

コンピュータ・システムにおける パフォーマンス管理

小幡 一郎

1. はじめに

最終電車に乗り遅れてしまい、首都高速をタクシーで家路につく途中、大渋滞に巻き込まれてしまった。渋滞の要素となっているのは大量のタクシー群であり、大渋滞の要因を作り出したのは1台の故障車であった。この場合、1台の故障車は「余分なタクシー代」という経済的効果(?)を首都高の流れを止めることで一瞬にして発生させたといえる。

本稿はコンピュータシステムのパフォーマンスについて解説したものである。企業にとっては、システムのパフォーマンスがあらゆる場面で経済的な影響を及ぼす。(たとえば、戦略的位置づけで使用されるリアルタイム・システムにおいて、あるいは劣悪なスループットが生み出す人件費への影響、等々。)

そこで、弊社が開発したソフトウェア・プロダクトー Performance Insight-のノウハウを通して、システムのパフォーマンスについて、その管理のための技法について焦点を当てて解説する。なお、Performance Insight は、UNIX システムや、Windows-NT 上で稼動する Oracle データベースのパフォーマンスを管理するためのプロダクトであるため、所々データベース・システム用語を使用せざるを得ないことを了承願いたい。

2. パフォーマンス管理：一定のレスポンスを維持する

システムを利用するエンドユーザから見た応答時間(レスポンス)を一定に保つことがパフォーマンス管

理の目的である。システムを管理する側は、エンドユーザに対して、サービスできるレスポンスをコミットした「レスポンス契約」を取り交わすべきである。また、エンドユーザは、その契約にもとづき、コンピュータ・コストを支払う……。

本来であれば、このレスポンス契約のように、コンピュータを利用する側が主導権を取り、受けたサービスに対するコストがシステム部門に支払われるべきであるが、コンピュータを操るシステム部門の人間は、利用者にとっては、まるでマジシャンのような、難解なコンピュータの「操り手」に見えてしまう。したがって、多くの企業では、システム部門が中心となったパフォーマンス管理となり、レスポンスに対し、特に経済的影響に対するシビアな見方ができていないのではないだろうか。

レスポンスを一定に保つためには、ある時間間隔内で流れるトランザクションを制御することが必要である。それを比喩的に表現したものが図1である。三角堰による放水量をバルブで制御する様子を図解したものだ。

水理学では、放水量を制御するためのパラメータを、放水される水が通過する三角堰の水位(深さ)ごとの経験値から導き出す。水位が変われば、制御のためのパラメータも変わる。理論だけでは現実の三角堰は制御できない。コンピュータ・システムについても同様のことがいえる。実システムを完全に記述し、制御できる理論はない。そこに必要になるものは、実システムを定量的に、客観的に表わした経験値(実績値)と

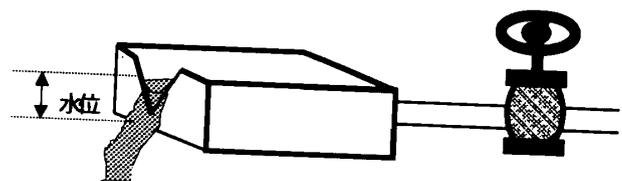


図1 放水量と三角堰

なる。

コンピュータ・システムにおいては、図1の水量に相当するのがトランザクション量である。バルブの調整は一定時間に処理するトランザクションを「実績値」から予測することに相当する。すなわち、パフォーマンス管理となる。

3. パフォーマンス評価の対象となるもの

パフォーマンス評価には「データベース・アーキテクチャ上の評価」と「システム・アーキテクチャ上の評価」という2つの側面がある。

多くのサイトで試みられていることはデータベースのパラメータを手探りで変更し、レスポンス向上を図ろうというものである。一方、後者のシステムの観点からのアプローチはなされていない。Oracle データベースには多数のパラメータがある。ユーザはパラメータの多さに振り回されてしまい、コンピュータ・システムのバランスにまで神経が回らないのが実状ではなかろうか。

もし、データベースのバッファ・キャッシュを増やすことにより、メモリ不足によるページングを誘発するようなことになったら、チューニングのための変更がすべてを台無しにしてしまう。

4. システム・アーキテクチャの観点からパフォーマンスを評価する

どのシステム資源がボトルネックになっているのか、あるいは、なり得るのかを知る——これがシステム・アーキテクチャ上の評価の目標である。システムの評価を行うには、まず「システムのバランス」を評価することが大事である。

