Designed for Microsoft BackOffice

NetIQ



White Paper
ホワイトペーパー

NetIQ AppManager Suite 適用事例
Lotus Domino のパフォーマンスとアベイラビリティを
監視するためのガイド

目次

1.0 はじめに 1

2.0 LOTUS DOMINO のアーキテクチャ上の主要コンポーネント 2

2.1 DOMINO サーバー タスク 2
2.2 データベースとドキュメント 3
2.3 NOTES ログと DOMINO ログ 3
2.4 ルータと MTA 4
2.5 NOTES 統計 4
2.6 どのように APPMANAGER がメッセージング インフラの表示をサポートするか 4

3.0 DOMINO サーバー管理の問題点と対策 5

3.1 DOMINO サーバー タスク 5
問題点 5
APPMANAGER はサーバー タスクをどのように管理するか 6
3.2 データベースとドキュメント 6
問題点 6
APPMANAGER はデータベースとドキュメントをどのように管理するか 7
3.3 NOTES ログと DOMINO ログ 7
問題点 7
APPMANAGER は NOTES ログと DOMINO ログをどのように管理するか 7
3.4 ルータと MTA 8
問題点 8
APPMANAGER はルータと MTA をどのように管理するか 8
3.5 NOTES 統計 8
問題点 8
APPMANAGER は NOTES 統計をどのように管理するか 9
3.6 サービス レベル要件の管理 10
問題点 10
APPMANAGER はサービス レベル要件をどのように満たすか 11

4.0 まとめ 11

1.0 はじめに

メッセージング ソリューションは急速に企業の通信基盤となりつつあります。特に現在、電子メールは企業運営に欠かせないユーティリティとみなされており、同じビルにいる社員間だけでなく、地球の裏側のビジネス・パートナーとの通信にも欠かせないものとなっています。電子メールがたとえ 1 時間でも使えなくなれば、フラストレーションの原因となるだけでなく、最悪の場合、企業の生産性ひいては収益にも重大な悪影響を及ぼしかねません。

強靱な統合メッセージング ソリューションを望む企業の多くは、その優れたメッセージングおよびグループウェア機能から Lotus Domino を選択しています。さらに、Lotus では Domino にウェブ サーバー機能を追加することにより、従来のメッセージングおよびグループウェア機能を拡張しています。このように Domino は複合ソリューション セットに進化し、企業の IT 部門に対して管理上のさまざまな挑戦課題に対応しています。

Domino の実運用環境に導入されるアプリケーションがますますビジネス クリティカルなものになるにつれて、経営陣は企業の IT 部門に対し以下の達成を強く望むようになっています。

- Domino サーバーの週 7 日 24 時間のアベイラビリティとパフォーマンスを確保する。
- Domino 環境の導入と管理に関連するサポート コストを削減する。
- 企業のサービス レベル要件を満たす。

これらのゴールを満たすことは、Domino 自体が急速に受け入れられていることによりさらに複雑になっています。また、Domino についての専門技術をもつシステム管理者の需要が供給をはるかに上回っています。システム管理者が経営陣の期待に十分応えようとするならば、Domino 自体と同じくらい信頼できる強靱かつ総合的な管理ツール群が必要となるのは疑いのないところです。

ロータス社は数多くのツールおよびコンポーネントを提供し、Domino サーバーの運用管理を支援しています。ロータス社のツールには、Notes Admin Client、Web Admin Tool、Domino console のほか、Reporter、Collector、Event の各サーバー タスクなどが含まれます。さらに、Windows NT Performance Monitor (PerfMon)では Notes 統計(Notes statistics)も利用できます。これらのツールは有用で、ある程度の管理サポートは提供しているものの、数台以上の Domino サーバーを管理するには効果的ではありませんし、Domino サーバーの数の増加に対応できるスケーラブルなソリューションでもありません。また、分散環境でサーバーを管理するための特殊サポートも提供していません。

さらに、これらのツールは、分析、グラフ作成、レポート作成のための統合化されたシームレスなメカニズムを提供していません。パフォーマンス データの収集は、Domino 環境を維持するための主要要件ではありませんが、システムの生産性や効率を分析したり、将来のキャパシティ要件を計画したり、サービス レベル要件を確認するためには絶対に欠かせません。

