2004－EVA－9　(10)
2004／6／11

キャパシティ管理ツールとユーザ独自データの併用効果

株式会社アイ・アイ・エム　技術部　塩川　孝雄

ユーザにおける稼働統計管理は、各業務部門のシステム化計画の形成並びにシステムの安定運用を目的として、各ユーザが独自に管理項目を設定している。それに対し、市販のキャパシティ管理ツールは負荷監視とボトルネックの分析機能に主眼が置かれ、多数のユーザを対象とする汎用性が重視されてきた。本稿では高負荷サーバ群の改善検討事例によりキャパシティ管理ツールとユーザ独自データ双方の稼働データを併用する効果を示すと共に、今後の総合的なキャパシティ管理への道筋を探る。

The merit of using capacity management tools together with the performance data of user sections' own

Takao Shiokawa, Engineering Deopartment,IIM Corporaion

Each user section decides their original performance index in their own to manage their service operation. On the other hand, capacity management tools are designed to be versatile to treat various service operations. I show the merit of using capacity management tools together with the actual performance data from user sections, in a sample case of the improvement on a high load server group. In addition, I give the route to the total capacity management in the future.

1. はじめに

分散システムの拡大に伴い、日常運用でのキャパシティ管理は障害発生後の対処療法（Reactive）的な管理が目立つようになってきている。

コンピュータ設備を適正規模に維持するには、障害の未然防止に向けた先付け（Proactive）の管理が重要である。

このためには、一般に言われていることではあるが、リソースの「使用率」に加えて「処理件数」および「処理時間」の、相互に検証が可能な３つの次元の異なる要素を適宜組み合わせて管理する必要があることを再認識したい。

本稿ではこの３つの要素を含む稼働データを「監視」から「統計」「分析」までの各局面で同時並行的に可視化することで、大規模なサーバ群の異常負荷の発生を早期に見極めることができた事例を取り上げる。

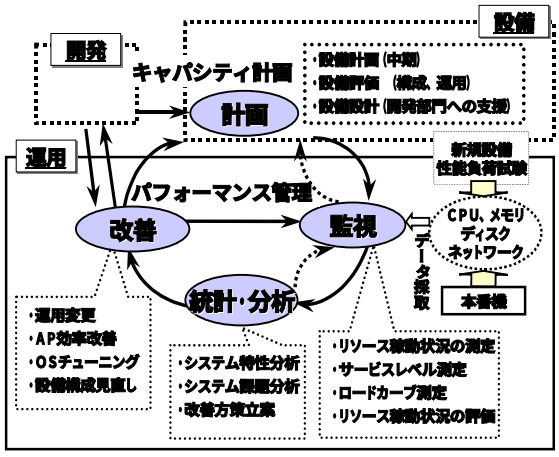
これまでのユーザにおける稼動統計管理システムの維持が困難になるなかで、今後もユーザ主導による総合的なキャパシティ管理を行なっていくための道筋はどのようなものか、日常のシステム運用管理の現場から探ってみたいと思う。

2. 本格運用後のキャパシティ管理

最初に、システム運用管理者によるキャパシティ管理について概括してみる。

① キャパシティ管理における運用部門の役割

キャパシティ管理は、システム開発段階での「キャパシティ計画」と運用段階での「パフォーマンス管理」の両者間のフィードバックサイクルとして考えることが出来る。



システム改訂や予測できない負荷変動等のために、開発段階だけでなく運用段階においても性能試験やチューニングおよび設備規模の見直し・更新が繰り返されて

いる。

運用部門は、日常の稼動監視によりシステム全体の負荷状況を体感的に把握し易い立場にある。

このため運用部門は開発部門と連携しながら運用変更やＯＳのチューニングから設備見直し提案までの一連の流れにおいて、中心的な役割を果すことが可能である。

② キャパシティ管理が難しいケース

一般に、大規模な分散システムのサーバ群を対象とした異常負荷の見極めは、通信回線負荷が関係することにより複雑となる。

日常のキャパシティ管理の要点は、オンライン処理では「レスポンス時間の遅延」、オンライン終了後の一括処理では「終了時刻の遅延」の状況を早期に把握し、問題の未然防止や極小化を図ることにある。