図2は、コンピュータ資源の各要素を图示したものである。この中の、メモリ、ディスク I/O、CPU、ネットワークに対する負荷評価を「システムの4大要素」として、ボトルネック分析の指標とする。

Performance Insight では図3のような、トランザクション数 (X 軸) と4大要素の使用状況 (Y 軸) の相関を取り、パフォーマンスに影響を与えている要素を探り出す。先にも触れたが、メモリ不足が起きているサイトでは、安易にデータベースのバッファ・キャッシュを増やすようなことはできないからだ。図3の対角線上に振られたドット上にサンプリング・データであるアスタリスクがすべて乗っている状態が「完全相関」の状態となり、ピアソン相関係数は1を示す。

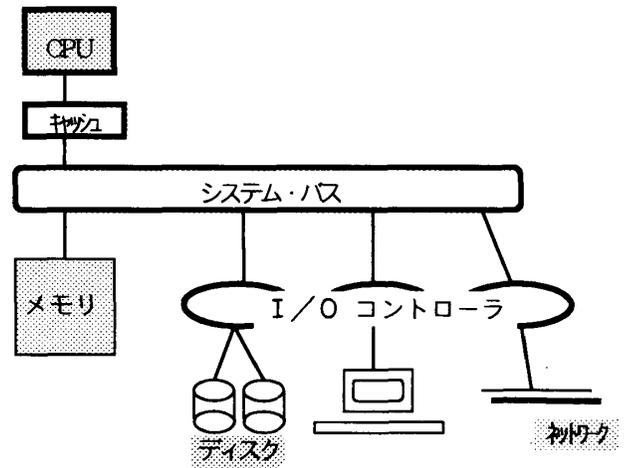


図2 コンピュータ資源

4大要素のうち1に最も近い要素が、パフォーマンスに最も影響を与えるであろう要素となる。また、数日分の相関係数のうち右肩上がり度で1に近づいている要素が今後のパフォーマンスを左右する要因になり、逆に右肩下がりの要素が他の要因により影響を受けている資源となる。

ある要素がシステムのボトルネックとなり始めると、他の要素の足を引っ張り始める。たとえば、十分すぎるCPUパワーを持ったマシンで、貧弱なディスク構成のシステムを動かすと、ディスク I/O が限界 (スレッッシング・ポイント) を向かえ、CPU パワーが出し切れなくなる。図4はそれを表わしたものだが、チューニングとはある意味で「スレッッシング・ポイントを1カ所に集める」ことと言える。

したがって、データベースから見たチューニング・ポイントもスレッッシング・ポイントを向かえている要素に影響のある項目から洗い出すべきである。

図5では、数日分の相関度の変化を Performance Insight のキャパシティプランニング・ツールで分析した結果である。

チューニング作業を行った8月17日以前はCPUがボトルネック要因の代表であったが、CPU 負荷の高い処理 (この場合は、高負荷なストアード・プロシージャ) に修正を加えたことにより I/O がボトルネック要因の代表となった。パフォーマンス・チューニングの世界ではしばしば「ボトルネック」という用語を使うが、この場合は「パフォーマンスに最も影響する要素である」という意味で使用している。なお、8/29の急激な変化は、ネットワーク・プロダクトの障害により、他の要素 (資源) が「足を引っ張られた」ためである。