NetIQ の AppManager for Lotus Domino は分散 Domino 環境のパフォーマンスとアベイラビリティを監視するための企業のニーズを満たす唯一の製品です。AppManager for Lotus Domino は、サーバーの稼働状況やアベイラビリティの保証からレポート作成に至る広範な管理ニーズをカバーする、総合的なスケーラブルで信頼性の高いソリューションです。

AppManager が、分散 Windows NT システムやサーバー アプリケーション(Lotus Domino、Microsoft SQL Server、Microsoft Internet Information Server、Citrix WinFrame など)のパフォーマンスとアベイラ

ビリティを監視するための、業界でもっとも総合的なソリューションであることは疑いありません。管理者は 1 台のセンター コンソールから「知識スクリプト(*Knowledge Scripts*)」と呼ばれる監視機能を設定することができます。知識スクリプトはパフォーマンス データを収集し、単純または複雑なイベントを監視します。イベントが起こると、知識スクリプトはダウンしたサーバーの再起動やポケベル メッセージの発行などの修正動作を行うことができます。これにより、高度に分散された Windows NT 環境において問題をプロアクティブに検出・解決できる、強力な自動化されたソリューションが提供されることになります。また、管理者は用意された 165 以上のレポートを実行したり、独自のレポートを作成することができます。

以下のセクションでは、分散 Domino 環境をどのように管理するかを分析し、お使いの Domino サーバーの連続稼働と高パフォーマンス運用を確実にするための実践的な戦略を提案します。最初に Domino の中核となるコンポーネントを記述し、次に運用管理の面からこれらのコンポーネントをどのように管理すべきかについて述べます。そして、リアルタイム パフォーマンスを保証し、サービス レベル要件を満たし、さらにキャパシティ プランニングを可能にするために必要とされるキーの Domino パフォーマンス メトリックスは何かを解説します。各トピックでは、それらのコンポーネントやメトリックスを管理するのに AppManager がどのように役立つかについても説明します。

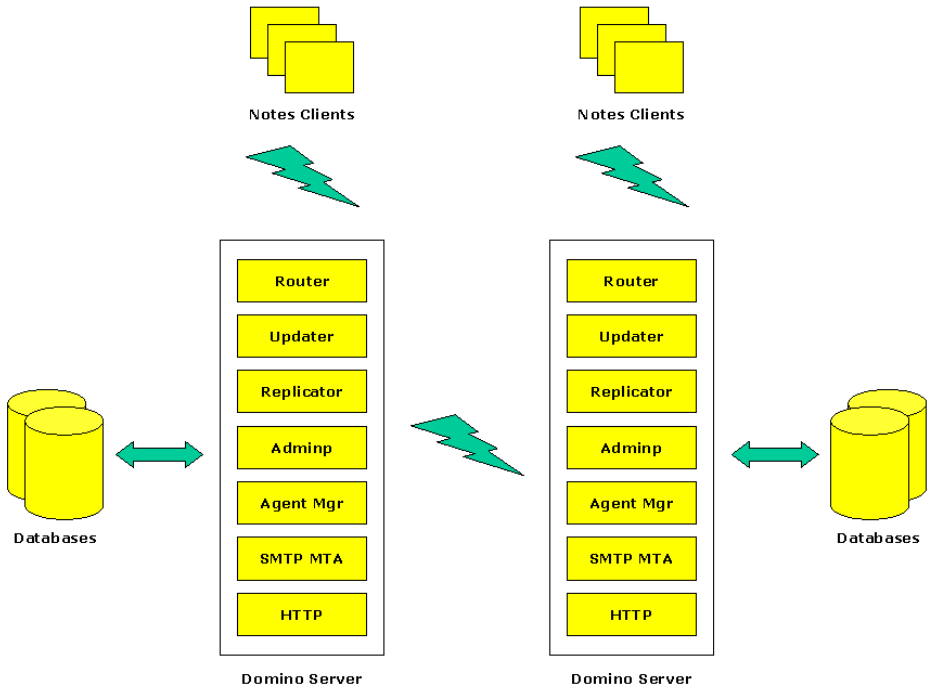
2.0 Lotus Domino のアーキテクチャ上の主要コンポーネント

Domino 環境でのパフォーマンスとアベイラビリティを適切に管理するには、Lotus Domino のアーキテクチャ上の主要コンポーネントをまず理解することが必要です。

