

エキスパートレベルの性能分析を 手軽に実現

- イベントトレースによるシステム性能測定とその応用 -

2004/11/04

The logo for NEC, consisting of the letters "NEC" in a bold, blue, sans-serif font.

システム基盤ソフトウェア開発本部
エキスパート 堀川 隆

◆ 1. はじめに

- 1.1 Performance (性能)とは？
- 1.2 SIの性能goal
- 1.3 Capacity planning
- 1.4 性能評価の要素技術

◆ 2. mevaletについて

- 2.1 測定結果例(タイムチャート表示)
- 2.2 トレース対象イベント(Kernel probe)
- 2.3 構成
- 2.4 測定方式の比較
- 2.5 操作
- 2.6 複数マシンの同時測定・一括分析

◆ 3. IPS-mevalet連携

- 3.1 連携効果
- 3.2 インタフェース
- 3.3 シミュレーションまでの作業

◆ 4. Case Studies

- 4-1-1 適用例(1/8)
- 4-1-2 適用例(2/8) - トランザクションサマリーの生成
- 4-1-3 適用例(3/8) - AMGによるXMLモデルの生成
- 4-1-4 適用例(4/8) - Topologyへのネットワークの追加
- 4-1-5 適用例(5/8) - シミュレーション実行結果 1/2
- 4-1-6 適用例(6/8) - シミュレーション実行結果 2/2
- 4-1-7 適用例(7/8) - 最適解の探索 1/2
- 4-1-8 適用例(8/8) - 最適解の探索 2/2

- 4-2-1 測定環境の概要
- 4-2-2 測定イメージ(1/3)
- 4-2-3 測定イメージ(2/3)
- 4-2-4 測定イメージ(3/3)
- 4-2-5 シミュレーションモデルのイメージ
- 4-2-6 シミュレーション結果の例

◆ 5. 利用局面

- 5.1 システムのライフサイクルと性能評価
- 5.2 処理単位毎のプロファイル分析(応用)
- 5.3 処理単位毎のプロファイル分析(具体例)
- 5.4 システム移行時のコンサルテーション

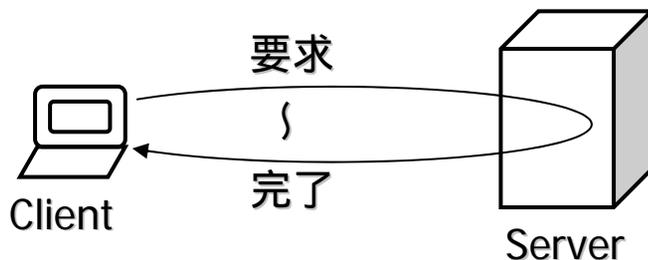
◆ 6. まとめ

IPS-mevalet連携の概要、
効果、利用局面について
説明します。

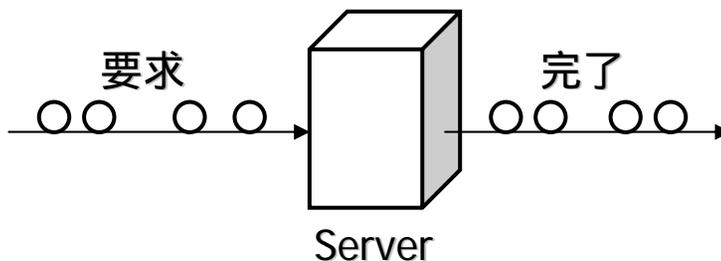
1.はじめに

1.1 Performance (性能) とは？

◆ 単純にいうと、レスポンスとスループット



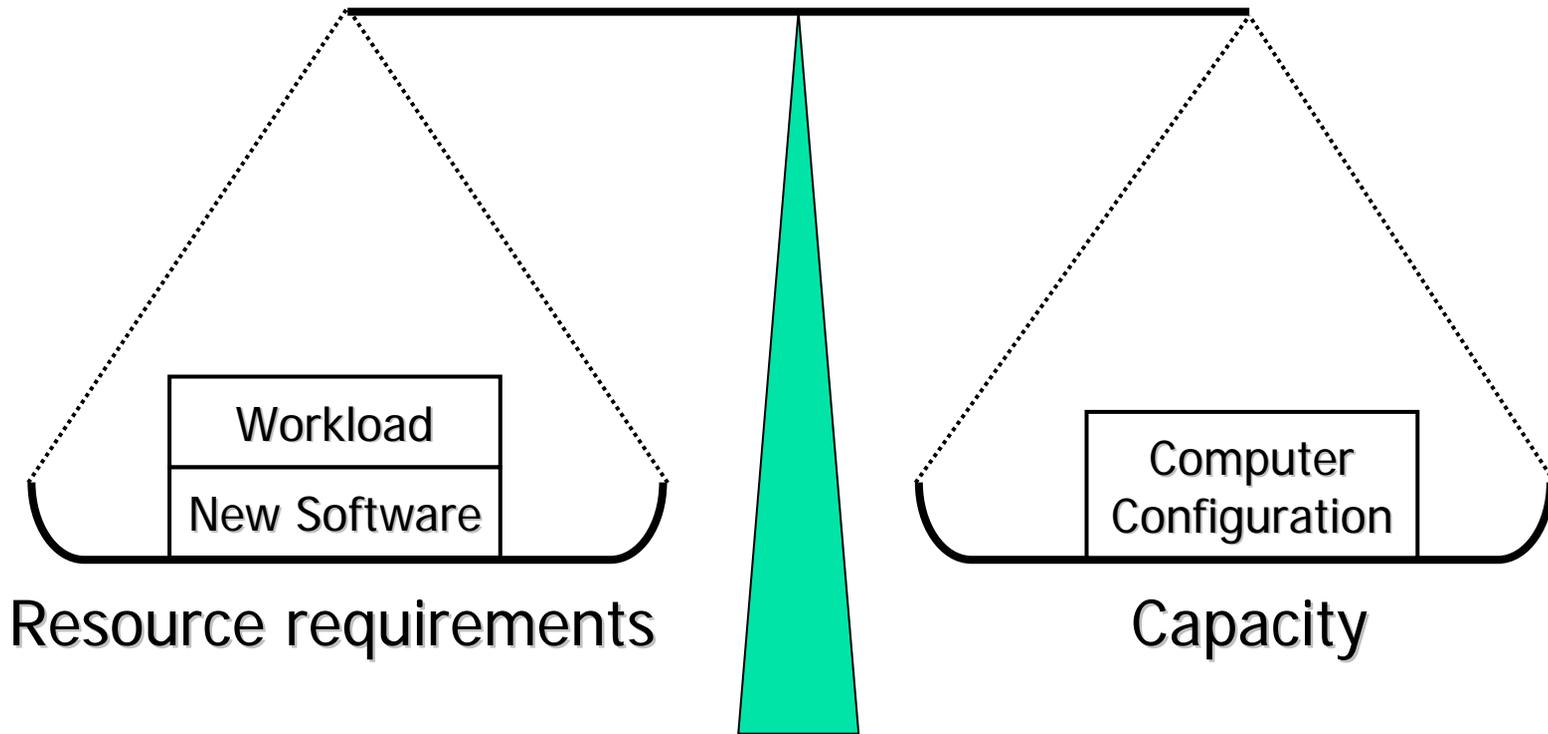
レスポンス: 処理の要求から完了までの時間



(目に見える)
結果

スループット: 単位時間に完了する処理量(数)