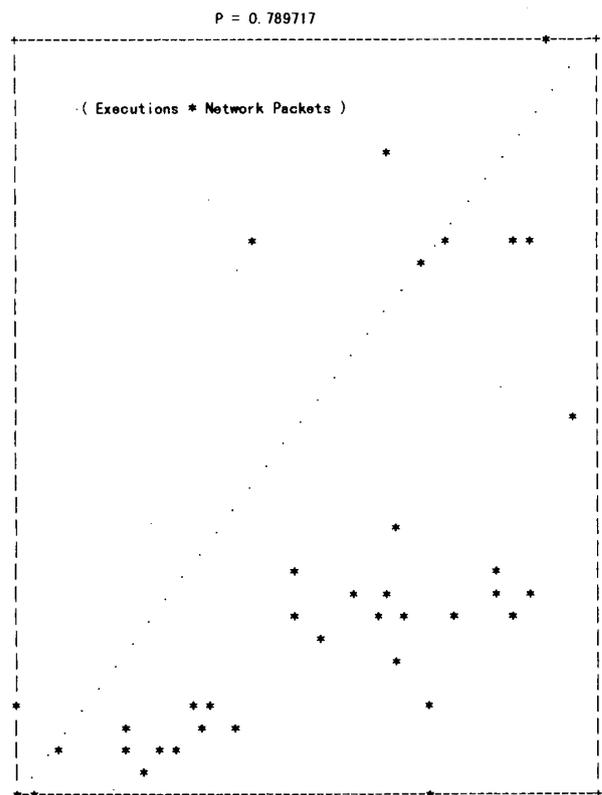
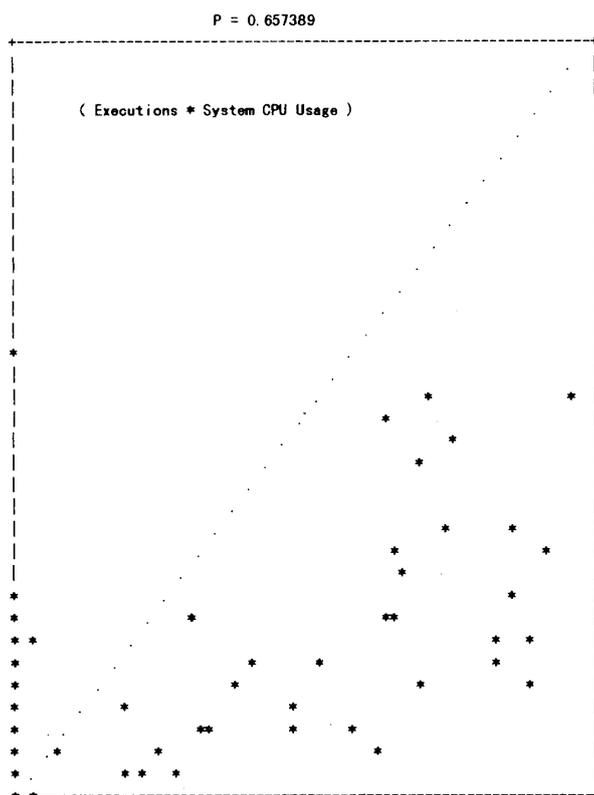
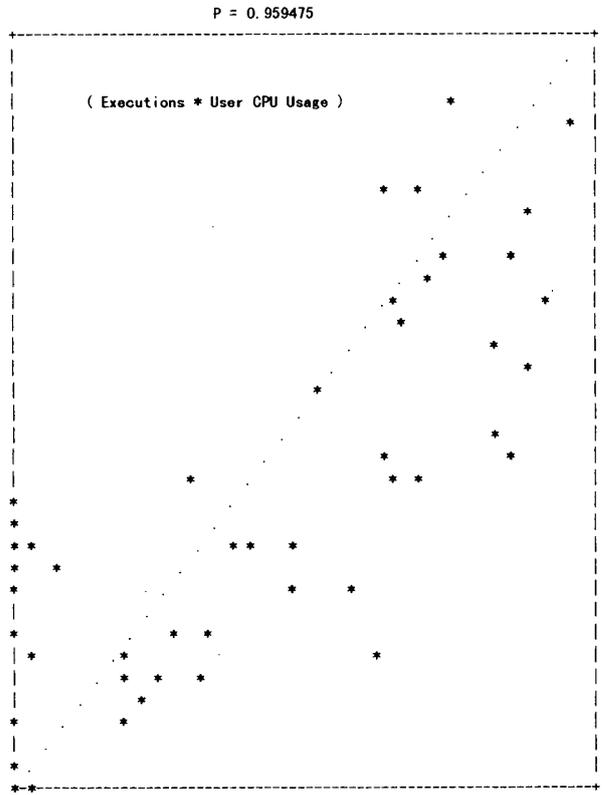
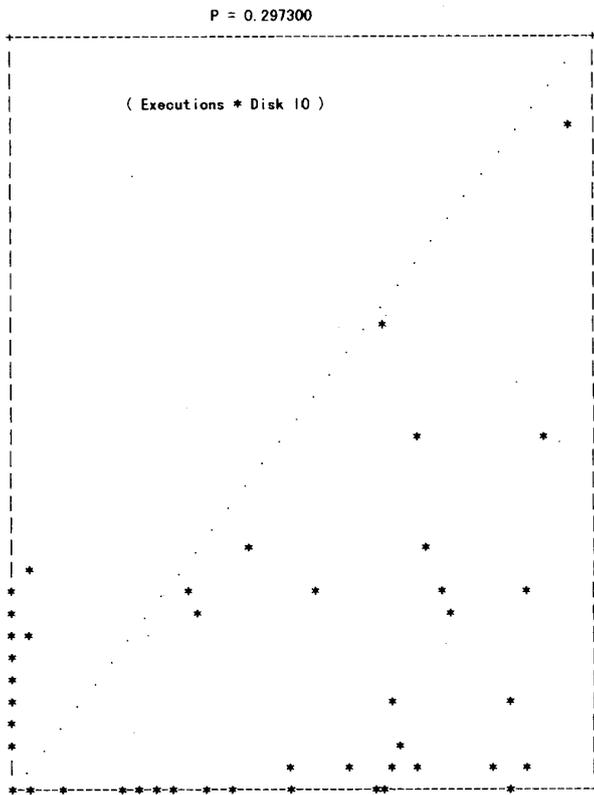