2.1 Domino サーバータスク

各 Domino サーバーは一連のサーバー タスクから構成されており、サーバー タスクとは Notes クライアントと他の Domino サーバーからのリクエストとトランザクションを処理します。Domino の管理者は NOTES.INI ファイルの中でこれらのタスクを設定し、サーバーの起動時にタスクが自動的にロードされるようにします。

Domino アーキテクチャは Lotus が LDAP (Lightweight Directory Access Protocol) や IIOP (Internet Inter-ORB Protocol)などの新しい機能をサポートできるように拡張したものです。このアーキテクチャにより、ISV は Lotus では考えられない拡張機能を提供できるのです。



2.2 データベースとドキュメント

Domino 内のデータは Notes データベースに個々のドキュメントとして保存されます。各データベースにはゼロ個以上のドキュメントを入れることができます。データ ストアはきわめて柔軟で、Domino 自体が「*公開アドレス帳(Public Address Book)*」と呼ばれる Notes データベースを使えるほど柔軟です。公開アドレス帳は Notes ドメインで定義されたオブジェクトのユーザー、サーバー、およびその他の設定情報のディレクトリを管理します。ディスクの使用状況の管理や Notes データベースへのアクセスの最適化は、Domino を監視する上で重要な側面です。

データベースへのアクセスと等しく重要なのはデータのセキュリティです。Domino 内にもいくつかのセキュリティ メカニズムはありますが、一般には ACL (Access Control List)を使って保護データへの許可のないアクセスを制限しています。主要なデータベースにおける ACL の変更を管理し追跡することは、かなりの時間と労力を要します。

Domino の技術的に大きな長所はデータベースのレプリケーションです。データベースのレプリカを複数のサーバー上に置くことができます。ユーザーはレプリカ上の情報をアクセス・更新し、Domino を使ってデータベースの同期を取ることができます。

2.3 Notes ログと Domino ログ

Domino サーバーが作成する通知、警告、エラー メッセージは、Domino コンソールに表示されます。さらに、Notes 関連メッセージは Notes ログに、Web 関連メッセージは Domino ログに保存されます。これらのログは一般には各サーバー上にローカルに保存されます。自動化されたソリューションがないと、個々のログを調べてクリティカルなエラー メッセージを検索するのは時間がかかるだけでなく間違いも起こりやすいものです。

2.4 ルータとMTA

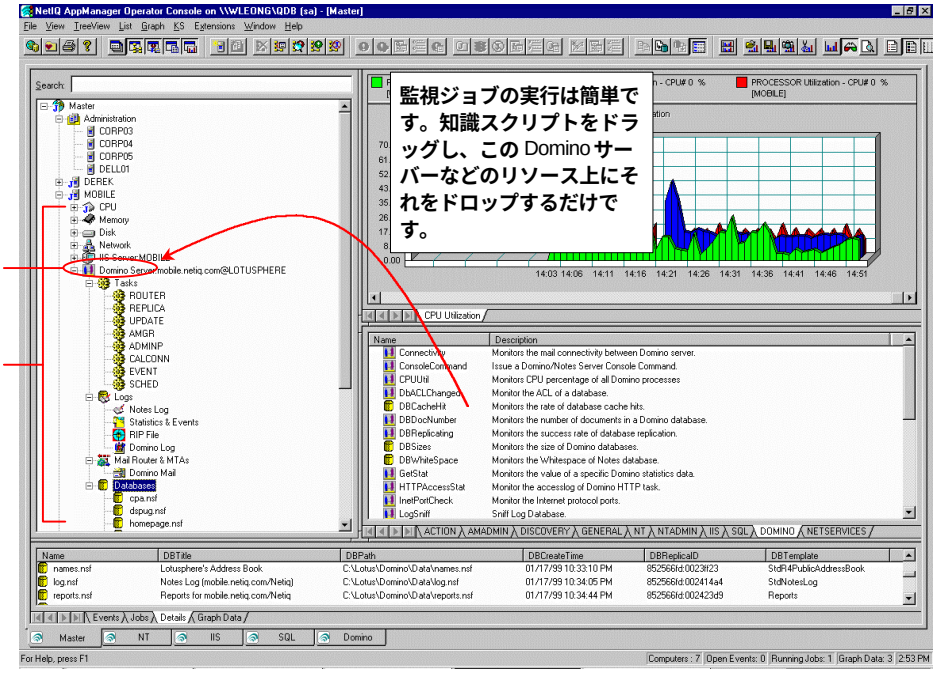
メッセージング プラットフォームとして、他のユーザー宛のメッセージの経路を選択し配送することは Domino にとってきわめて重要です。Lotus Domino では複数のサーバー タスクを使ってこの雑用を処理するように設定することができます。ネイティブの Notes メール ルータはメールを他の Domino サーバーに宛てたり Notes ユーザーに配送するのに使われます。MTA(Message Transfer Agent)は Domino と他のメール システム(SMTP、X.400、または cc:Mail)との間のメール転送に使われます。