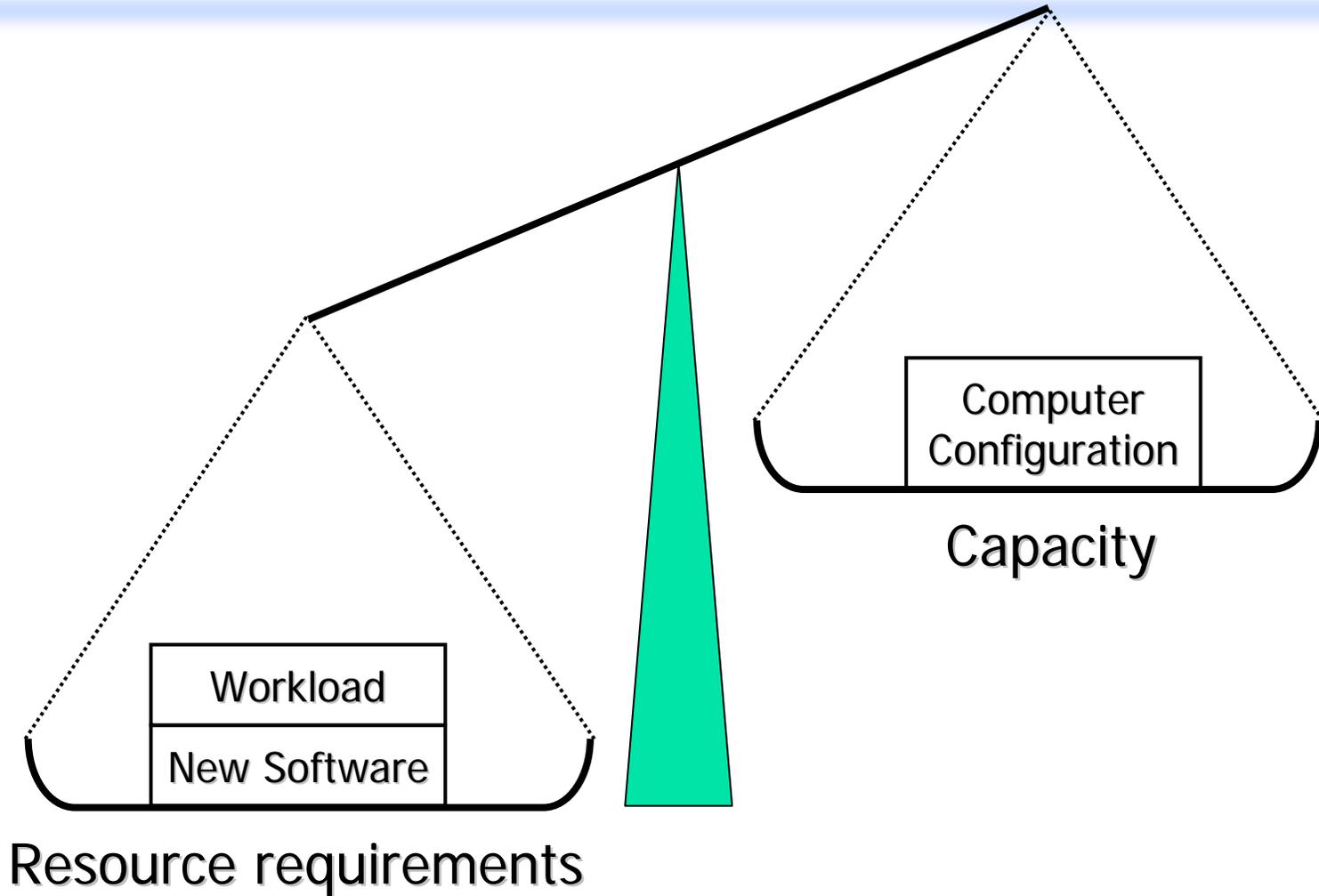
1.2 SIの性能goal



Performance Balance

C. U. Smith, Software Performance Engineering, Addison-Wesley, Reading MA, 1990より

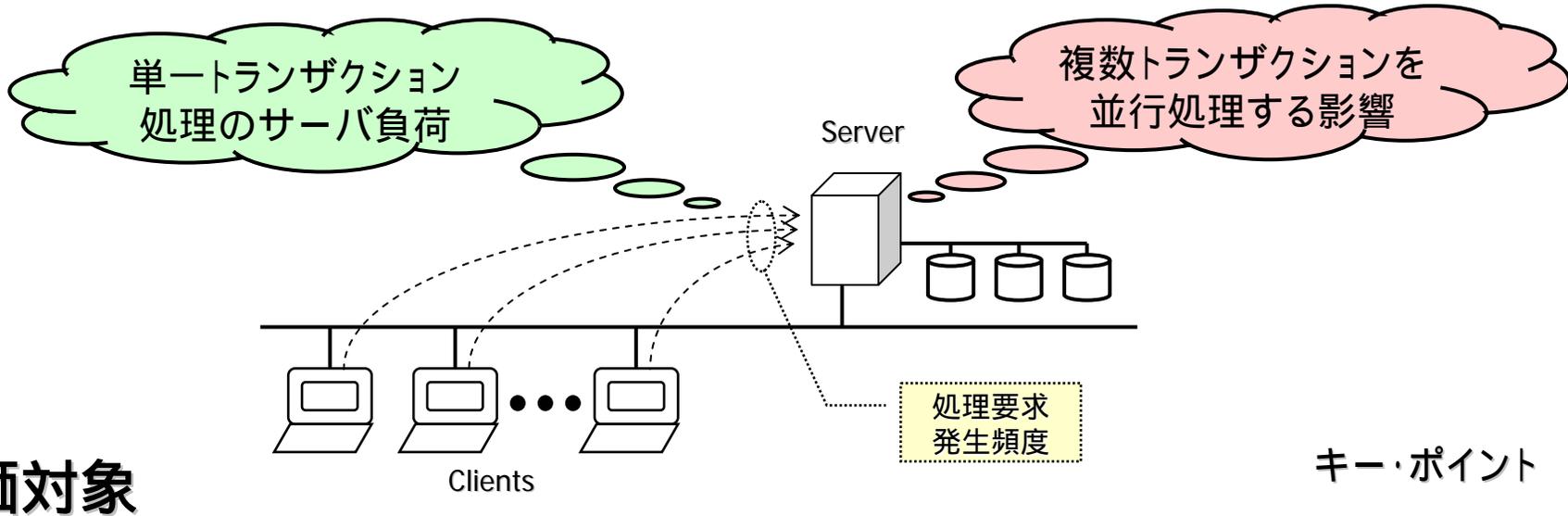
1.2 SIの性能goal



Performance Balance

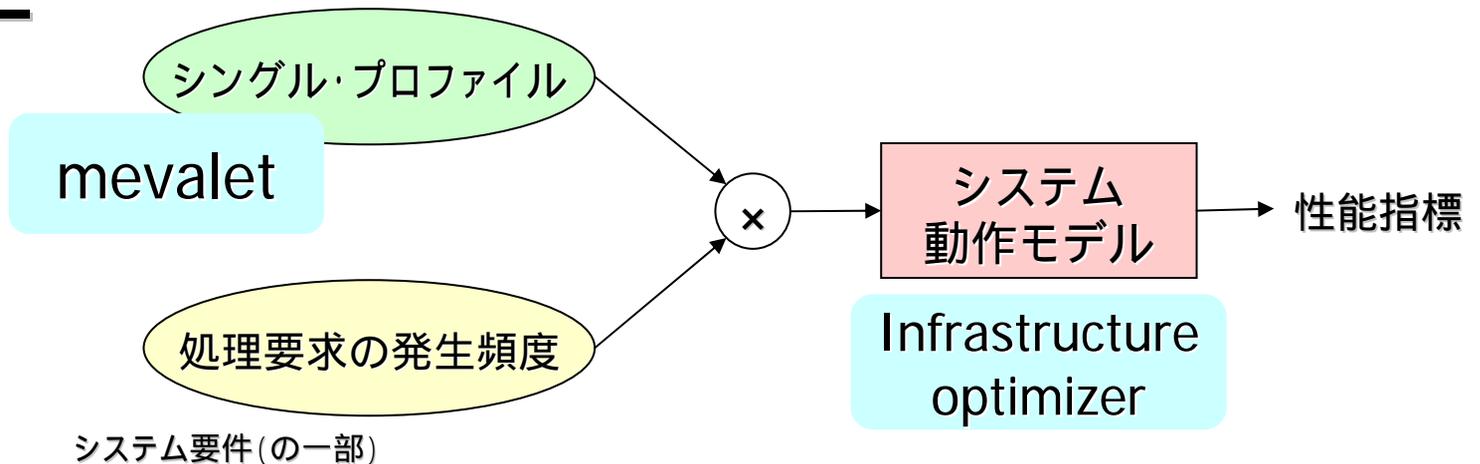
C. U. Smith, Software Performance Engineering, Addison-Wesley, Reading MA, 1990より

1.3 Capacity planning

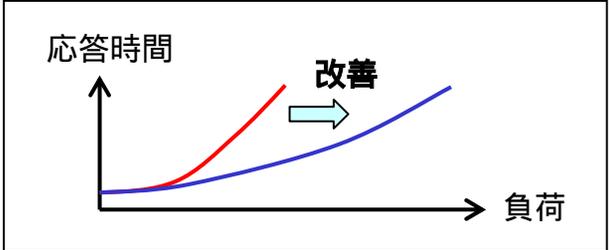
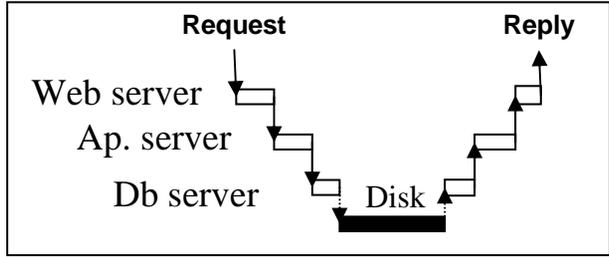
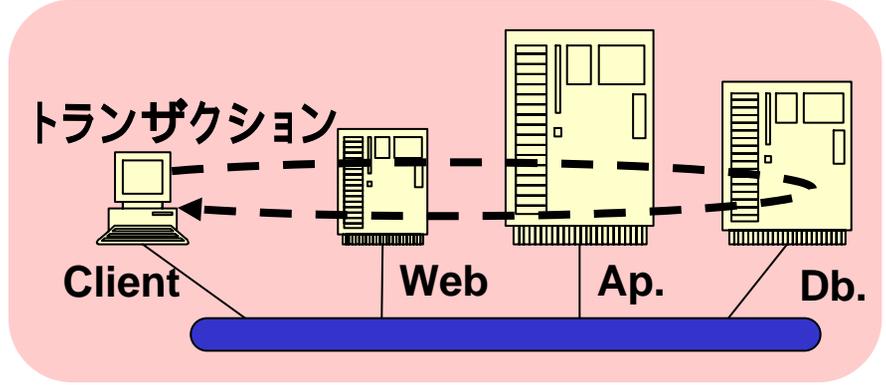
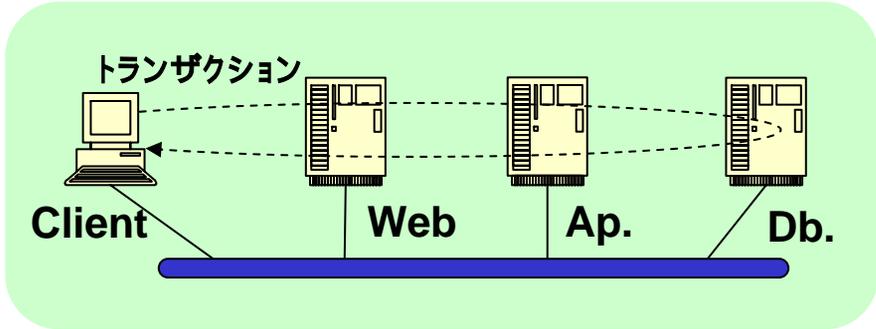


評価対象

評価フロー



1.4 性能評価の要素技術



測定・分析

「どこで何が起きているか」
を把握

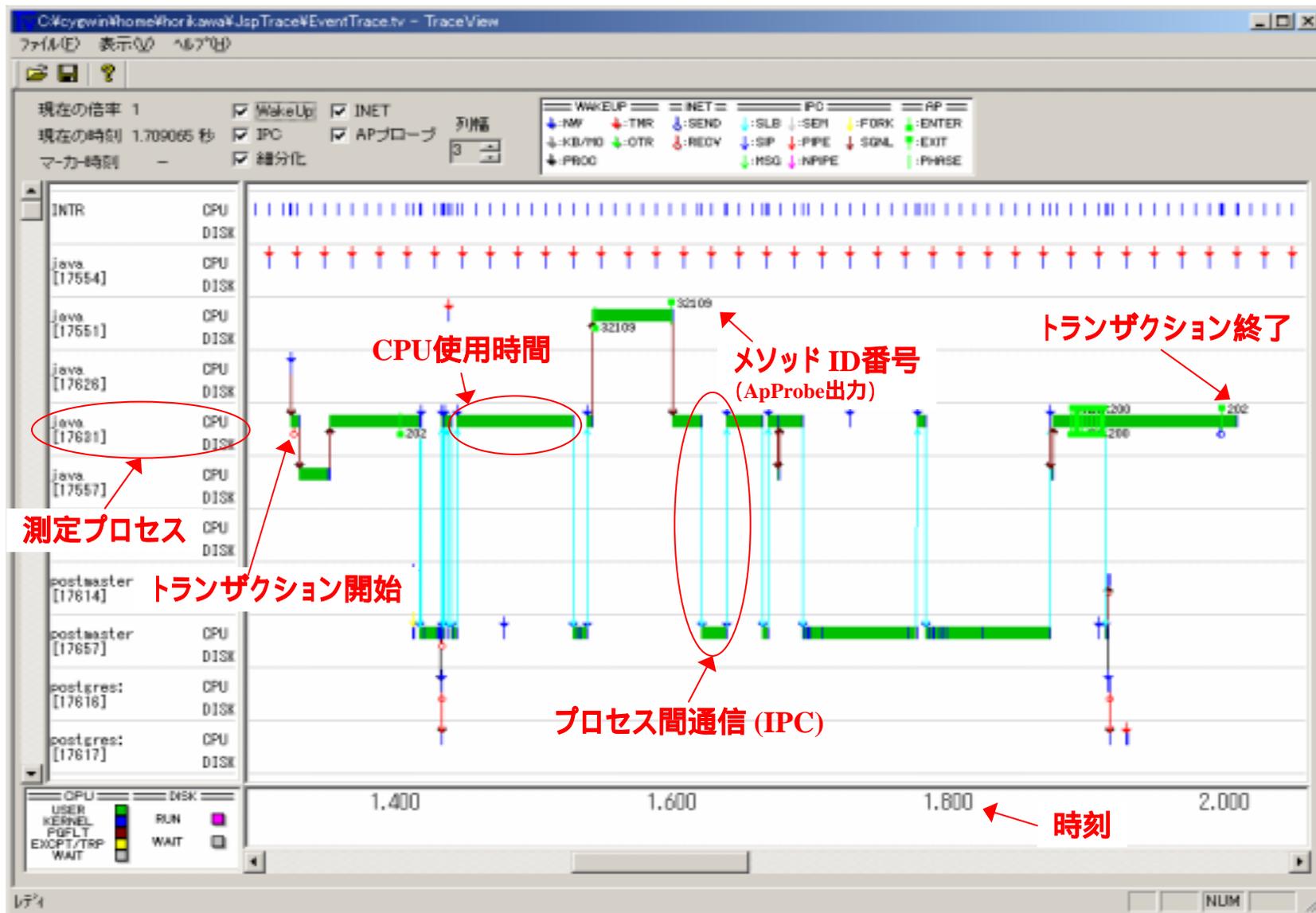
予測

「システムが変わるとどうなるか」
を見積もる

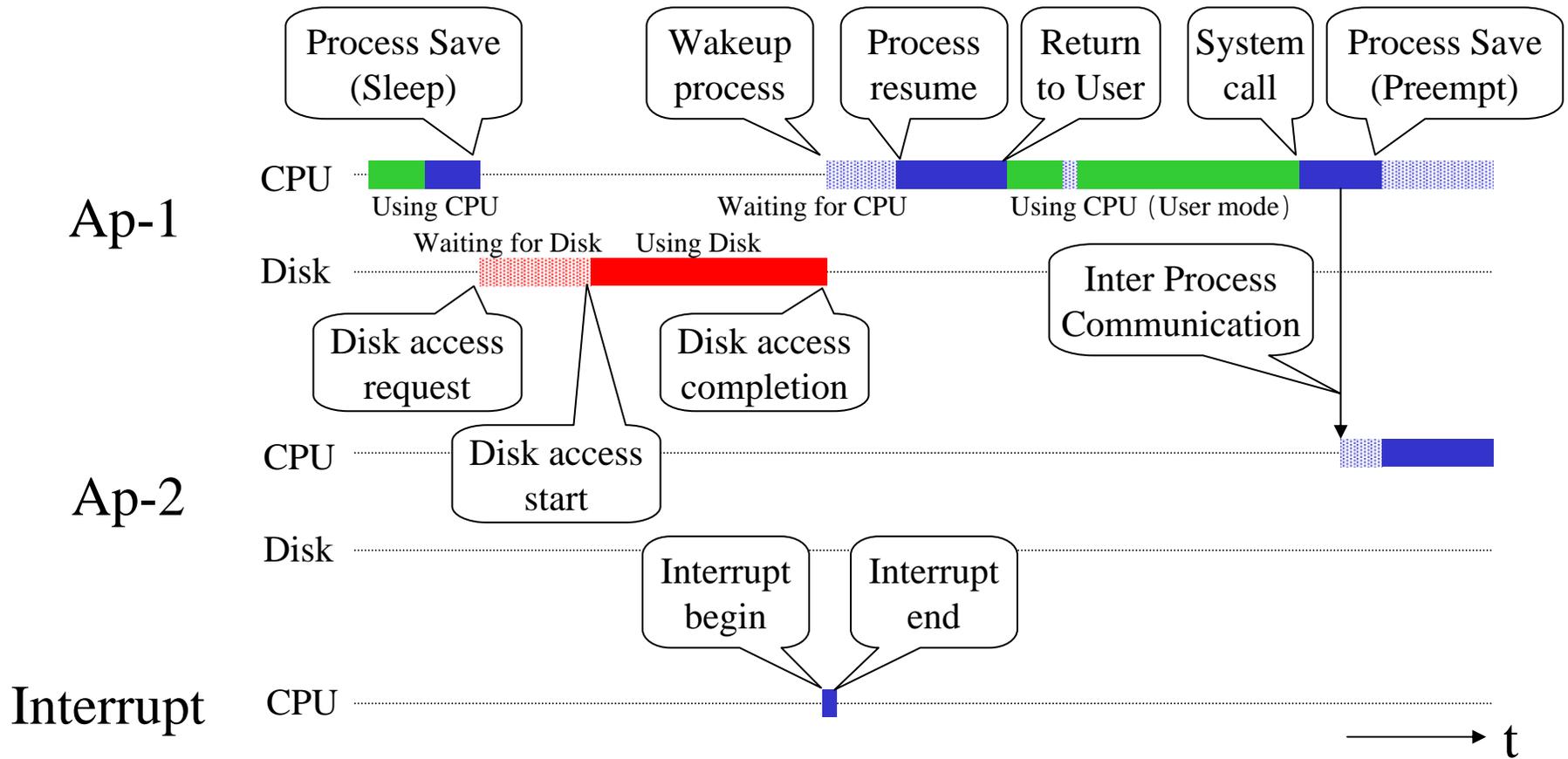
2. mevaletについて

イベント・トレース・ベースの性能測定・分析ツール

2.1 測定結果例(タイムチャート表示)