システムのライフサイクルからみると、処理時間やレスポンス時間の遅延で問題になるのは、本格稼働開始直後の急激な処理件数の増加と長期間稼働しているなかで徐々に処理件数が増加する２つのケースが考えられる。

前者のケースでは、多くの関係者が注目するなかでＤ／Ｂインデックスやプログラムのチューニングが行われるため、システム運用管理者としては比較的容易に関係者の協力を得易いといえる。

しかし、後者のケースでは、処理件数の再予測や再チューニング、設備増強等の対策に至るまでシステム運用管理者が主体的に行動を起こさなければならない。

また、システム開発要員の交替やドキュメントの不足、メーカサポート期限切れ等の複数の要因により、問題の発生に気付くこと自体が難しく、改善までに長期間を要することがある。

エンドユーザから「レスポンス時間が長すぎる」との連絡が入ったら、システム運用管理者はまず何を確認するだろうか。

サーバや回線負荷が現在どのレベルにあるか、障害が発生していないかを最初に確認することであろう。

しかし、障害も無くいつもと同じしきい値以下の数値だったとしたら、その情報が正しいのかと疑うことになる。５Ｗ１Ｈで正確に連絡して欲しい。今発生しているのか、過去の話なのか、それとも遅いと感じただけなのか。忙しいので後回しにしていたら、その後連絡が無い。等々・・・。この様なことは、大規模なネットワークに限らず、システム運用管理者は一度ならず経験していることである。

本来、この様な場合の対処方法は簡単なはずである。それは、「レスポンス時間」を現地に採りに行くこと、「処理件数」の推移を確認すること、そしてリソースの「使用率」と対比してみることである。ところが、現実にはシステム運用管理者が現地確認をすることの様々な難しさがある。その役割は開発部門か本店の業務主管部門かもしれないし、社内統制やセキュリティ上の理由等で業務主管部門から断られる可能性もあり、ましてやアウトソーシング受託をしているとすればなおさらの難しさがある。

しかし、重要な問題であれば、必ずエンドユーザから同じ連絡が入る。そして、問題が大きくなってから慌てているのが現状である。

③ 次元の異なる3つの基本管理項目

いずれにしても、「リソースの使用率」「処理件数」および「レスポンス時間（終了時刻）」の３つの基本管理項目の把握が重要である。

処理件数の変化に対し、ＣＰＵなどの使用率やレスポンス時間がどのように変化したかを相関的に把握することにより上限のしきい値を設定することができる。

使用率については、現在のところＣＰＵの使用率の継続測定が有効である。処理件数はＣＰＵ使用率に最も相関のある処理項目を選択して、その相関のずれを把握できるようにする。また、レスポンス時間は、現地の端末での体感時間を測定できればベターであるが、サーバの業務プログラム（ＡＰ）で擬似的なレンポンス時間を継続測定することも有効である。

一般に前記②のような理由でレスポンス時間の測定を放棄する傾向にあるが、必要な時は躊躇せず収集するよう心がけることが肝要である。

④ 運用段階での監視、統計、分析の同時性

システム開発段階において管理しきい値を設定しておくことが理想である。

しかし、実際にはシステムのブラックボックス化等により、運用開始後においても負荷予測やしきい値が設定されているとは限らない。

また、開発と運用段階での分析項目は基本的には同じといえる。

このような中で、システム運用管理者はＳＬＡを維持するため、問題に気付くことから課題の見極めまでを短時間で行うことが求められる。

このため、後に事例として述べるように、監視と統計・分析をほぼ同時並行的に行う必要がある。

⑤ 日常運用でのキャパシティ管理の課題

開発部門やメーカなど多くの関係者が同一の認識を持って課題解決に向けて検討が行えるようになるには、問題となる現象がどの様なものかを最初に明らかにしなければならない。