図3 トランザクション数と4大要素の相関

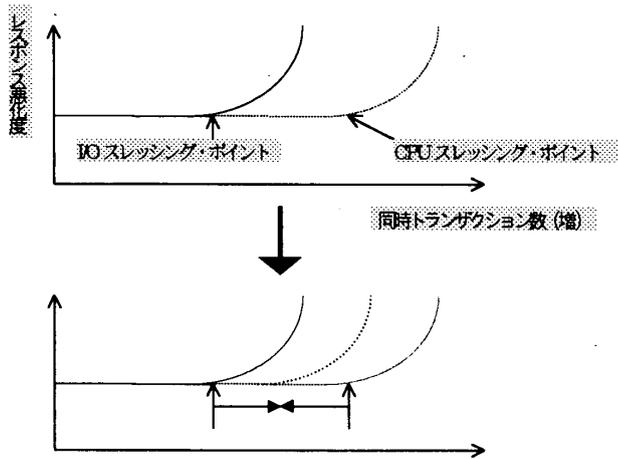


図4 スレッシングポイントとレスポンス悪化

ボトルネック相関チャート

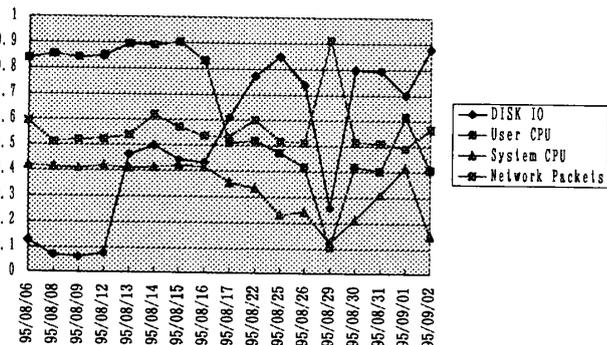


図5 数日分の相関度の変化

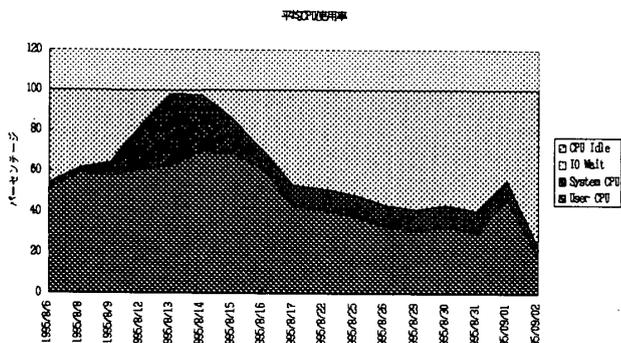


図6 チューニング後のCPU使用状況の変化

5. EPS レスポンス指標

「このマシンは限界に達しているのか」、あるいは「いつ限界に達するのか」を予測することは非常に重要である。そのために欠かせないのが、システムのレスポンスを数値で客観的に表わす評価尺度である。

評価尺度には様々なものがある。たとえば、データベース分野で有名な TPC ベンチマークテストでは、1秒間に処理するトランザクション量を tps (Transaction Per Second, トランザクション数/秒) という単位で表わす。だが、実際のサイトでは、この tps 値通りにマシンの性能を引き出せているのであろうか。大い