2.2 トレース対象イベント (Kernel probe)

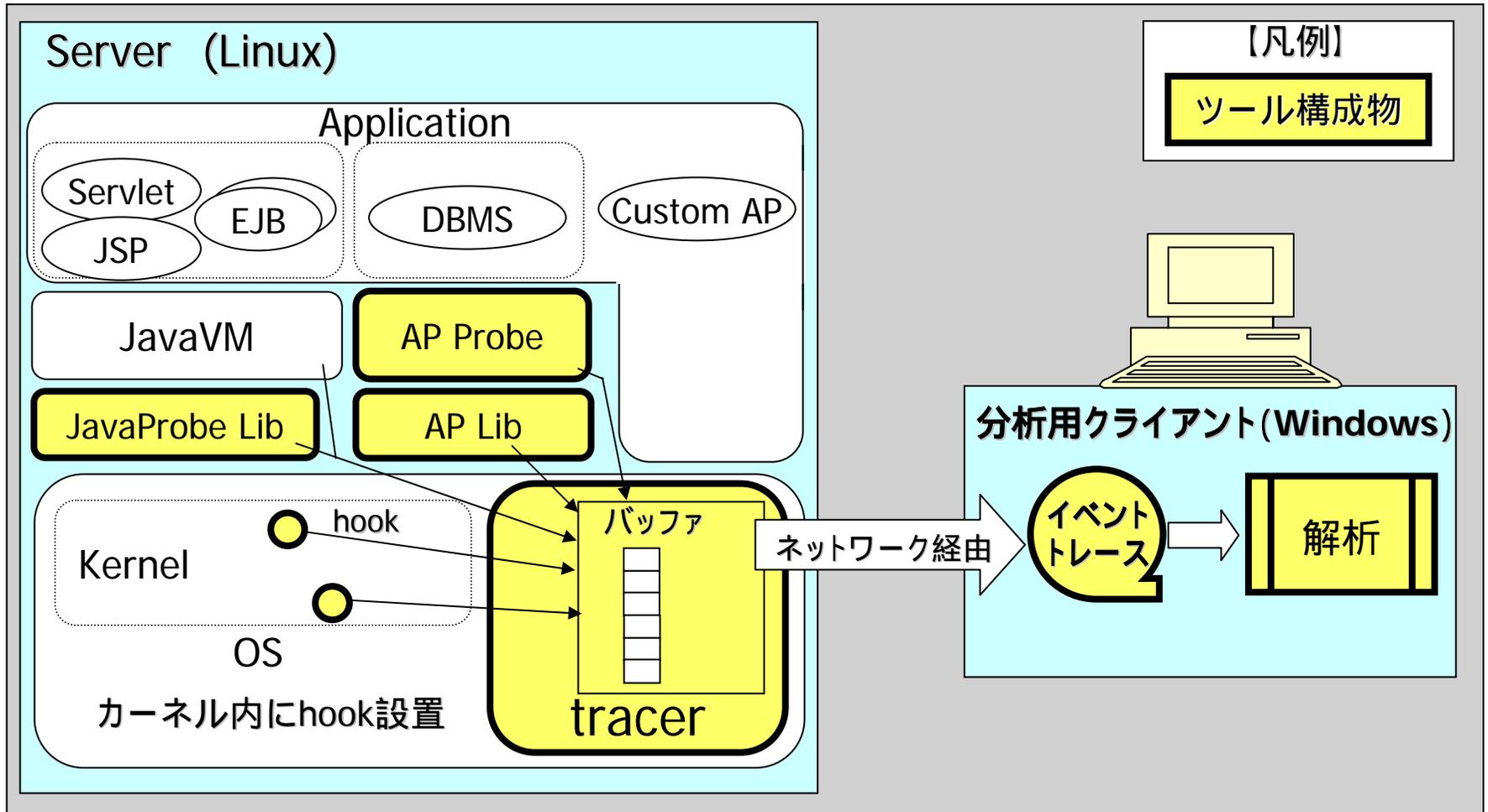


各アプリケーション(プロセス/スレッド)単位でのサーバ・リソース使用状況、アプリケーション間の連携状況を確認可能

単体、結合試験時に実装状況、性能を精密に確認可能

2.3 構成

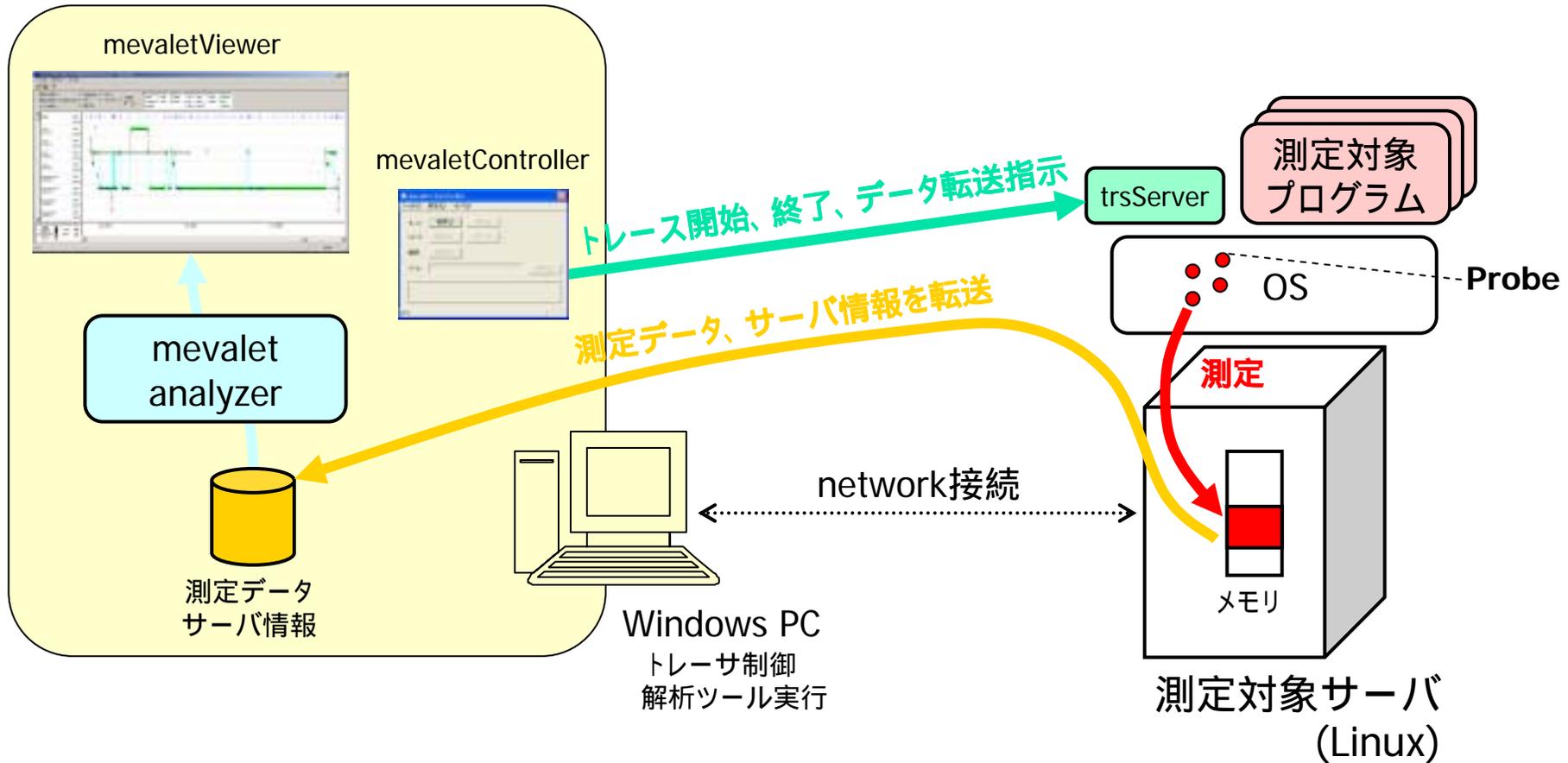
- ◆ OSレベルでイベントを検出
アプリケーションの修正が不要 (アプリケーション、ミドルウェアを選ばない)
- ◆ 解析対象: CPU, Disk, Networkの使用状況および動作状況。



2.4 測定方式の比較

	サンプリング	イベント・トレース
例	sar, ps	mevalet
実装	容易	× 複雑(要編集ツール)
オーバーヘッド	小	数%程度(CPU時間)
データ量	1データ / 数秒	× 1データ/1イベント
事象の因果関係	× 統計量のみ	把握可能
得意分野	長期間(数日～数ヶ月) の動作を概観	短時間(数秒～数+秒)の 動作を詳細分析
適所	運用段階での 動作状況監視	設計開発段階での 性能確認、問題点分析

2.5 操作



- ◆ 全操作(測定・解析)をPCからGUIで実行可能
 - サーバでは測定補助daemonを起動しておく
- ◆ 開発環境と試験環境が離れていても使用可能

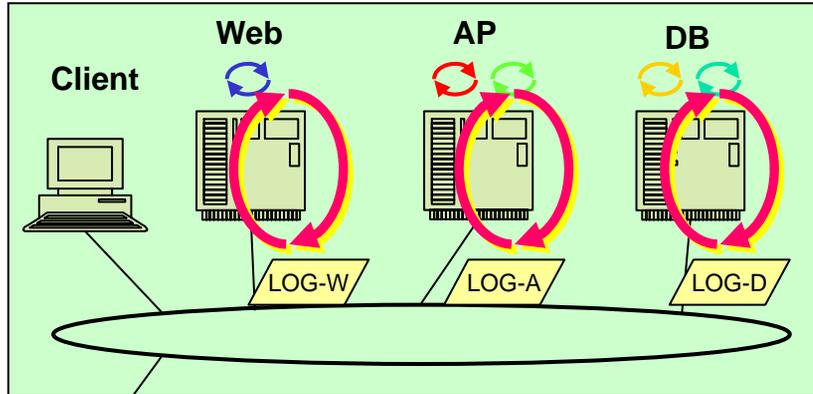
2.6 複数マシンの同時測定・一括分析

従来

多種多様なツールを使用

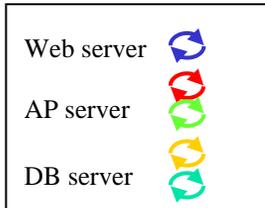


Measurement



データ形式や意味が異なる
各種ログの分析が必要

個別解析



個別マシンのリソース負荷

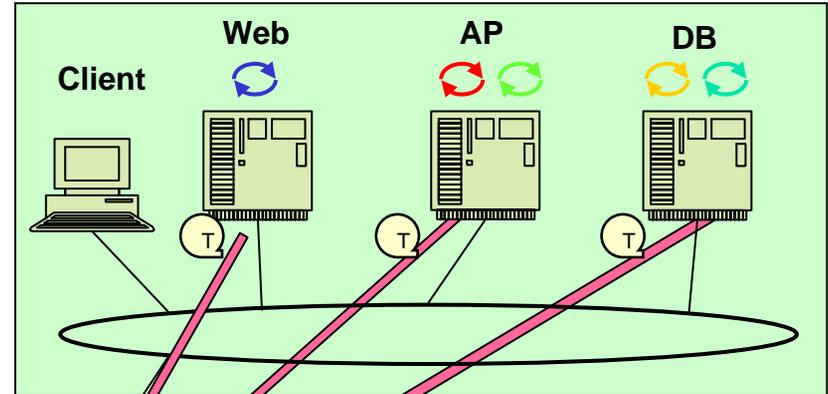
複数サーバを関連つけた分析は困難

新方式

mevalet 使用

Measurement

Trace analysis



同時測定

ツールがプロセス依存関係を把握

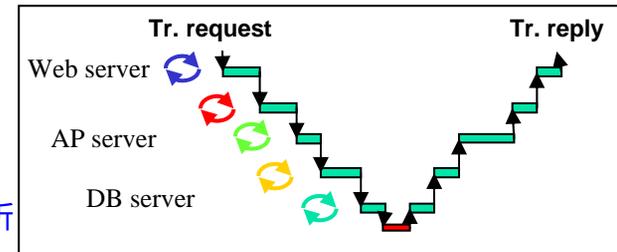


PC (mevalet)

データ集計、分析



データ一括解析



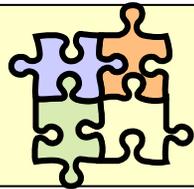
各トランザクションに着目した
分析が容易(高分解能)

3 . IPS-mevalet連携

3.1 連携効果

性能測定・分析技術

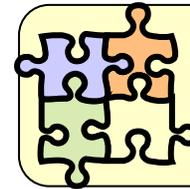
アプリケーションのモデル化
(ミクロな視点)