これには、ユーザの各システム環境の実態に合わせた工夫が必要となる。

リソースの「使用率」に関しては市販のキャパシティ管理ツールの標準機能により比較的容易に採取できる。ところが、「処理件数」と「レスポンス時間」の採取はエンドユーザとのＳＬＡの維持においてシステム毎に要点が異なり、市販の汎用的な管理ツールだけでは把握するのが困難である。

一方、部門別・ＡＰ別等の処理件数や処理時間は、一般に各ユーザにおいて各業務部門のシステム化計画の形成やシステムの安定化のための稼働統計管理データとして採取されることが多い。

しかし、これらのユーザ独自の稼働データは、システムの運用管理段階へ引き継がれ、日常のキャパシティ管理の現場で有効に活用されているのだろうか。

3． 日常運用でのキャパシティ管理の事例

以下にキャパシティ管理ツールとユーザ独自データの併用例を日常運用でのキャパシティ管理モデルとして述べてみよう。

これは、２項の②で述べた「長期間稼働における処理件数の増加によるレスポンス時間の遅延のケース」で運用対応が比較的難しい事例といえる。

＜モデルサーバ群の概況＞

・該当のサーバ群は、広範な地域の各事業所に分散設置され、中央にて一括管理（約５０台）
・オンライン処理はサーバで行い、一括処理はホスト機で処理
・システム開発から数年以上経過し、処理データ量の増加に対してチューニングやサーバ間でのリソースの融通および設備増強をその都度行い、延命化

① しきい値設定等による監視状況

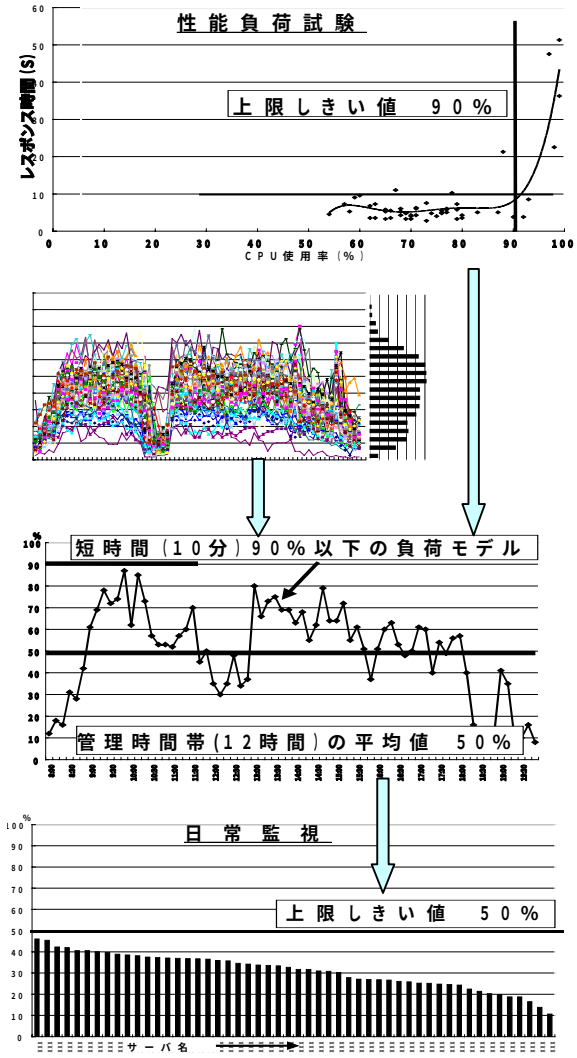
市販のキャパシティ管理ツール（ＥＳ／１　ＮＥＯ）を活用して、各サーバからＣＰＵ使用率のほかメモリー、ディスクなどの一連の稼働データを採取している。

このうちＣＰＵ使用率については、管理用パソコンで横並びグラフにより１日数回定期的に自動更新表示して監視を行っている。

同一の監視用グラフをＷｅｂ方式により開示し、常時、開発や設備部門も負荷の把握を可能としている。

この他、該当のモデルサーバ群はお客様対応用のため特に重点監視サーバとして位置づけられており、１日のＣＰＵ使用率の推移グラフも表示される。

オンライン時間帯での上限しきい値は、１２時間平均でＣＰＵ使用率５０％（月平均で４５％を上限）としている。しきい値の設定は、開発部門との共同で行なった負荷テストにより１０分平均で上限が９０％であることと日常の負荷モデルのロードカーブにより設定されたものである。