2.5 Notes 統計

Notes 統計は Domino サーバーの稼働状態とパフォーマンスを監視するための重要なソースを提供します。Windows NT 上では、この情報は Windows NT Performance Monitor (“PerfMon”)内、または Notes API 経由で利用できます。使用可能な Notes 統計は 200 を超えますが、管理面からどの統計を監視するのが重要かを知ることは挑戦課題です。

2.6 どのようにAppManager がメッセージングインフラの表示をサポートするか

NetIQ の AppManager を使うと、Domino サーバーのエレメントを容易に表示することができます。1 台のセンター コンソールから、システム、アプリケーション、およびこれらのシステムやアプリケーションに対応するリソースをアイコンとして表示することができます。以下の例では、*MOBILE* と呼ばれる Domino サーバーが AppManager Operator Console に表示されています。



ここに見るように、AppManager の素晴らしい機能の 1 つが単一の GUI(グラフィカル ユーザー インターフェイス)からお使いの全システム リソースを一元的に表示できることです。つまり、Domino サーバーだけでなく、Windows NT システム、Microsoft Internet Information Servers、およびその他のリソースをトータルに表示することができます。サポートされているシステムの完全なリストにつ

いてはhttp://www.netiq.comを参照してください。また、サーバーを地理的条件、組織、機能などによってグループ化することもできます。

AppManager Operator Consoleから、ドラッグ アンド ドロップで、知識スクリプト(*Knowledge Scripts*)と呼ばれる監視機能を任意のリソース上で起動することができます。知識スクリプトを設定することで、ユーザーは自在にパフォーマンス データを収集したり、どのようなイベントでも監視することができます。また、知識スクリプトが特定のしきい値を超えたことを検知した場合には修正アクションが発生するように定義することもできます。たとえば、Dominoサーバー タスクがダウンした場合、自動的にそのタスクを再起動し、電子メールを送信し、ポケベル メッセージを発行し、ネットワーク マネージャへSNMPトラップを生成することもできます。

前の図で、システム管理者が*SMTPQueueLen*知識スクリプトを*MOBILE* Dominoサーバーにドラッグ アンド ドロップし、SMTPのキューが長くなりすぎないように継続的に監視することができます。また、同様にシステム管理者は知識スクリプトをDominoサーバーのグループ上にドロップして、そのグループ内のすべてのDominoサーバーに対してSMTPのMTAキューを監視することもできます。

3.0 Dominoサーバー管理の問題点と対策

では、ここでは、前のセクションで記述したDominoサーバーの各コンポーネントについて、クリティカルな管理上の問題点を分析します。まず、それぞれの問題点を洗い出し、なぜこの問題がDominoサーバーの正常な運用稼動状態を守るために重要なのかを説明し、さらにどうしてAppManagerが最適なソリューションを提供できるのかを記述します。また、サービス レベル要件に関する管理上の挑戦課題と、AppManagerがそれらをどのように解決するかを調べます。

3.1 Dominoサーバータスク

問題点

Lotus Dominoサーバーは複数のサーバー タスクから構成されています。各サーバーは異なるサーバー タスク セットで構成され、Notesドメイン内に特化された機能を提供することができます。たとえば、Dominoメール サーバーは他のメール サーバーとの間でメッセージの送受信を行います。Dominoアプリケーション サーバーはデータベースのリクエストに応えます。Dominoハブ サーバーは主要なデータベースを、接続されているDominoスポーク サーバー間でレプリケート(複製)します。Domino ウェブ サーバーはHTTP リクエストを実行します。さらに、Lotus は Domino を拡張し、多くのインターネット ベースの標準機能を取り入れています。これらの追加サーバー タスクの監視に加えて、ユーザーに対してこれらのサービスの連続可用性を保証することが一層重要となってきています。