+

性能シミュレータ

システム全体のモデル化
(マクロな視点)

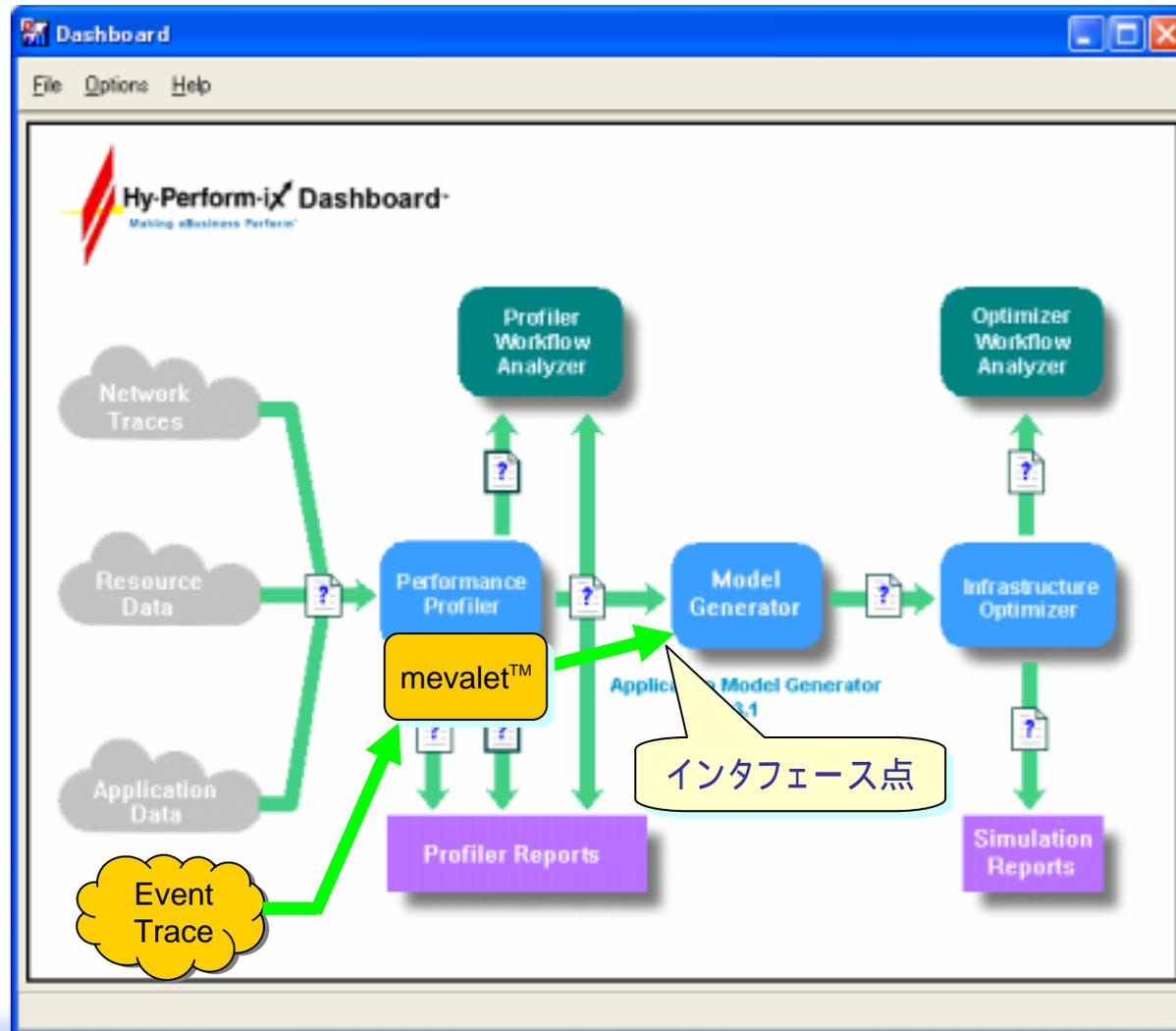


- ◆ プロトタイプ作成段階でネットワークの影響も含むシステム全体の性能設計の“失敗”を極力防止

- ◆ 現状のシステムをトラフィックデータからモデル化することで将来的なボトルネックの分析が可能となる
- ◆ 諸元変更(アクセス数増加、アプリケーションの更新、システム構成変更)対応のための事前計画時などに複数の選択肢から“誤った解の選択”を極力防止

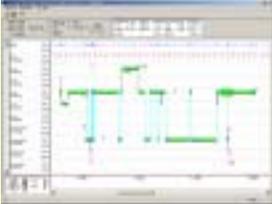
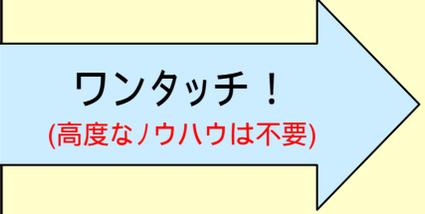
3.2 インタフェース

- ◆ mevalet: AMG用CSV形式のプロファイル(モデル)を作成
- ◆ IPS: AMGによりシミュレーションモデルを作成



3.3 シミュレーションまでの作業

◆ mevalet と Integrated Performance Suite の組み合わせで、エキスパートレベルのサイジング作業を効率的に実行可能

シミュレーションモデルの作成手順	性能測定データの収集	データ解析	プロファイリング	シミュレーションモデルの作成
<p>新方式</p> <p>性能測定・分析ツールを使用した場合</p>	<p>イベントトレースデータを簡単に採取。</p> 	<p>解析ツールで挙動詳細を掌握。</p> 	<p>ワンタッチ！ (高度なノウハウは不要)</p> 	<p>生成されたトランザクションサマリを元に簡単に作成。</p> <p>◆ Point</p> <p>一連の操作で高度なノウハウなしでも容易にモデル作成可能。</p>
<p>従来</p> <p>従来の方法</p>	<p>システムログなどの結果採取(手間大)。</p> 	<p>ログや仕様書などのデータを手間をかけて詳細解析。</p> 	<p>解析結果を元にプロファイルを作成。 (高度なノウハウが必要)</p> 	<p>プロファイル情報を元に作成。</p> <p>従来方式では、モデル作成に高度なノウハウ・経験が必要かつ時間をかける必要あり。</p>

4 . Case studies

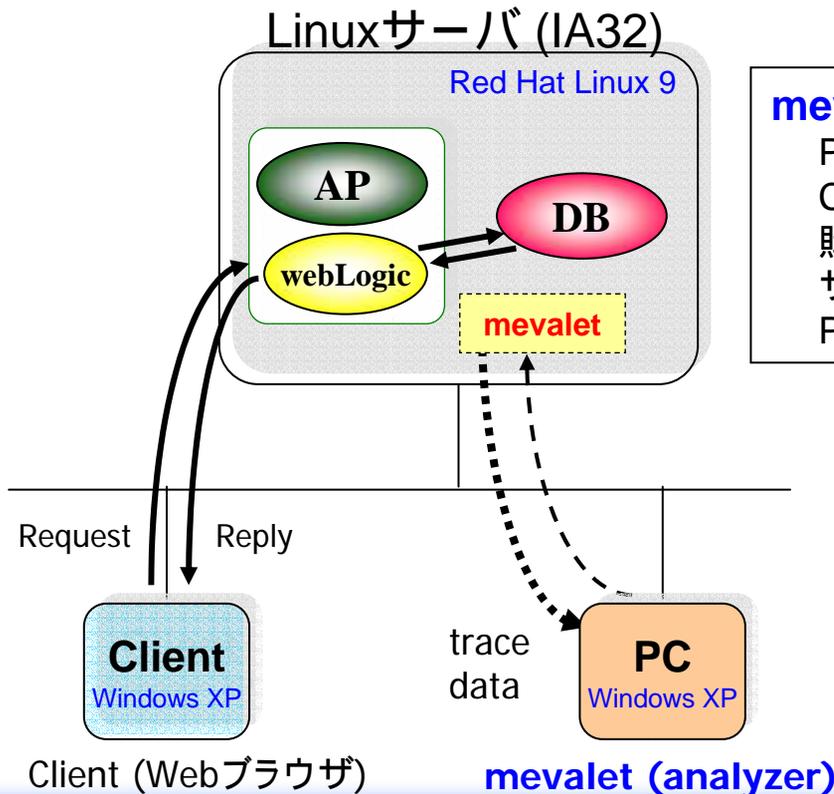
4-1-1 適用例(1/8)

◆ 目的:

単体テスト環境でのwebLogicサンプルアプリケーション(PetStore)の実測性能を元に、サーバを2台に分散した場合のシステム全体のスループット、レスポンスを見積もる。

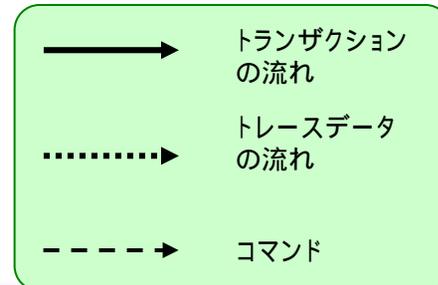
◆ 方法:

mevaletの測定データから1トランザクション当たりのリソース消費量を求め、HyPerformix IPS により、トランザクション並行処理時のスループット、レスポンスを見積もる。



mevalet によりLinuxサーバを計測

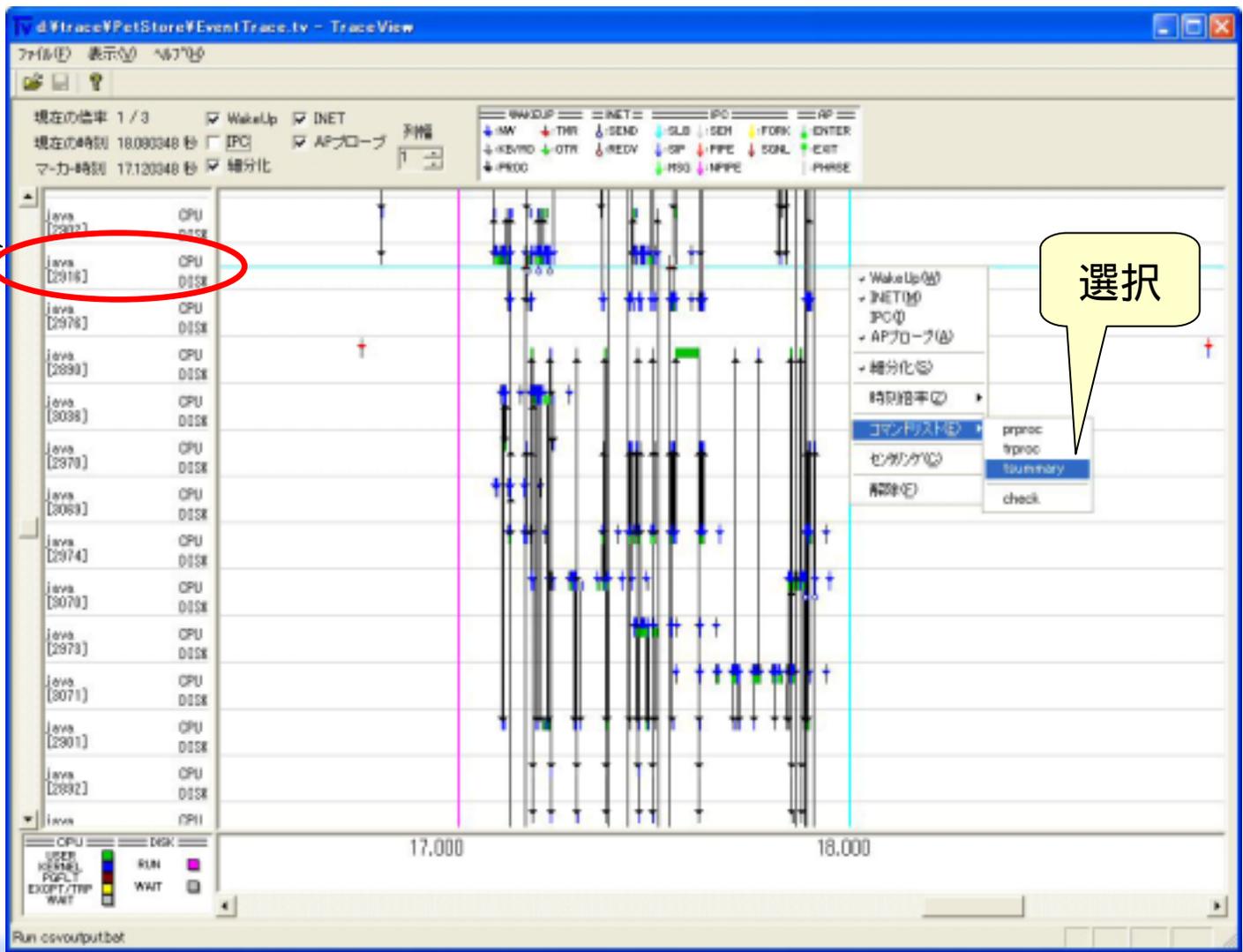
PCからLinuxサーバでのイベントトレースの採取開始を指示
Client からPetstoreアプリケーションの購入処理を実行
購入処理の終了を確認後、トレースの終了を指示
サーバから採取したデータがPCに自動転送
PCでmevalet分析結果の画面を起動