② レスポンス遅延発生への懸念

エンドユーザから「レスポンス時間が遅いのでは」との連絡があり、１ヶ月前に分析したが異常が見つからなかった。また、同時期に回線負荷に起因するレスポンス遅延の問題が発生しており調査中であったことから、サーバ負荷としては異常が無いものとしていた。

翌年度は、若干の負荷増（ＣＰＵ使用率で５％程度）があるとの情報もあったが、これまでのチューニング成果により乗り切れるとの予測があった。

しかし、これまでの推移からシステム運用管理者の勘としては徹底した調査の必要があるのではないかと感じていた。

③ 問題を見極めるためのユーザ独自データの活用

具体的な改善のための詳細分析を行うには、異常負荷を含む全体を網羅する稼働データを分析することが肝要である。

特にサーバ台数が多い場合の詳細分析では、十分な見極めをせず適当な日の負荷を分析することは問題の発

見を遅らせるために危険である。

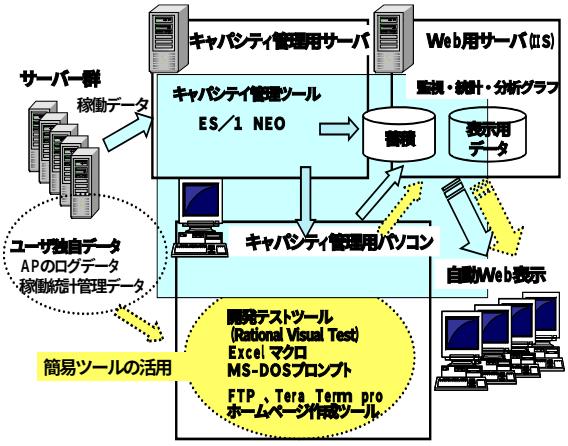
異常負荷のパターンがどのようなものか、またどの時間帯で発生しているかを見極めるには、まず先に述べたリソースの使用率と処理件数、それにレスポンス時間に関するデータが必要となる。

このため、ユーザの稼働統計管理システムの蓄積データやログファイル、ＡＰの出力ファイルなどを調査する。

この例では、稼働統計管理システムに蓄積されていた「ＡＰ単位の処理件数」の推移データとサーバ内のＡＰによるレスポンス時間監視の「タイムアウトエラーのタイムスタンプ」のログアウトを使用している。

そして、既導入のキャパシティ管理ツールのグラフ表示に加え、必要な項目の自動採取、グラフの自動作成までを自動化しＷｅｂ表示することとした。

キャパシティ管理ツールとユーザ独自データの組合せ



ここで重要なことは、サーバのＣＰＵ使用率と稼働統計管理システムで蓄積された処理件数、およびＡＰによりログ出力されたタイムアウトエラーを同一画面上で統計処理をして相関が分かるように表示することである。備え付けの管理ツールの機能に無いからやらないというのでは先へ進まない。

対象の約５０台の全サーバから稼動データをＦＴＰにより採取してグラフ画面を編集し、併せてデータの蓄積も行う。

ここで、自動化の手順を簡単に説明してみよう。

最初に、自動化するための雛形として、システム運用管理者は、ネットワークの日常運用管理で行っているのとほぼ同様の方法で管理用パソコンにより一連の操作を一度だけ行う。

この操作の記録を開発テストツール（Windows のプログラム開発用）と Excel マクロのトレース機能をセットして行うことで、ほとんどコーディングレスにより稼動データの採取からグラフ表示までを記録できる。