に疑問である。

実際のトランザクションは TPC-A/B ベンチマークで規定されたトランザクション (銀行のオンラインシステムを想定したもの) とは異なる。実際の運用のための尺度としては適当でない。TPC ベンチマークで得られる tps 値は、あくまで各種のプラットフォームを同一条件で比較する目的で測定する特殊な条件下での数値なのである。

では、どのような評価尺度が適当なのであろうか。Performance Insight では、システムの限界を予測するための評価尺度として EPS (Executions Per Second) という評価尺度を規定している。EPS とは Performance Insight のロギング機能がシステムの 4 要素 (CPU、メモリー、ディスク IO、ネットワークパケット量) をサンプリングする間隔の内、同時に動いていた SQL 数 (トランザクション数) をカウントし、それをサンプリングに費やした秒数で割ったもので表現される。サンプリングに費やした時間とは固定の 30 秒にその時のレスポンス (サンプリング・オーバーヘッド) が加算されたものとなる (図 7)。

一定間隔でサンプリングした、SQL 数と EPS をそれぞれ X 軸と Y 軸に設定し、散布図にして見ると、サイトの状態を、図 8 のような 3 つのケースに分類することができる。

レスポンス・ダウンを定量的に示すには……散布図からピアソン相関係数をとる

限られた一定時間内に処理されるトランザクション (SQL) の数が限界に達したときにレスポンスはダウンする。そこで、「1秒間に処理される SQL の数 (EPS) が横ばい状態になっているか？」を散布図 (相関関係図) からピアソン相関係数を求めることにより導き出すことができる。ちなみに図 8 の例では、それぞれの係数は以下ようになる：

ケース 1 = 1.00, ケース 2 = 0.825, ケース 3 = 0.640

ケース 2 はレスポンス悪化が起きはじめている状況であり、ケース 3 ではスレッシング・ポイントの発生により EPS ダウンが起きている状況である。

Performance Insight では、この相関係数を「EPS レスポンス指標」と規定し、システムのレスポンスを評価する指標としている。

図 9 は、常時 100 人程度のユーザが使用している受発注システムの EPS 散布図の実例である。一見、EPS の

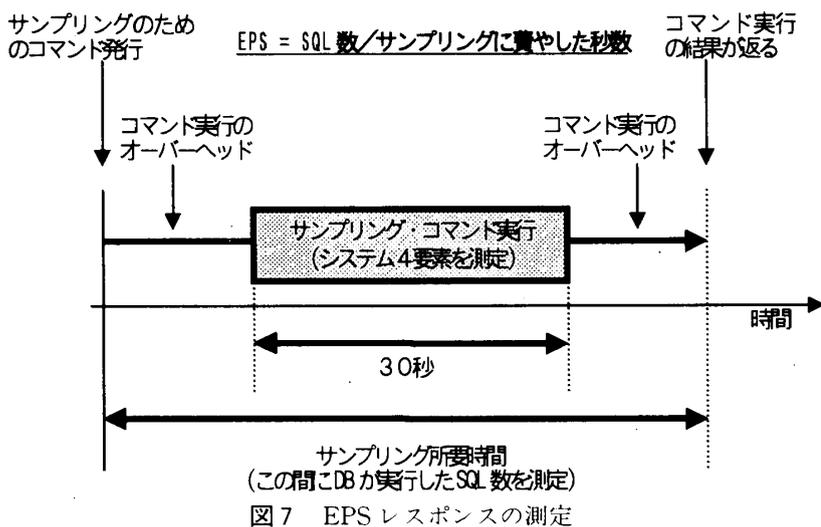


図10の棒グラフ部分を見ればチューニング前と比べSQL数が3倍程度に増えたことがわかる。

多くの場合、EPSレスポンス指標は、この例のように0.9以上であれば多少ブレている方が「良いシステム」といえる。

良いシステムとは、資源を限界まで出し切れる、チューニングされたシステムを指す。

8/17が第1フェーズのチューニングであるとする、今後は増え続けるトランザクション数から、新たなチューニングの第2フェーズに向かっていく。

6. システムが許容する最大同時トランザクション

良いシステムであるが故に、システム限界に達し、チューニングの第2フェーズを迎える場合がある。8/17のような第1フェーズのチューニングをクリアした新たなレベルの高いパフォーマンスの問題が発生する。先にも触れた「レスポンス契約」を実際に締結するのであれば、システム部門側は、システムの許容する「最大同時トランザクション数」を条件に盛り込む必要性がある。