4-1-2 適用例(2/8) - トランザクションサマリーの生成

該当プロセスをマーキングした後、ポップアップメニューから tsummaryを選択します。トランザクションサマリの生成が始まります。

該当
プロセス



選択

4-1-3 適用例(3/8) - AMGによるXMLモデルの生成

mevaletで生成されたトランザクションサマリ (CSV)をApplication Model Generatorに読み込みXML Modelファイルを生成します。

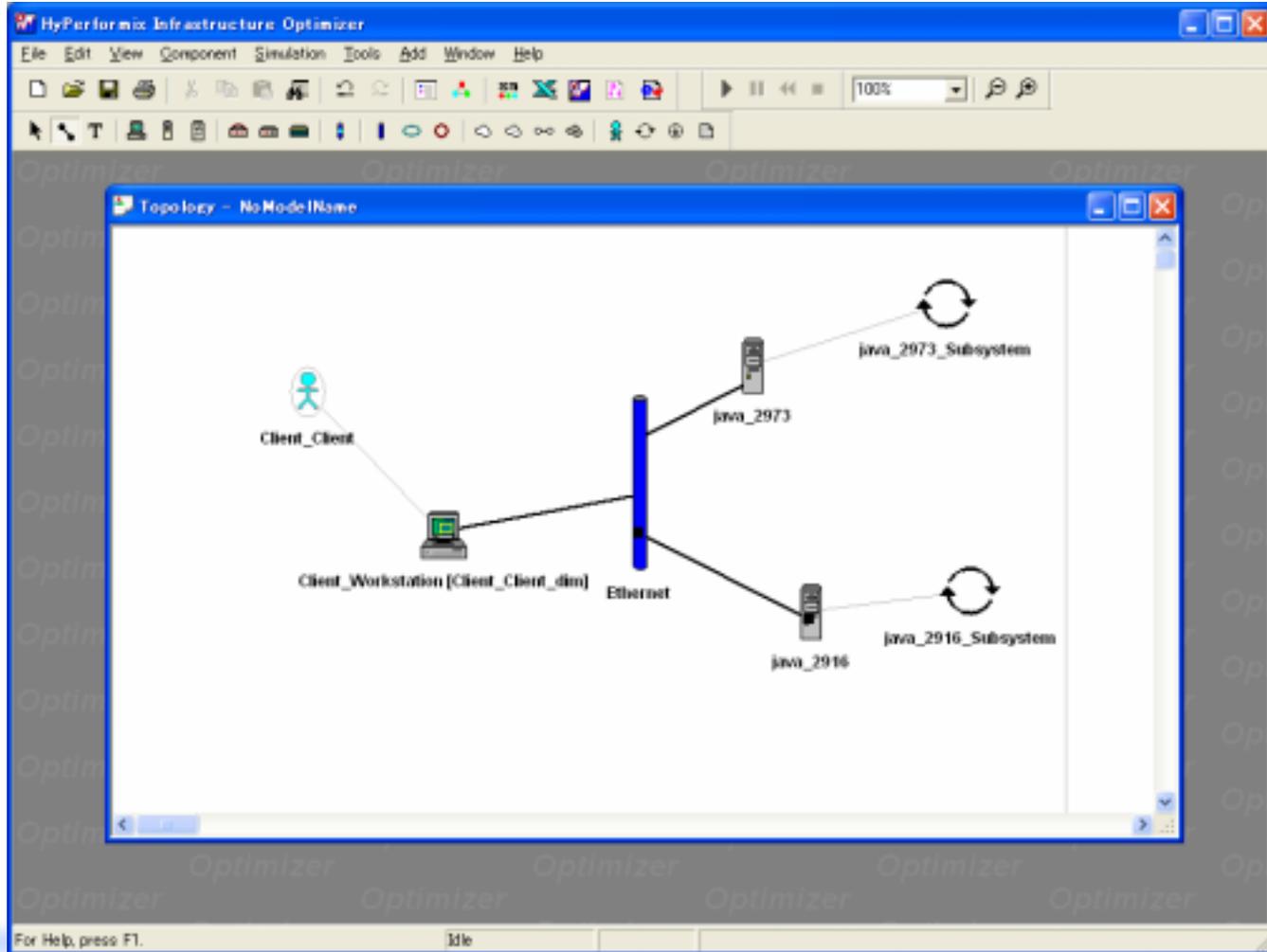
初めにメニューのOpen - Transaction report in New Profile を選択してトランザクションサマリ (CSV) を指定します。

The screenshot shows the AMG Model Generator application window overlaid on a Microsoft Excel spreadsheet. The spreadsheet displays a transaction summary with columns for Requestor, Replier, Visit Count, Transaction, Request Time, Server Time, and Local Time. The AMG Model Generator menu is open, showing options like 'New', 'Open', 'Add Transaction Report into Active Profile', and 'Generate Model Files'. A yellow callout box points to the 'Generate Model Files' option with the following text:

transaction summaryファイル (CSV) の表示後、Menuの Generate Model filesを選択してXML Model ファイルを生成します。

4-1-4 適用例(4/8) - Topologyへのネットワークの追加

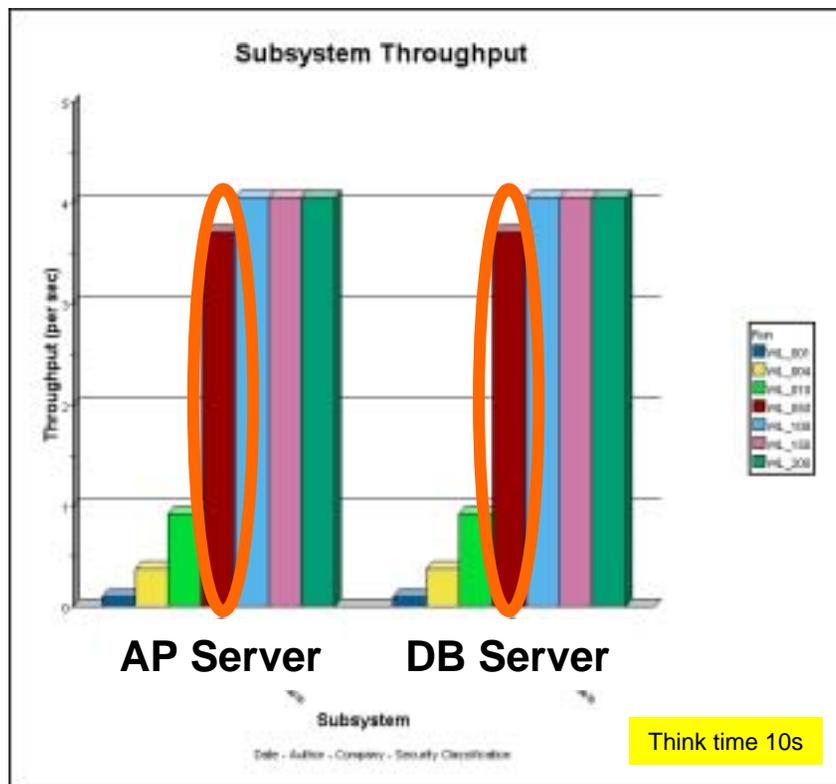
Infrastructure OptimizerにてAMGで生成されたXMLファイルをインポートします。画面にトランザクションを構成するサーバや端末、プロセスなどのコンポーネントが生成されます。LAN(スイッチやハブ)、WANなどのネットワークを追加し、各ノードを接続します。



4-1-5 適用例(5/8) - シミュレーション実行結果 1/2

◆ スループット

APサーバ - DBサーバの分散構成システムでクライアントの台数を変えて (1、4、10、50、100、150、200)、システムのスループットをシミュレーションし、グラフ出力したものである。



Server (AP、DB)



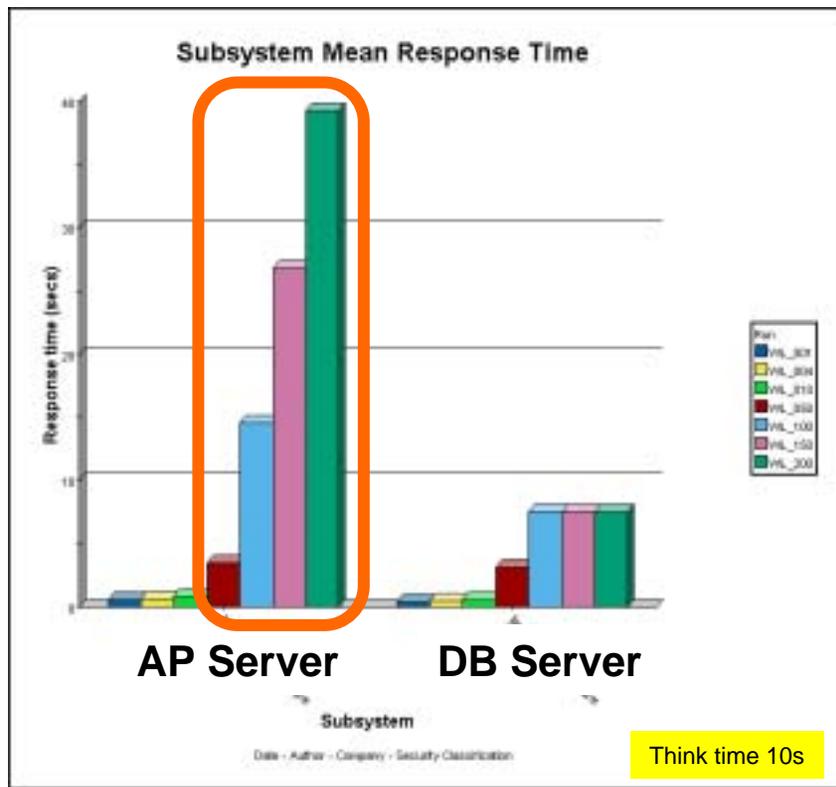
Client (PC)

Point : クライアント台数が50程度でサーバのスループットが飽和することがわかります。

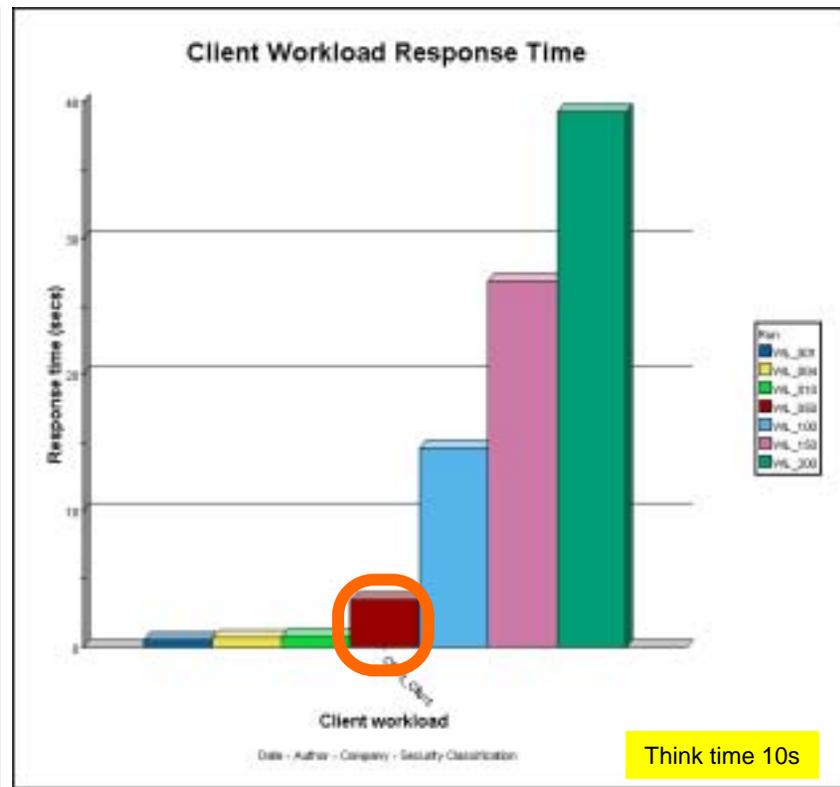
4-1-6 適用例(6/8) -シミュレーション実行結果 2/2

◆ レスポンスタイム

同条件でのAPサーバ - DBサーバの分散構成システム処理のレスポンスタイム結果のグラフ出力。



Server (AP, DB)



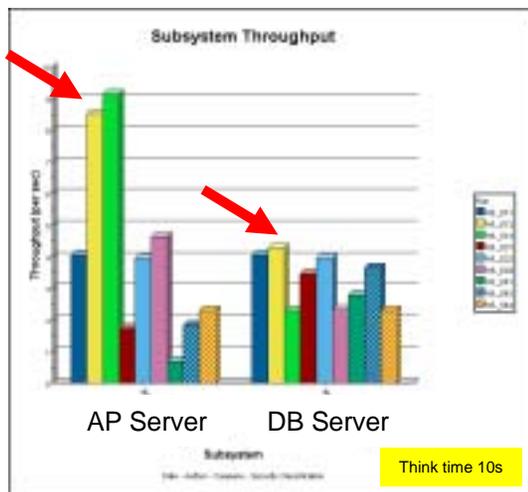
Client (PC)

