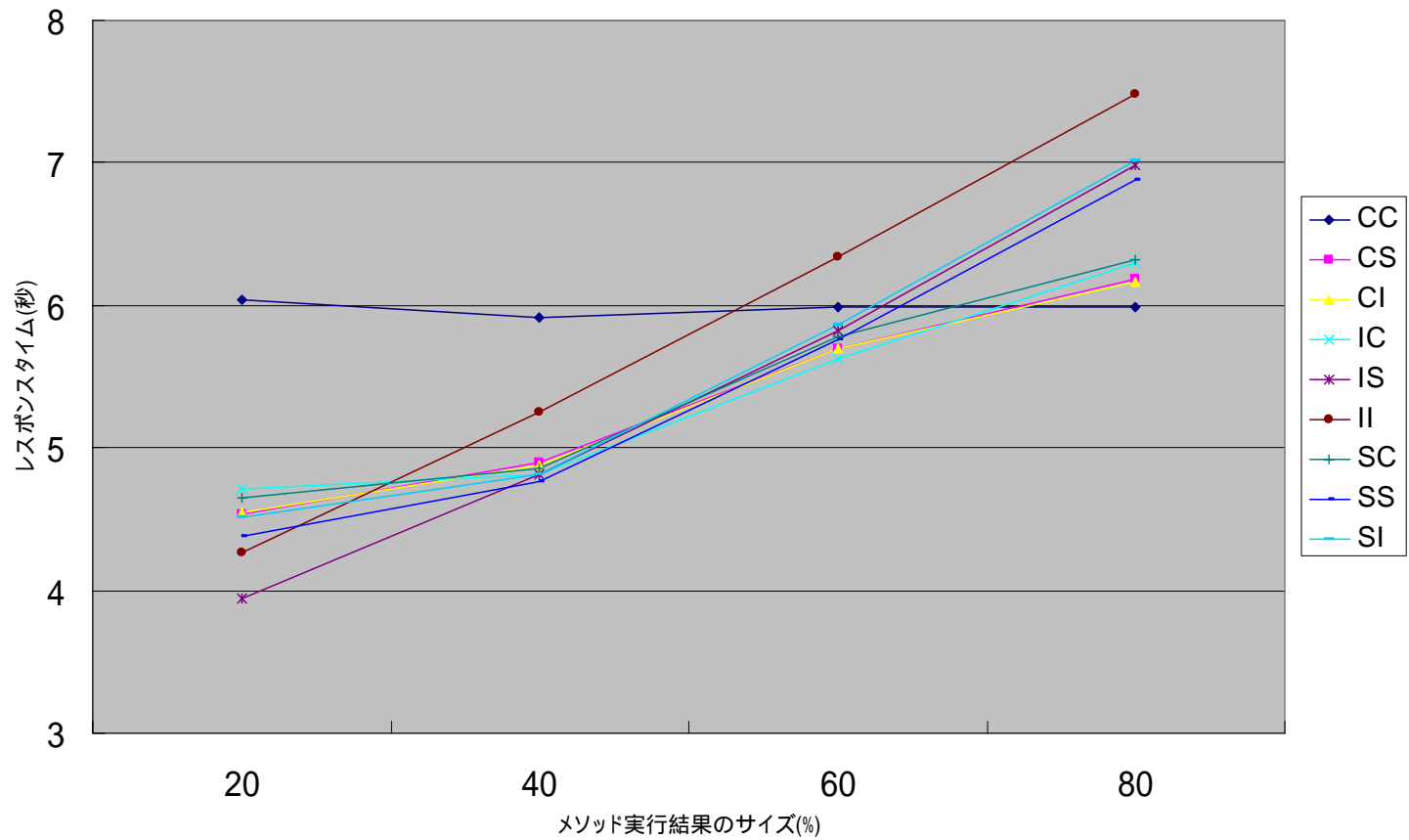
実験結果: 負荷80%



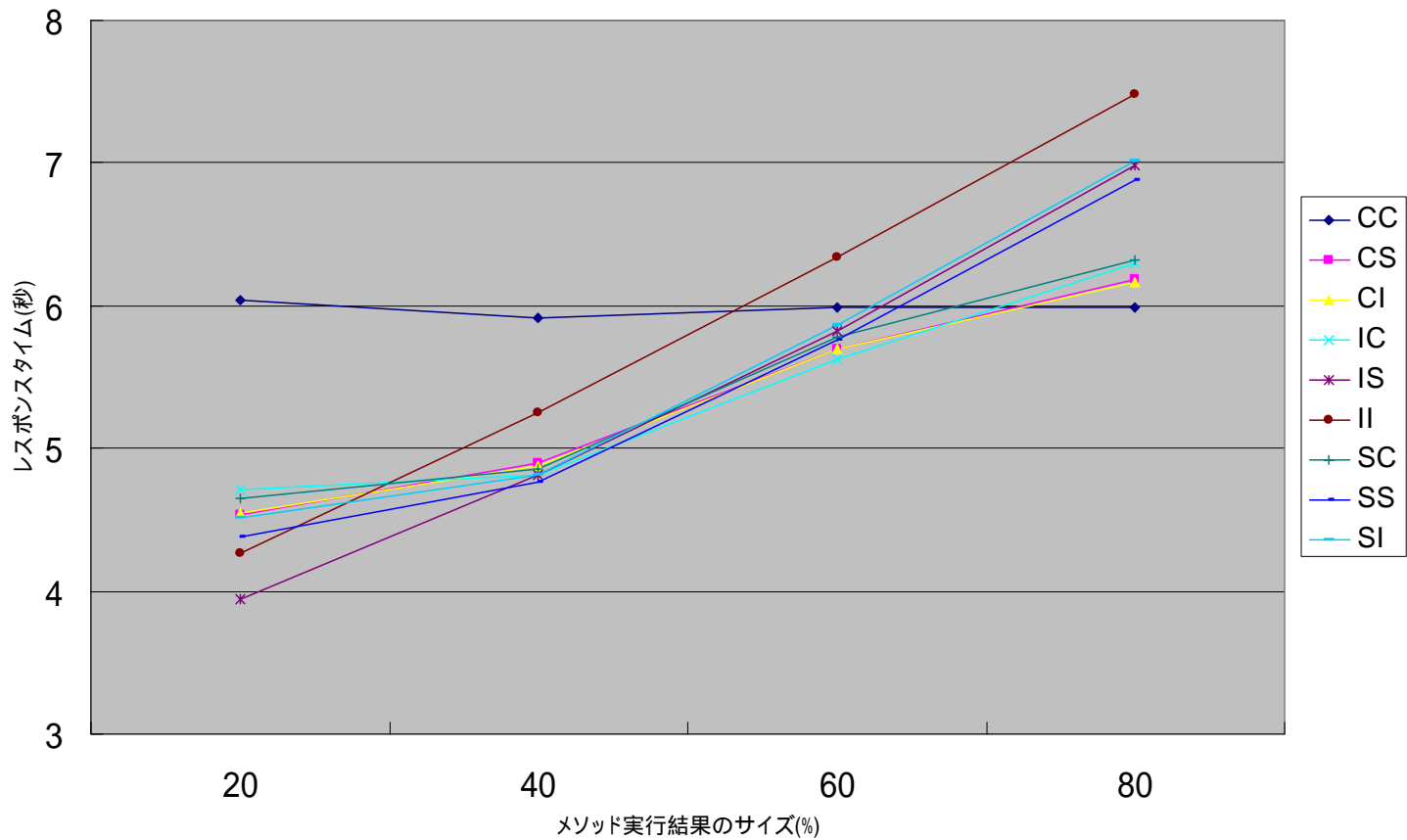
実験結果: 負荷50%



実験結果: 負荷なし



実験結果: 負荷80%



まとめ

- データマイグレーションとメソッドマイグレーションのみの組合せよりも、アイドル計算機を加えることによって、さらに効率化できる場合を確認できた

今後の課題

- メソッド実行に最適なサイトを決定するコストモデルを構築する
- 実験結果と比較する