各種のDominoサーバーが適切に機能することを保証するには、サーバー タスクのアベイラビリティを監視するのが良いでしょう。まれにですがDominoサーバー タスクがビジー(busy)状態のサーバーに固定されてしまう可能性があります。Dominoコンソールでサーバー タスクのステータスが通常(normal)となっていても、対応するWindows NTプロセスが非稼働状態または存在していない可能性もあります。このような非決定的な条件下ではオペレータが介入してサーバーをリブートする必要もあるでしょう。

良い管理ツールとは、各Dominoサーバー上のすべてのサーバー タスクを自動的に検索し、ユーザーがドラッグ アンド ドロップのような手早く簡単な方法で監視できるようにしたものでなければ

なりません。Domino ユーザーに対してダウンタイムをなくすため、管理ツールは障害サーバー タスクを検知し再起動(オプション)できなければなりません。管理ツールが Domino コンポーネントの稼働状態に関係なく、監視対象 Domino サーバーとは独立して機能できるようになっているのが理想です。また、管理ツールはポケベル メッセージの発行などの通知アクションに対して Domino とは独立しているべきです。

AppManager はサーバータスクをどのように管理するか

AppManager はあらかじめパッケージ化された知識スクリプトを用意しており、Domino サーバー タスクのステータスを監視し、障害タスクを再起動するオプションを備えています。さらに、AppManager for Lotus Domino はすべての Domino サーバー タスクまたは特定のサーバー タスクについて CPU とメモリの使用状況を追跡します。許容可能なしきい値を指定することができ、それを超えるとイベントが生成され Operator Console が変更されます。また、イベントにより、ポケベル メッセージを発行したり SNMP トラップを送信するなどのアクションをトリガーすることもできます。ユーザーは他のシステム リソース(Domino サーバー自身にインストールされていないリソースであっても)を利用するアクションを設定することができ、Domino サーバーの稼働とは独立した高信頼性の監視機能を提供します。また、後日の検索やレポート作成に使えるようにパフォーマンス データを AppManager の SQL Server ベースのリポジトリに記録するオプションもあります。

インターネットベースのプロトコルを実装している Domino サーバー タスクに対しては、AppManager は*公開アドレス帳*に保存されている *サーバー(Server)*ドキュメントを調べることによってその設定を判定し、インターネット ポートが有効か否かを検知します。

3.2 データベースとドキュメント

問題点

Notes データベースは Domino 環境内でデータのオブジェクト ストアとして機能するため、データのリクエストを効率よく実行することが Domino サーバーのパフォーマンスにとって最重要です。データベースのパフォーマンスは、バッファやキャッシュに使用できるメモリ、並列アクセスの数、アクセスのタイプなどを含む多くの要因によって変化します。データベースのサイズとデータベースのフラグメンテーション(ホワイトスペース)も大きな要因です。

データベースのスループットを最大にすることに加えて、すべての管理者にとっての関心事は、Notes データベースに保存されている機密情報を許可のないアクセスから保護することです。 いったんアクセス コントロール リスト(ACL)が設定されると、管理者は主要データベースの ACL の変化を監視して、その環境とデータの安全性を確保する必要があります。

データベースのレプリケーション機能は長い間、他のソリューションより秀でた点として Domino のセールス ポイントでした。企業全体でアクセスできるようにデータベースを分散するのが非常に容易である一方、エラーのない、しかもタイミングの良いデータベースのレプリケーションを保証するのは Domino の管理者に特有の挑戦課題です。さらに、使用パターンは時間の経過と共に変化します。あまりにも以前に作成されたレプリカは必要なくなり、貴重なシステム リソースを消費しているに過ぎないこともあります。

最後に、Notes データベースはアプリケーションがドキュメントに含まれているデータを保存するためのキューとして使用することがよくあります。状況によっては、「作業の進捗」を監視するためにデータベース内のドキュメント数を追跡することは有益です。

AppManager はデータベースとドキュメントをどのように管理するか

AppManager は前述の管理上の問題点に対応できる知識スクリプトを提供します。ユーザーは以下のように知識スクリプトを設定することができます。

- データベースのサイズを監視する。たとえば、AppManager は Domino サーバー上でメール データベースが最大級のユーザーを特定することができます。このリストを使って、ディスク スペースを大量消費しているユーザーには、定期的に電子メールをアーカイブするか一部削除するように促すことができます。