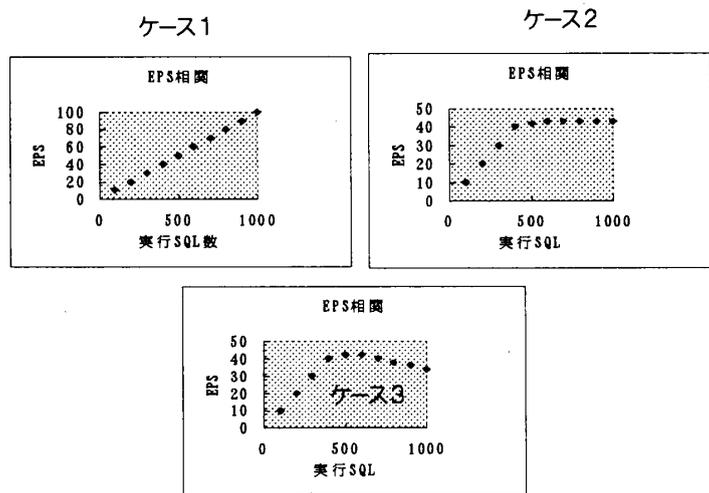
7. おわりに 一数值でシステムを評価する習慣を付ける一

Performance Insightは現場の状況をさまざまな数値から判断する。この数字の意味を理解し、数字で考える習慣をつければ、システムを客観的に評価できるようになる。

その模様を、ある企業のシステム担当者の会話で示してみよう。カンに頼る場合、図11(a)のような会話にならざるを得ない。一方、数値で評価するシステム担当者の間では、図11(b)のような具体的会話が可能になる。

どちらが、問題を的確に把握しているか(評価)、問題の対策を示せるか(診断)、過去の推移をもとに将来必要となるシステムへの投資(メモリ増設など)を示せるか(予測)——これは明らかに後者であろう。

このような定量的な評価・診断・予測の手段を提供し、最終的なゴールである「パフォーマンスを一定に保つ」ことが、パフォーマンス管理の目的である。



「横ばい」状況は起きていないように見えるが、右上の隅に若干の兆候を見ることができる。この相関図ではX軸のトランザクション(SQL)数が自動スケールされるため、今後のシステム使用率、すなわち、同時トランザクション数が増え続ければ、右上隅の横ばいは徐々に下に移行して行くことが予想できる。

レスポンス悪化と同時実行トランザクション数の関係

図10は、EPSレスポンス指標を時系列でチャートにしたもの(上部の折れ線グラフ)である。チューニングを行った8月17日を境に、このレスポンス指標は0.9より上の状態で「ブレ」ている様子が覗える。この場合、チューニングによりI/Oスレッシング・ポイント、すなわち、I/Oの限界に近づいたため、レスポンス指標にブレが生じたのである。このシステムを利用するエンドユーザから見れば、レスポンスが改善された分、処理できるトランザクションを増やせたのである。