Point : クライアント台数が50台以上からレスポンスが悪化し、AP Serverがボトルネックとなることがわかります。

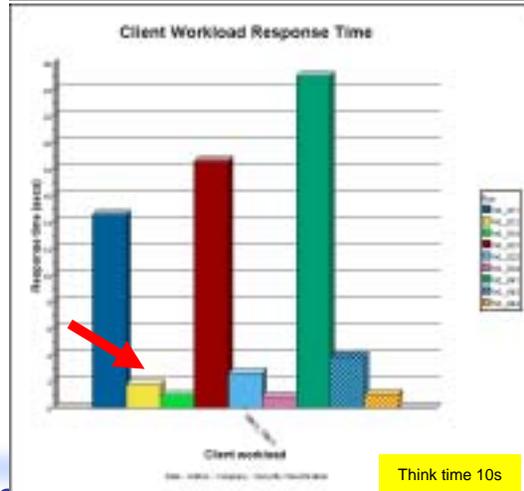
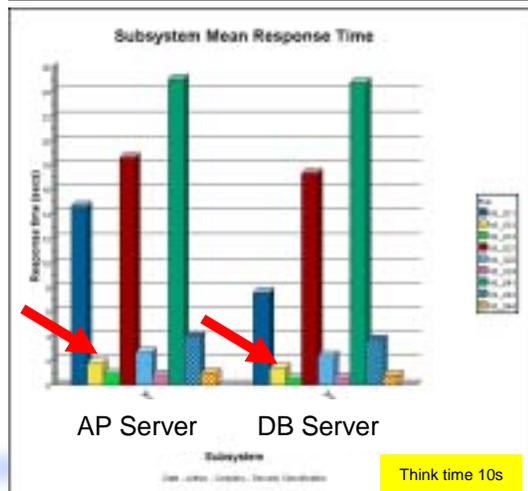
4-1-7 適用例(7/8) - 最適解の探索 1/2

◆ サーバ台数の組み合わせによる最適解の探索

APサーバ - DBサーバの分散構成システムでクライアントの台数固定で、APサーバとDBサーバの台数の組合せを変更して (1、2、4)、シミュレーションを実行 (クライアント台数100台)。APサーバ1台、DBサーバ2台の組合せで十分なスループットやレスポンスが得られることから、最適なサーバ台数の構成と想定できる。



スループット

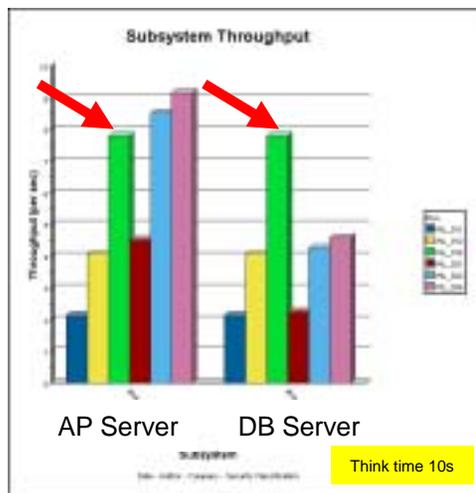


レスポンス

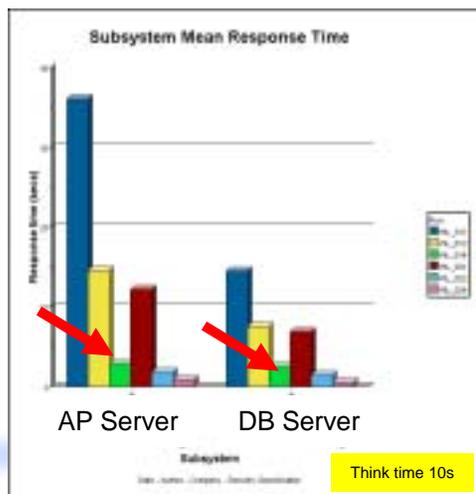
4-1-8 適用例(8/8) - 最適解の探索 2/2

◆ サーバ台数とCPU数の組み合わせによる最適解の探索

前項のAPサーバ1台、DBサーバ2台の構成を基本として、DBサーバの台数とCPU数の組合せを変更して (DBサーバ台数1、2 ならびに CPU数1、2、4)、シミュレーションを実行 (クライアント台数100台)。DBサーバ1台、CPU数4の組合せで他組合せと比較して遜色ないスループットやレスポンスが得られることから、DBサーバの最適な構成とする。



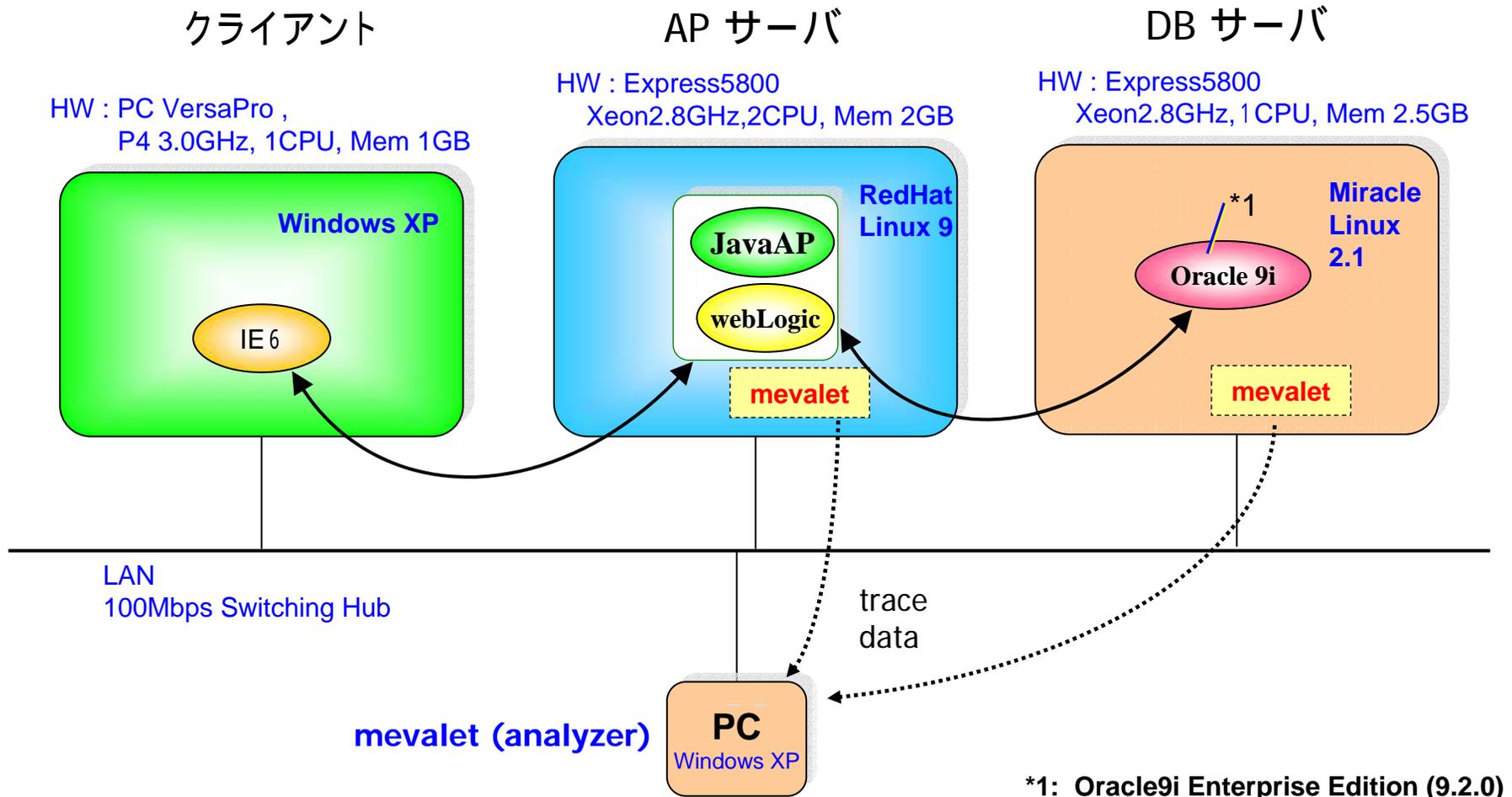
スループット



レスポンス

4-2-1 測定環境の概要

- ◆ クライアントからデータベースへのユーザ登録処理トランザクションについての性能解析を実施

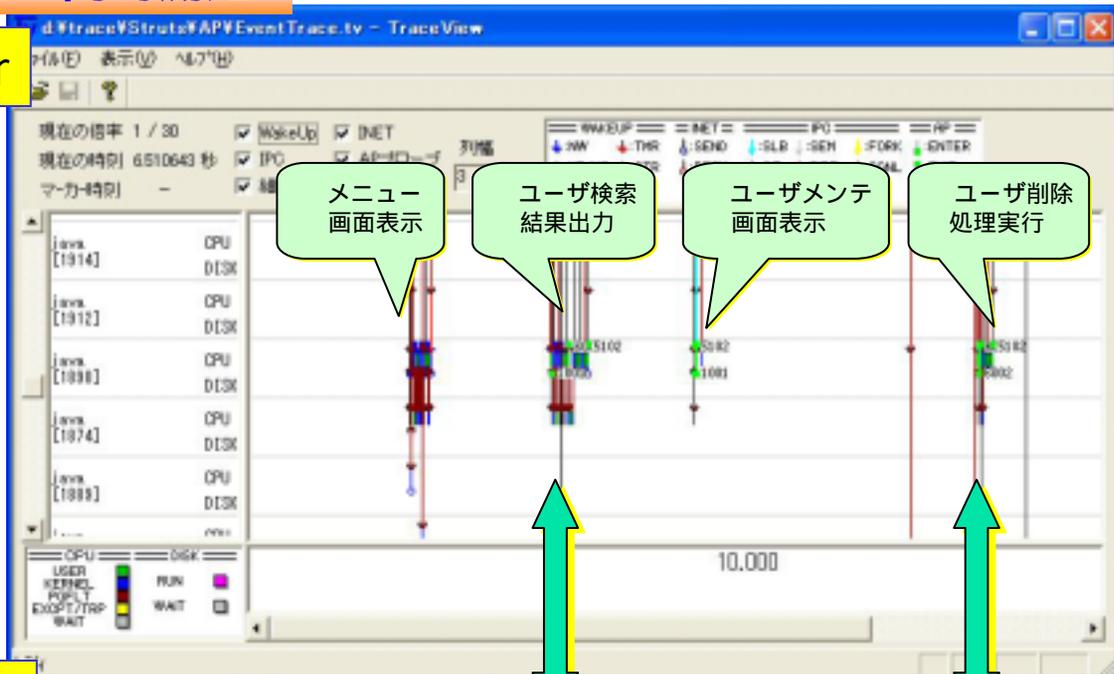


*1: Oracle9i Enterprise Edition (9.2.0)

4-2-2 測定イメージ(1/3)

複数サーバの同時測定

AP Server



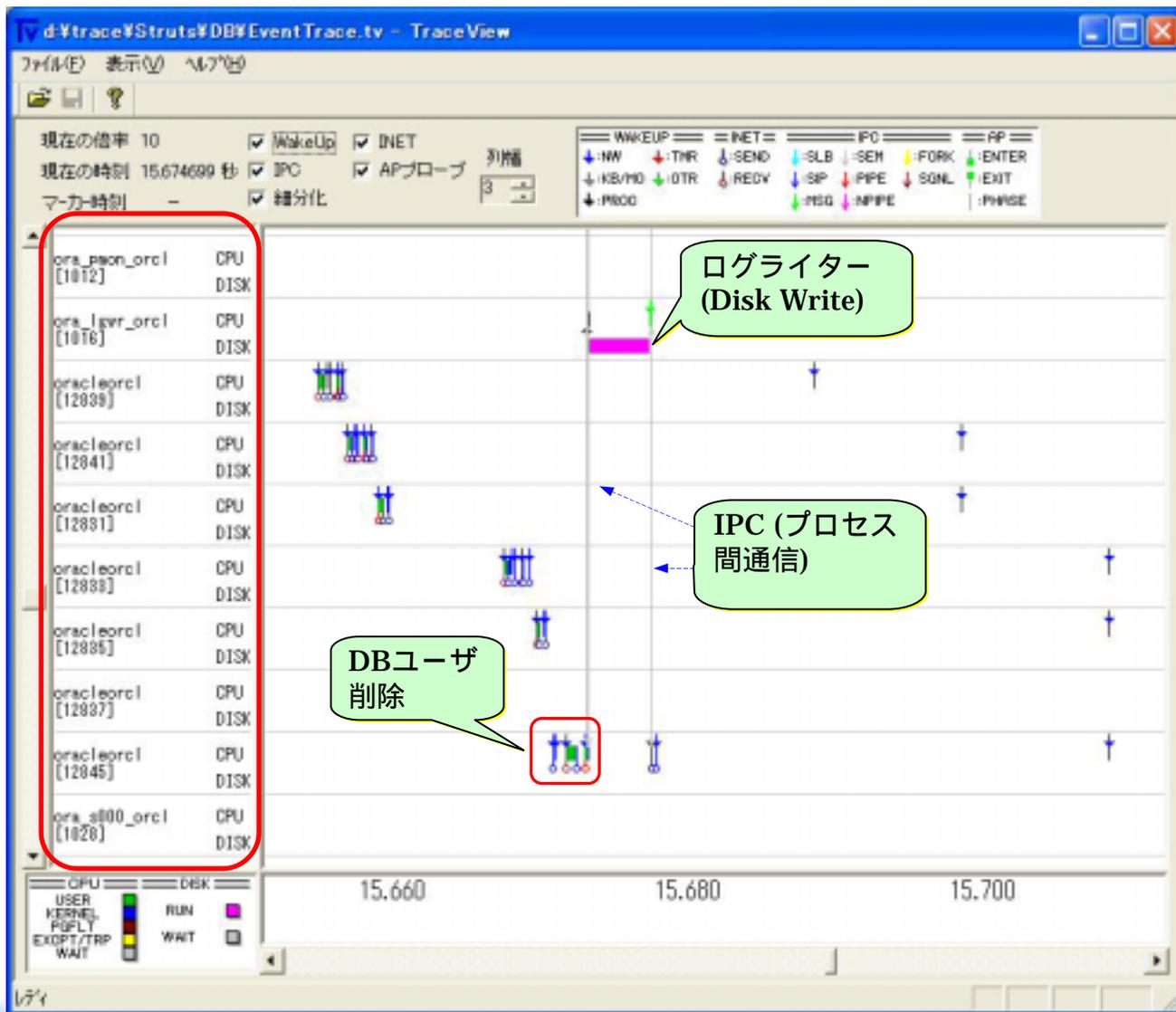
DB Server



4-2-3 測定イメージ(2/3)