- ホワイトスペースの許容率(%)を定義し、そのしきい値を超えるデータベースを圧縮対象として選択する。

- データベースの ACL が修正されると、ユーザーにアラートする。

- データベースのレプリケーションがうまく行ったかどうかを判定する。知識スクリプトはデータベース名と追加/削除/更新されたドキュメントの数を返します。典型的なハブ アンド スポーク構成でのレプリケーション サイクルは一日中実行されるようにスケジューリングされます。AppManager は、ある時間内にあるサーバーのレプリケートが行われなかった(レプリケーション サイクルが抜けた可能性を示します)かどうかを検知することができます。

- アクセス頻度に基づいてデータベースを監視する。データベースの使用状況によって、頻繁にアクセスされるデータベースのレプリカを複数のサーバー上に作成するのも良い方法かもしれません。逆に、しばらくアクセスされていないデータベースのレプリカは、データベースの合併整理の対象、あるいはアーカイブまたは削除の対象とすべきです。

また、AppManager では、ビュー名または選択式によってデータベース内のドキュメント数を監視する、汎用目的の知識スクリプトも用意されています。これは Domino の FAX およびポケベル アプリケーションのキューの長さを決定したり、アプリケーションのワークフローを監視するための適切なメカニズムを提供します。

3.3 Notes ログと Domino ログ

問題点

Domino コンソールは、サーバー上でトランザクションが起こると、決まったメッセージ ストリームを表示します。実運用サーバーのほとんどでメッセージ ボリュームが非常に大きくなり、管理者がエラーや障害の状況をスクロールしながら注意深く見直すのは不可能です。幸い Domino はこれらのメッセージを Notes ログと Domino ログに記録します。

個々の Domino サーバー上でこれらのログを検索することは可能ですが、この作業は退屈で間違いの起こりやすいものです。この作業を自動化することによって、すべての Domino サーバー上に一様のエラー検知機能を提供することができます。

AppManager は Notes ログと Domino ログをどのように管理するか

AppManager は Notes ログからテキスト文字列を検索することができます。一般に、この検索パラメータはエラーまたは障害条件を示すテキスト文字列となります。ファイル内に指定された一連の正規式を使って、AppManager の知識スクリプトはこれらの検索パラメータと一致する文字列を含む/

含まないメッセージを検索します。これは、メッセージ ボリュームから AppManager Operator Console でレビューできる程度の部分集合を抽出する強力な 2 パス フィルタです。

また、Lotus Domino は Web 関連の動作も Domino ログに記録します。AppManager を使うと、HTTP アクセス リクエストの数を監視できるほか、転送されたデータのサイズも監視できます。 さらに、AppManager は許可のないアクセスの数や障害(見つからない)リンクをレポートすることもできます。

3.4 ルータと MTA

問題点

メッセージングは Domino の核となる機能です。電子メールが企業内で円滑に流れることは、Domino の管理者にとって最重要事です。ネイティブの Notes メール ルータに加えて、Lotus は MTA(Message Transfer Agent)を使ってその接続をまったく異なるメール システムに拡張しました。SMTP MTA を実行することは、インターネット経由でパートナや顧客と通信する必要のある企業にとって必須です。

AppManager はルータと MTA をどのように管理するか

Notes ドメイン内で AppManager の知識スクリプトは以下を実行することができます。

- さまざまなサーバー間のメール接続 を監視する。
- サーバー間のメールのルーティングに要した時間をレポートする。往復時間が長いのは通常サーバーに問題があることを示します。
- Domino メール サーバーのスループット(ルーティング、ペンディング、または無効メッセージの数など)を監視する。このデータは後で動向またはピーク時の負荷を調べることで把握できます。
- ルータおよび MTA キューを監視してメール システムのパフォーマンスを示す。キューに入れられたメッセージの数が設定したしきい値を超えた場合は、知識スクリプトからアラートが得られるため、エンドユーザーからのクレームを待たなくてもプロアクティブに問題に対応することができます。

3.5 Notes 統計

問題点

Domino サーバーの稼働状況とパフォーマンスを反映する統計が数多く用意されています。このデータはさまざまなソース(PerfMon、SNMP、または Notes API)から使用できます。

Microsoft の PerfMon ユーティリティは数台を超えるサーバーの監視には不十分です。SNMP はすべてのプラットフォームで広く利用できますが、連続的なポーリング メカニズムに頼っており、過剰なネットワーク トラフィックを招きかねません。