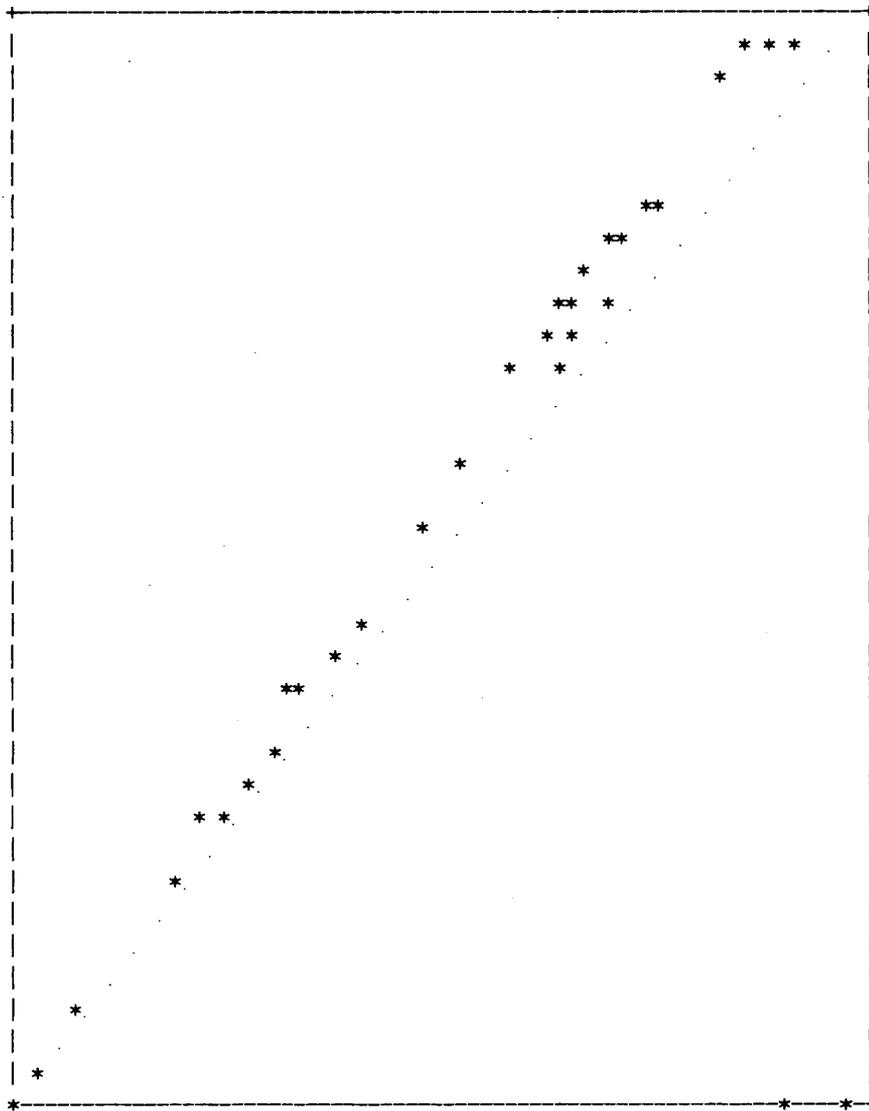


図9 実例

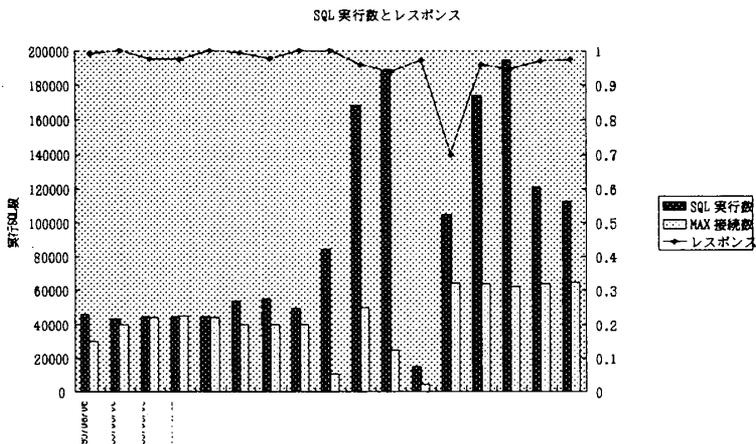


図10 EPS レスポンス指標時系列チャート

(a) カンでパフォーマンス管理をする場合の会話

先輩社員「このところXX支店のユーザから苦情きてる？」
 後輩「やっぱり日中は相当レスポンスが悪そうですね」
 先輩「やっぱりマシンの限界にきてるのかなあ」
 後輩「今度の月末処理がヤマですね」
 先輩「だけど半年前までは問題なかったのに」
 後輩「ユーザ数が増えましたからねえ」
 先輩「どのくらい増えたの？」
 後輩「おそらく倍ぐらいになったんじゃないかな」

(b) 数値で評価する場合の会話

先輩「XX支店のEPSは維持できてる？」
 後輩「朝9時から10時までと夕方4時から5時まではシステム・オーバヘッドによるEPSダウンが起きてますね」
 先輩「ボトルネックの要因は？」
 後輩「専用サーバを使う処理が集中するのでメモリ不足からのシステム・オーバヘッドが多すぎるんです」
 先輩「半年前と比べてどこが違うの？」
 後輩「EPSは平均で5%ぐらいしか上がっていませんけどI/O量が2倍になりました」

図11 システム担当者の会話