DBサーバ (詳細)

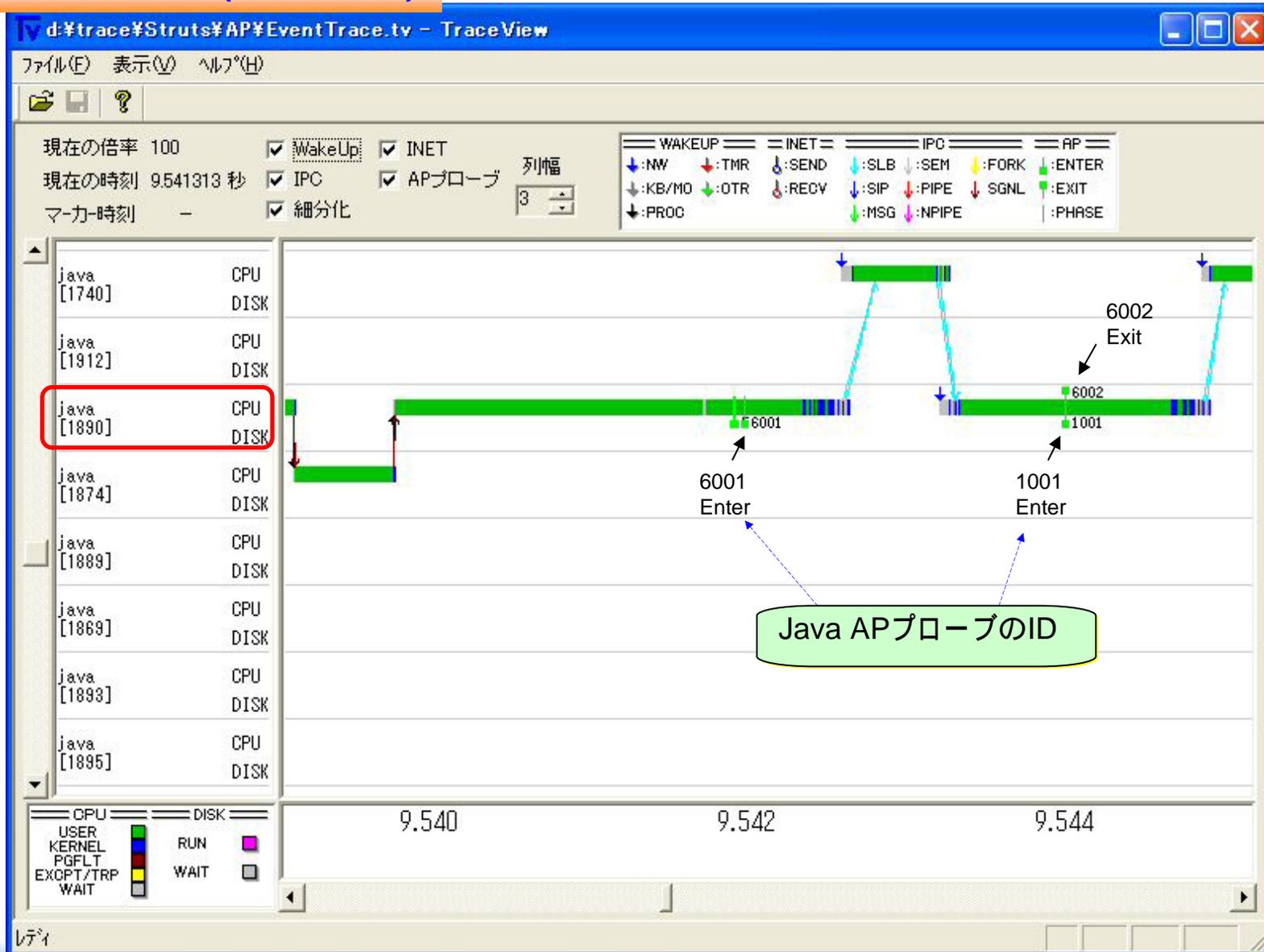
Oracle
プロセス



4-2-4 測定イメージ(3/3)

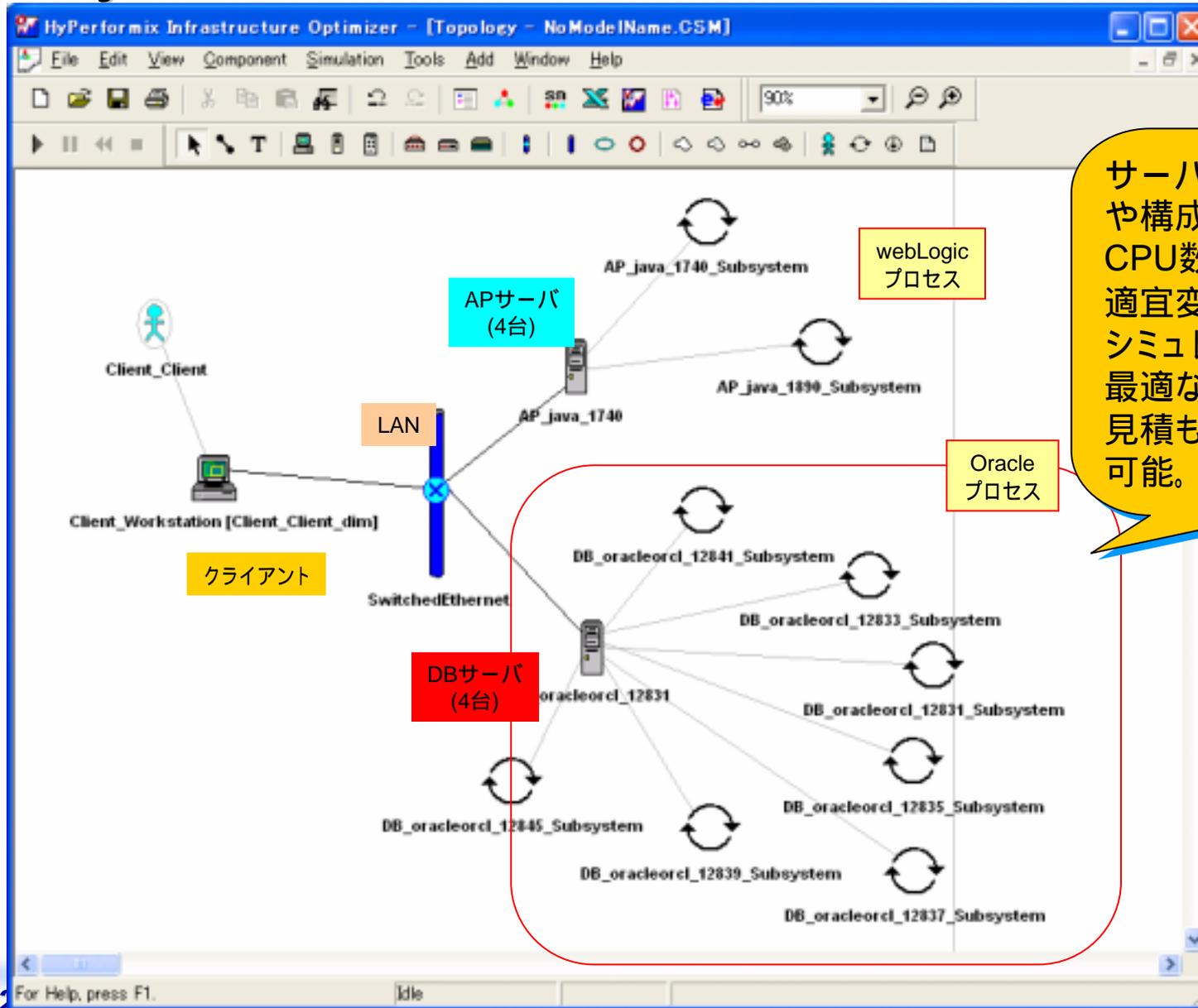
Java AP プローブ(APサーバ)

メイン
プロセス



4-2-5 シミュレーションモデルのイメージ

◆ HyPerformix IPS上にシミュレーションモデルを作成



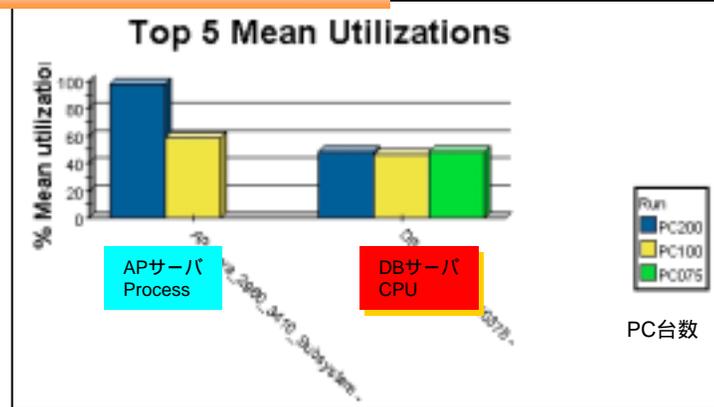
サーバ、クライアントの台数や構成機器の緒元(モデル、CPU数、メモリなど)などを適宜変更して、シミュレーションを行い、最適なシステム構成の見積もり・探索を行うことが可能。

4-2-6 シミュレーション結果の例

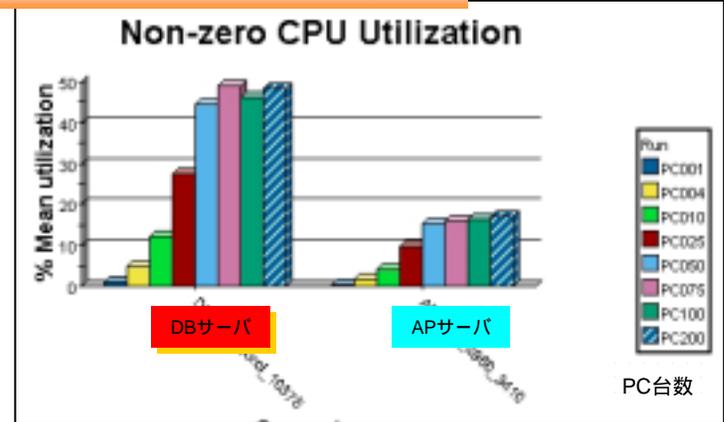
- ◆ ユーザ登録処理トランザクションを1,4,10,25,50,50,75,100,200 と変動した場合のトランザクション処理時間をシミュレーション
 - APサーバがボトルネックのため、APサーバ台数を4台から8台に増設して、同一条件で実行したAPサーバの負荷が低減されたことを確認

改善前

リソース使用率(Top5)

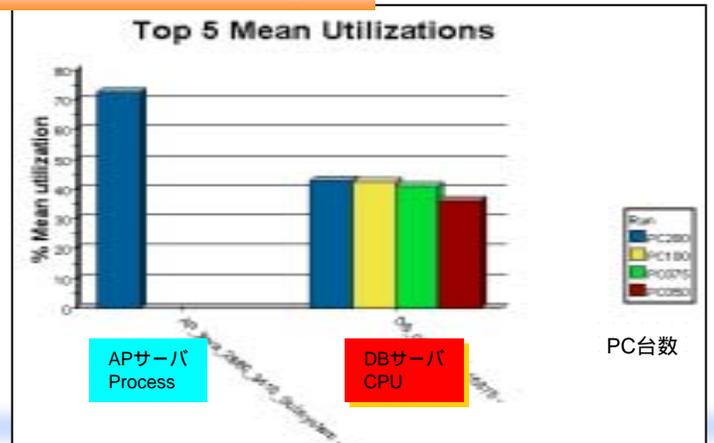


CPU使用率(サーバ)