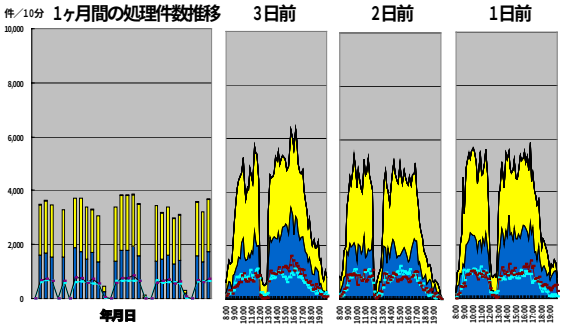
この１回の一連の操作記録（コーディング）をコピーして、ＩＰアドレスやファイル名の変更などを行う。データ採取から出力グラフをＷｅｂ画面で見られるようにするまで、システム開発というよりはむしろ環境設定という感覚で行うことが出来る。

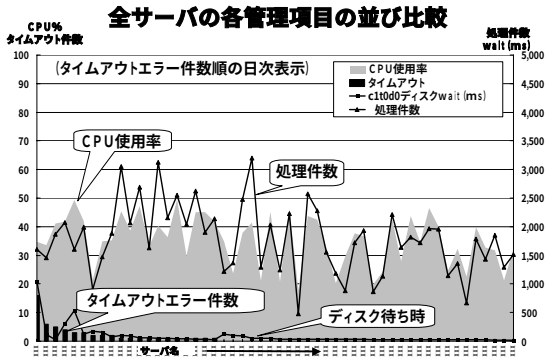
これらの一連の作業はネットワーク全体を把握しているシステム運用管理者であれば、少しの努力で効率よく必要機能を設定できる。但し、暫定的な対応のため汎用性は持ちえない。出力画面は５種で、１日約２００画面を自動作成したが、これに要した期間は１人のシステム運用管理者が次の④項の問題の見極め分析を含め、１ヶ月程度で実施できた。

④ システム運用管理者による監視･統計･分析

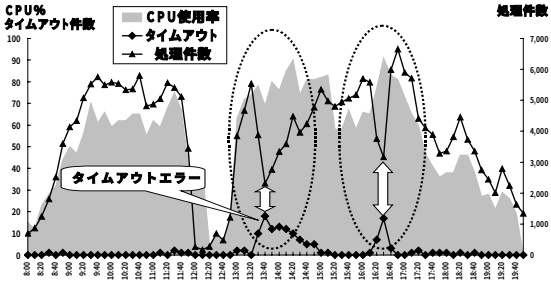
③項の段取りにより、監視と統計・分析を並行して同時に行うことができた。

環境設定を行いながら、約１ヶ月間の負荷の傾向を把握した結果、短期間のデータ採取では異常負荷は発生しないが、月内のある特定の時期に多くのタイムアウトエラーが発生していることをつきとめた。





ＣＰＵ使用率、処理件数、タイムアウト件数の推移



そして、処理件数低下と同時にタイムアウトエラーが多発している推移グラフが確認された。

これは、全サーバの「使用率」と「処理件数」および「レスポンス時間（当事例ではタイムアウト件数）」の横並び比較、および同一推移グラフ上での相関表示による傾向把握ではじめて発見されたものである。

ここでは、監視、統計および分析を同時並行して行っている。どれか一つの項目の監視では気がつくのは難しい。

この結果を受けて、タイムアウトの多発時間帯におけるユーザ・コマンド別のＣＰＵ使用率、ディスク、メモリーの状態および通信のパケット数・エラー数等の各種のデータを個別に採取し分析をした。

これによると、「ＡＰの実時間監視によるタイムアウト検知」→「通信回復処理」→「同処理結果のディスクへの大量なログアウト」→「ディスク待ち時間の遅延」という連鎖により、結果としてオンラインレスポンス時間が遅延したものであった。これはエンドユーザ側でのレスポンス遅延時間帯と一致していた。

以上がシステム運用の現実である。

⑤ ボトルネック解明と改善

異常負荷が発生しつつあることが明確に確認できたことにより、詳細分析によるボトルネックの解明作業に移行することができ、メーカを含む開発・運用部門および設備の各部門の共同により改善案が検討された。

この分析は、市販のキャパシティ管理ツール（ＥＳ／１　ＮＥＯ）の活用などにより詳細分析が行なわれ、同メーカのＳＥの協力も得て行なわれた。

なぜＡＰでタイムアウトが検知されるのか一次原因を明らかにする必要があったため、過去のチューニング経過の資料、ＣＰＵ、メモリー、デバイスに関する管理ツールでの相関分析、Ｄ／Ｂの諸数値の採取、コマンド別の処理推移分析等を行なった。