AppManager は Notes 統計をどのように管理するか

Notes 統計を監視し、しきい値を設定し、例外条件に対してアラームを生成する機能を提供できる、すなわち Domino ネットワークを効率よく管理できるのは AppManager だけです。

AppManager は Notes 統計をデータのソースとして使用する知識スクリプトを提供します。AppManager では *MailThruput(メールのスループット)*または *Connectivity(接続)*などといったわかりやすい名称を使って管理上の問題点に対応する適切な知識スクリプトを容易に選択できるようにしています。各知識スクリプトは、ユーザーが問題を検知し必要な通知または修正動作をトリガーできるようにデフォルトのしきい値セットを使って注意深く設計されています。また、AppManager は豊富なブーリアン(Boolean)式を揃えており、必要に応じてデフォルト設定をカスタマイズすることができます。

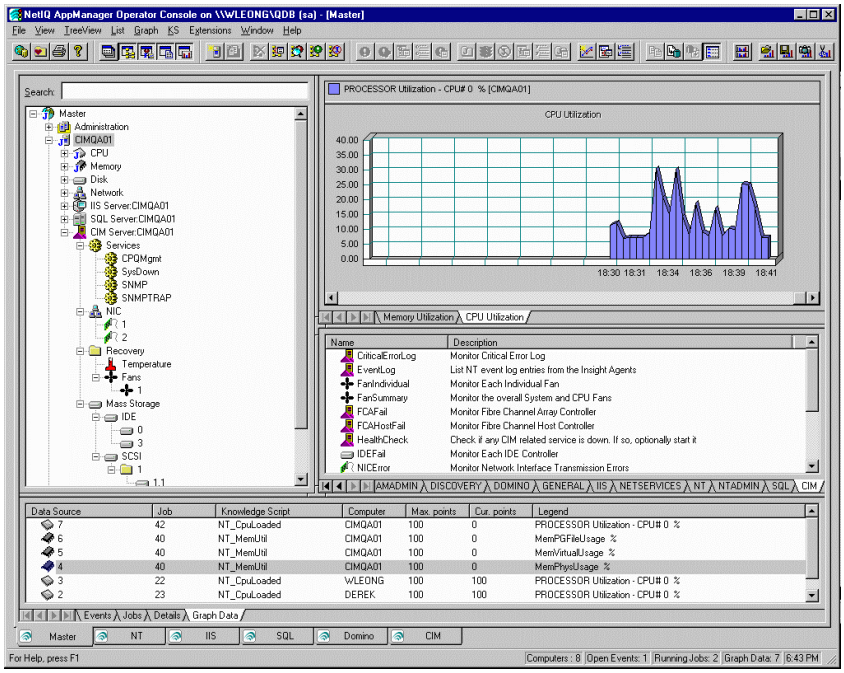
AppManager が知識スクリプトの中で明示的に使用していない Notes 統計を監視したい場合には、汎用目的の *GetStat* 知識スクリプトを使って指定したパフォーマンス データを監視することができます。AppManager はそのデータを AppManager リポジトリの中で把握することができます。このデータについては AppManager Operator Console でグラフに表示したり、AppManager Report Manager を使ってレポートを作成することもできます。

以下の表に Domino の管理者が興味を持つと思われる、代表的な Notes 統計をまとめます。

Notes 統計	説明
Database.BufferPool.Used	使用されているデータベース バッファ プールの数
Database.BufferPool.Peak	ピーク時の データベース バッファ プールの数
Database.BufferPool.Maximum	データベース コントロール プールの最大サイズ
Database.BufferPool.PerCentReadsIn Buffer	バッファ プール読込みのパーセンテージ
Domino.Command.Login	サーバーが最後に起動されてからサーバーに入ってきた「ログイン」URL の数
Domino.Command.Total	サーバーが最後に起動されてからサーバーに入ってきたすべての URL の数
Mail.AverageDeliverTime	メール送信の平均時間
Mail.AverageSizeDelivered	送信されたメールの平均サイズ
Server.AvailabilityIndex	サーバーのアベイラビリティの現在のパーセンテージ インデックス
Server.Sessions.Dropped	トランザクション中にドロップしたセッション
Server.Trans.PerMinute	最後の 1 分間のサーバー トランザクションの数
Server.Trans.PerMinute.Peak	1 分間に起こったトランザクションの最高数
XPC.COMX.Status	シリアル ポートのステータス