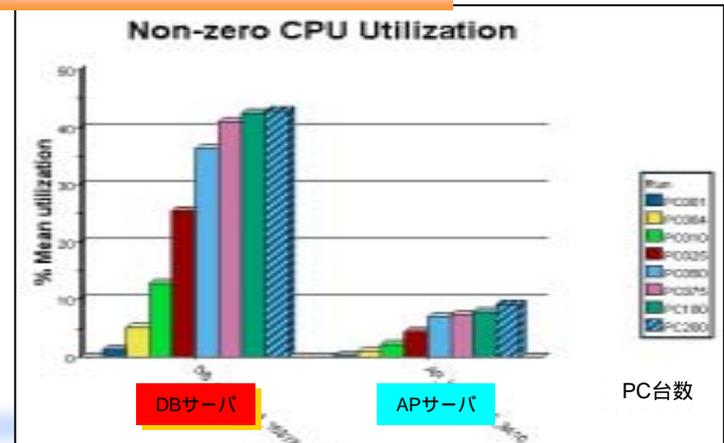


改善後

リソース使用率(Top5)



CPU使用率(サーバ)



5. 利用局面

5.1 システムのライフサイクルと性能評価

定量的な性能評価に基づいたシステム移行
種々のシステム構成・機能配置を比較・検討

複数トランザクション処理時の不具合分析
予測結果との差異を分析

単一トランザクション処理の性能確認
コンポーネント毎のリソース消費分析

Capacity planning
開発初期からの高精度見積り

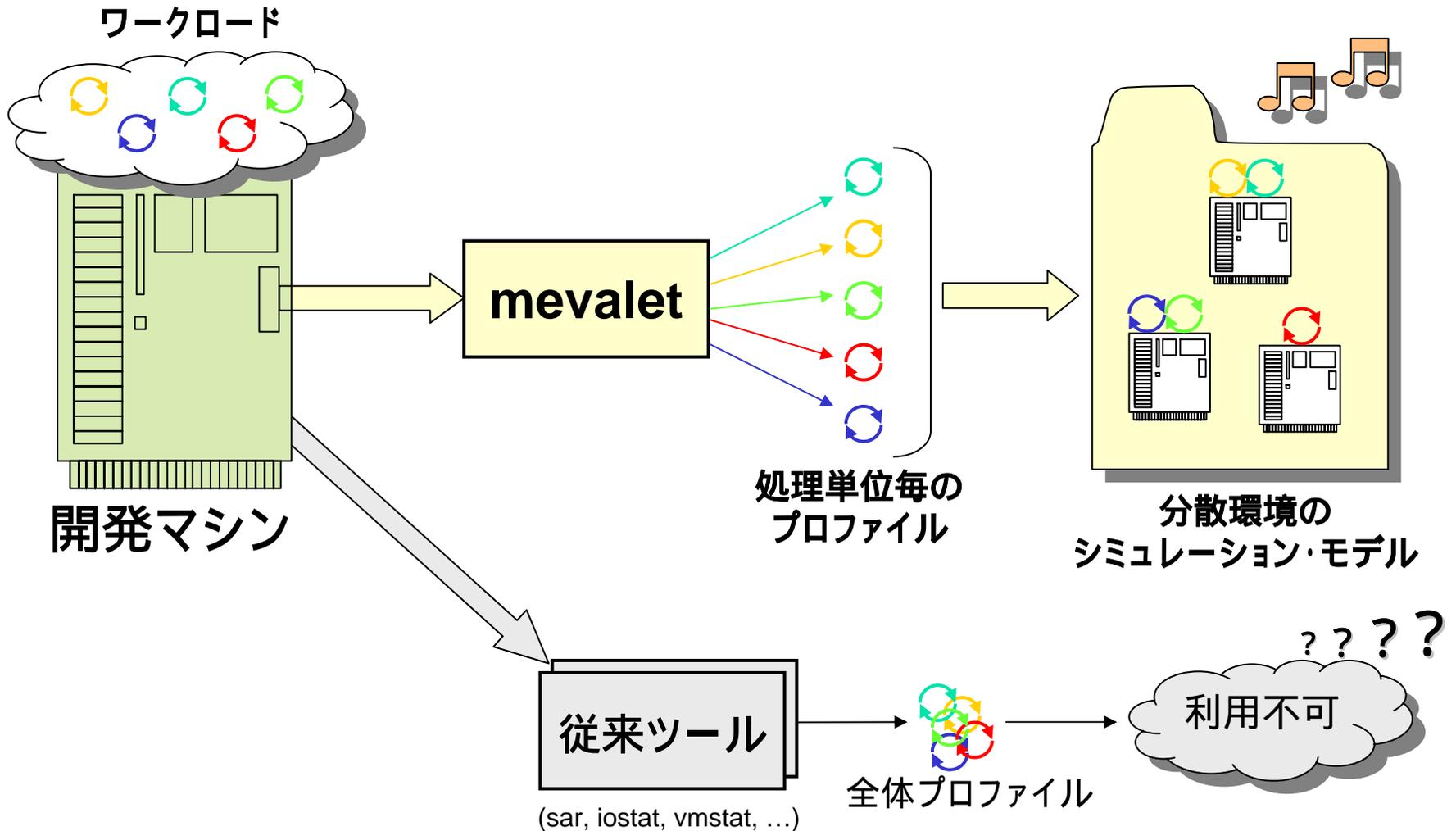
プロトタイプ

開発システム

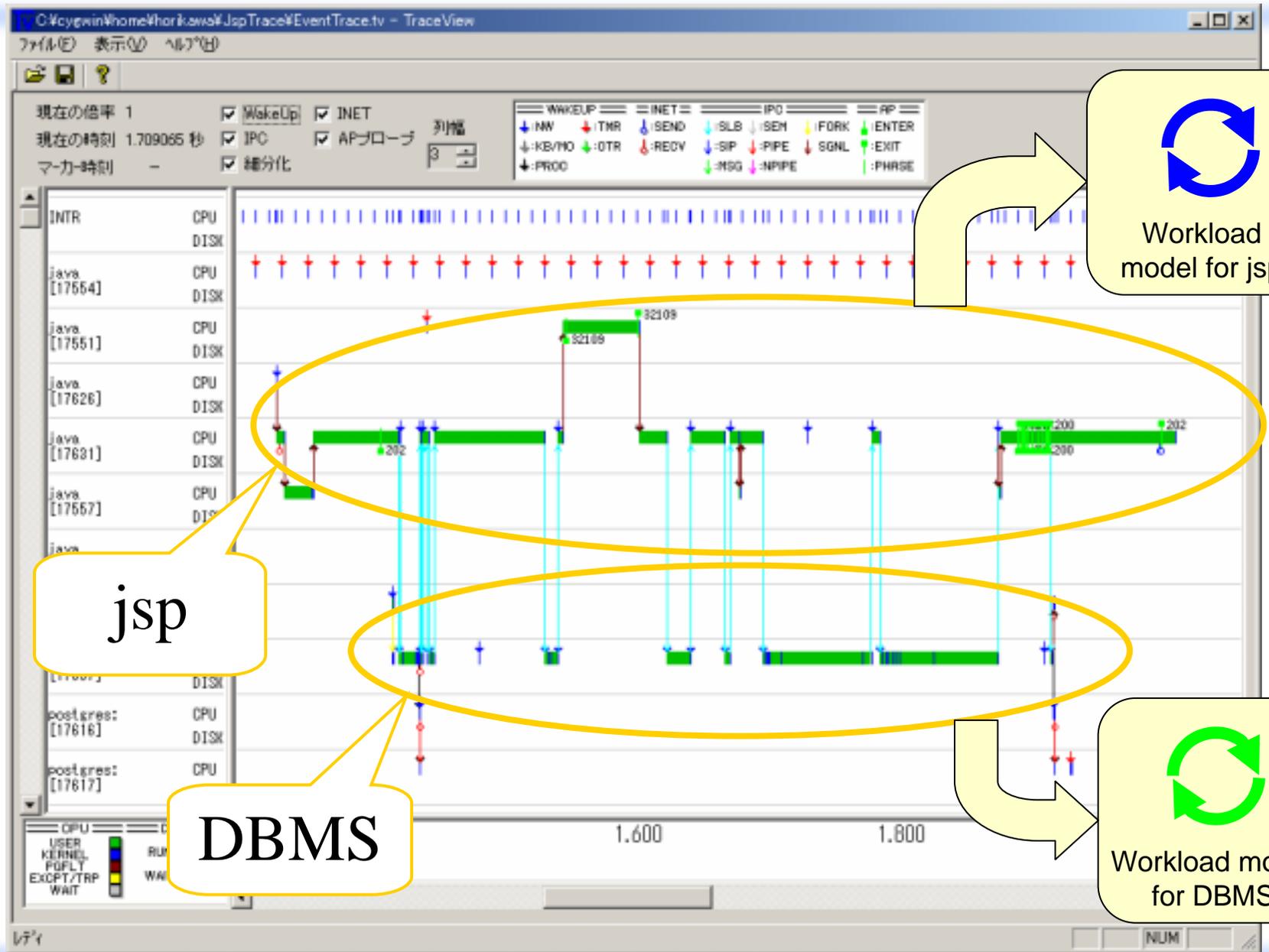
設計 コーディング 単体テスト 結合テスト 実運用

フェーズ

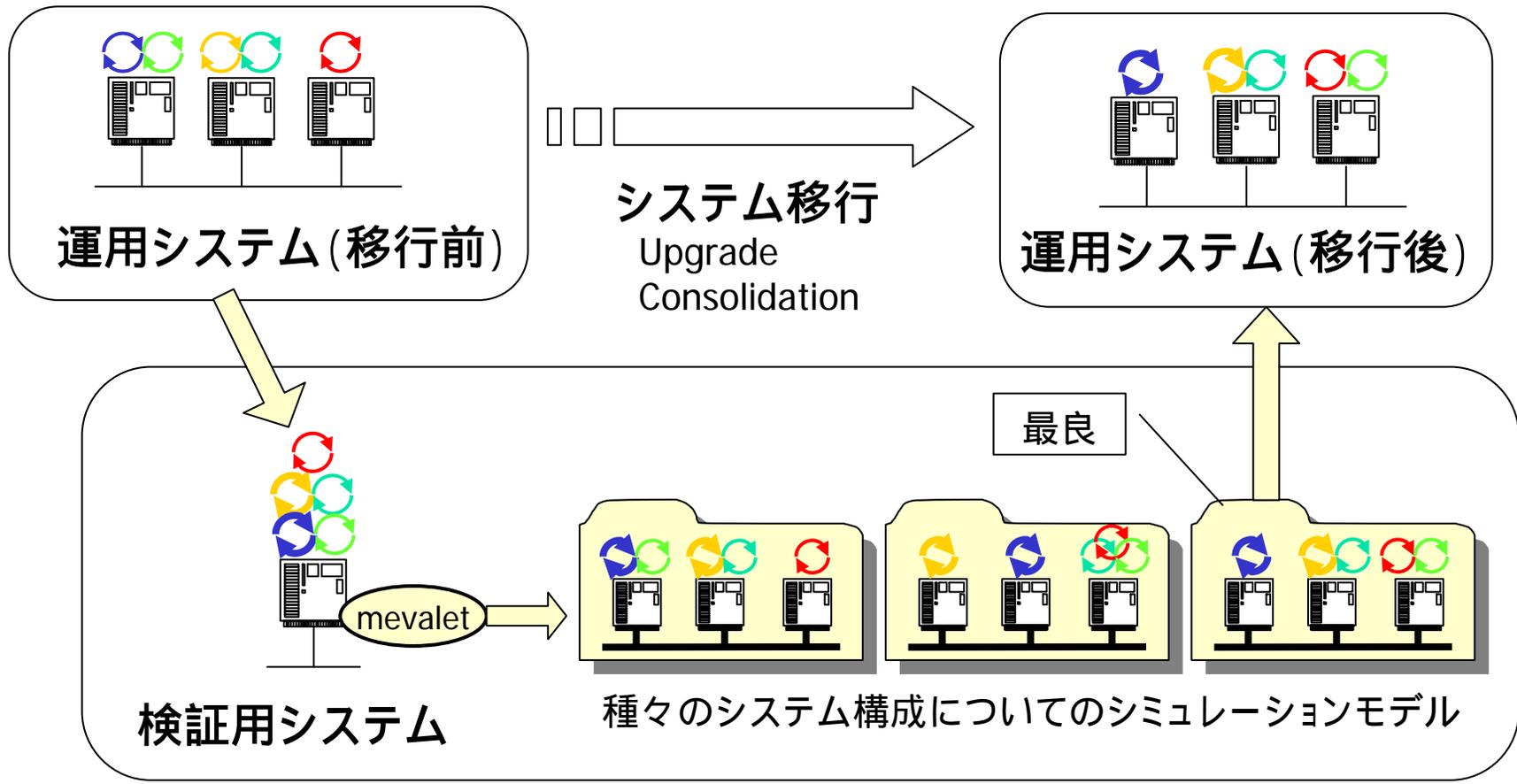
5.2 処理単位毎のプロファイル分析(応用)



5.3 処理単位毎のプロファイル分析(具体例)



5.4 システム移行時のコンサルテーション



定量的な性能評価に基づいたシステム移行をサポート

6. まとめ

◆ 性能測定・分析技術とシミュレータの連携効果

- ミクロな視点(アプリケーションのモデル化)

+

- マクロな視点(システム全体のモデル化)

◆ IPS + mevalet

- プロファイリング、シミュレーションモデル作成を簡略化
エキスパートレベルのサイジング作業を効率的に実行可能

E.O.F

Empowered by Innovation

NEC