この結果、一次原因はＣＰＵ負荷を抑えるためメモリー使用量を大きくし過ぎて、メモリー不足がボトルネックになっていると判断された。この判断に基づき、ユーザ利用実態に合わせたＡＰのメモリー常駐数の調整を中心としたチューニング、ログアウトの制限などの改善が実施され、安定稼動を維持することができた。

また、将来の負荷増を考慮してサーバ群全体の設備更新の検討をするまでに至っている。

4. キャパシティ管理ツールの概要

前にも述べたが、市販のキャパシティ管理ツールは、各種の異なるＯＳやＤ／Ｂなどを考慮し、稼働データを取り扱う上で汎用性をもたせてある。

ＩＴの進展に合わせて、Ｗｅｂサービスの端末や回線制御装置、ＥＲＰパッケージからの処理件数や模擬的なレスポンス時間の採取など、多くの機能向上が図られつつある。

しかし現段階においては、ユーザ独自で作成された稼働データを市販のキャパシティ管理ツールに取り込むには、ツールの提供会社とユーザでのＡＰ開発段階からの事前準備が必要である。

汎用的なキャパシティ管理ツールは、開発段階でのキャパシティ計画・テスト用、および運用段階でのパフォーマンス管理で使用する監視用およびボトルネックの分析用（開発テストでも使用可能）に大きく分けられる。また、これらの複合的なツールもある。

運用段階で使用されている管理ツールの主要な機能には、稼働データの採取と蓄積、負荷の監視、分析機能、および可視化機能などがある。

5. 各種のユーザ独自データ

大規模なコンピュータシステムの管理では、稼働統計管理システムを保有してきたユーザが多いと思われる。

この他、ユーザ独自の稼働データは、前にも述べたＡＰによる各種のログアウト、システム運用管理システムでの管理データ、およびエンドユーザ向けの出力帳票や画面表示の中にも存在している。

次に、ユーザ独自データの一般的な蓄積事例を確認する。

① 稼働統計管理システム

「店所コード」「サーバＩＤ」「年月」「ジョブコード」「サブジョブコード」「プログラム名」および「ユーザ名」などを体系的にコード化して稼動統計データとして蓄積保管している。

コード別の稼働データには、「処理件数（実行回数）」、「経過時間」「ＣＰＵ使用時間」、「ディスク使用・空き状況」「ログイン失敗状況」などがある。

これらは、ＡＰ開発段階で保存対象とすることで、稼働統計管理システムへ稼働データとして取り込まれる。

システム資源全体に関わる稼働状況データとしては、「ＣＰＵ使用状況」「メモリー使用状況」「ディスク使用状況」などが蓄積される。

また、定期的な日・月・期報などにおいて、合計・平均・ピーク等の統計処理がされ、オンライン検索機能などの工夫がなされている。

② APログや出力帳票での稼働データ

レスポンス時間や処理内容別の処理件数がＡＰによりログアウトされ、運用段階で継続して蓄積することも多々行われている。また、オンライントランザクションが一定時間以内に終了しない場合、「タイムアウトエラー」としてログファイルに記録される。

サーバ内で採取されるレスポンス時間は模擬的なものであるが、レスポンス時間を重視するシステムではパソコン端末内部でログしている場合もある。

これはＳＬＡの維持管理用として有用なデータである。

システム開発段階から管理者へのログ情報の内容が引き継がれないことがあるが、少なくともＳＬＡに関わるレスポンス時間のデータについては引き継いで欲しいものである。

③ 各種のシステム運用管理のデータ

日常のシステム運用管理では、ファイル管理（ディスク容量管理、ＭＴ管理等）、障害管理、プログラム入れ替え管理、ＪＣＬ管理、およびジョブスケジューリング、ＡＰのエラー、システム停止件数・時間やメンテナンス時間等の各種の管理データが存在する。

④ その他の稼働データ

以上のほか、ＤＷＨやＣＲＭ（Customer Relationship Management）に蓄積されているお客様件数や取引件数などのユーザオリエンテッドのデータはキャパシティ管理の基礎データとして活用可能である。

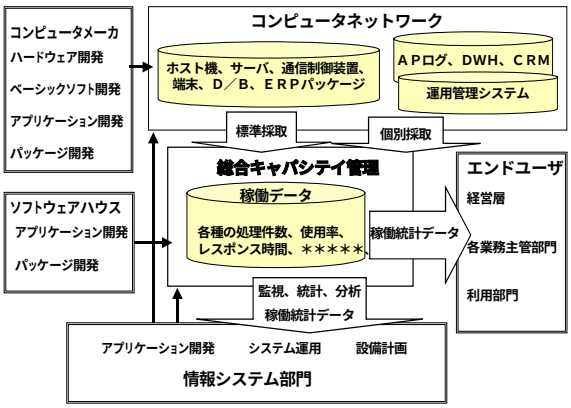
6． 今後の総合的なキャパシティ管理に向けて

これまで、情報システム部門はリソースの「使用率」、業務主管部門は各業務の「処理件数」、またエンドユーザは「処理時間（レスポンス時間）」と、各部門の関心事が異なりデータも分散する傾向にあった。

そのため、これらの稼働データを集めるのに時間を要し、前項までに述べたように日常のキャパシティ管理が後追いになり易かった。

今後は、ＩＴＩＬ（Information Technology Infrastructure Library）でも示唆している通り、稼働データの蓄積ファイルについては、情報システム部門やエンドユーザおよびメーカなどの関連個所がＡＰの開発段階から容易に共有可能とする入・出力の標準化が必要と思われる。

総合的なキャパシテイ管理の運営



一方で、稼働統計管理システムは分散システムの拡大や多様化に伴うコスト高を理由に、維持が困難になる状況が見受けられる。

このような中において、稼働統計管理システムを包含した総合的な市販のキャパシティ管理ツールの早期実現を期待してやまないところである。

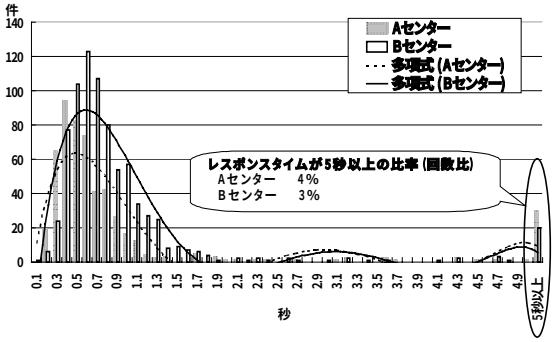
しかしながら、それぞれ事情の異なる多くのユーザが満足できる管理ツールを実現するには、ユーザ間の時間的なずれなど多くの困難が予想される。

そのためには、今後の日常におけるキャパシティ管理においては管理の視点を変えていく必要があると考える。

これまでのキャパシティ管理はリソースの「使用率」に重点をおき、他の管理項目はおろそかになりがちであったことから、問題に気付くことから課題の見極めまでに時間を要してしまっている。

少ない人的資源でより多くの効果を挙げるには、ＳＬＡ中心の管理、つまり各エンドユーザに最も理解が得やすい端末の体感的なレスポンス時間（定点測定など）の可視化を中心としたシンプルな日常管理方式へ移行することが最善と思われる。

ＳＬＡに基づくレスポンスタイムの監視



これを推進するには、ネットワークシステム全体を最も把握し易い立場にあるシステム運用管理者が、能動的な運用ＳＥとして、総合的に技術力を駆使して自由にレスポンス時間の実態の把握を行えることが大切である。

そして、この運用ＳＥを中心にメーカをはじめ開発・設備の各関連箇所が垣根を越えて支援し合えるシステムの日常運用の環境作りこそ、今後のキャパシティ管理の鍵であると思われる。

参考文献：

Adam Grummitt,Metron「Performance Assured by Capacity Management:Experience refined through ITIL practice」