AppManager は、Domino の稼働状況やパフォーマンスに影響するのであれば、Domino 以外のメトリックスも監視します。たとえば、障害ディスク ドライブや他のアプリケーションからのメモリー リークは Domino のアベイラビリティに影響します。したがって、AppManager は Microsoft Internet Information Server や Microsoft SQL Server などの他のアプリケーションを監視します。さらに、以下の図に示すように Compaq Insight Manager を統合し、ハードウェアの問題を OS やアプリケーションの問題と関連付けすることができます。

分散 Windows NT 環境の稼働状況とアベイラビリティに影響を与えるハードウェア、オペレーティング システム、データベース、およびアプリケーションの総合的なビューを提供できるのは AppManager だけです。



3.6 サービス レベル要件の管理

問題点

ダウンの発生やパフォーマンスの劣化は、CEO から現場の作業者までおよぶ組織内の全員の業務に影響します。測定可能な目標を立て、エンドユーザー側の信頼を促進するために、多くの組織は IT 部門とサービス レベル合意(Service Level Agreement :SLA)を結ぶようになってきています。SLA は、たとえば Domino システムに対してエンドユーザーが期待する特定のサービス レベルを記述するものです。

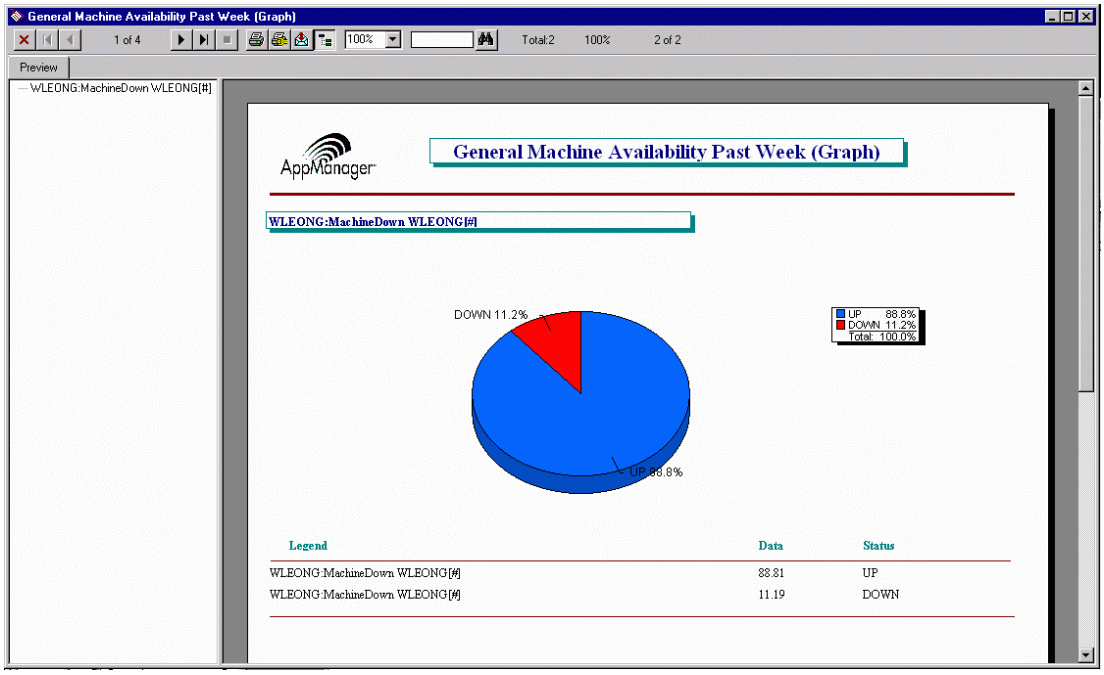
Domino の SLA は、Domino サーバーのアップ時間率、メール配信の保証レベル、応答時間の保証レベル、ディスクの使用可能領域の保証レベルなどが含まれます。このようなコミットメントを満たすためには、Domino サーバーの状態を継続的に監視することができる管理ツールが必要となります。また、管理ツールは、いずれかのサーバーがダウンした場合、このサーバーを自動的に再起動することができなくてはなりません。さらに、診断が必要な場合は、通知を発行することができなくてはなりません。このほか、管理ツールは、すべての Domino メール サーバー間のエンドツーエンドの接続性を監視できなければなりません。電子メールを配信して、同時に応答時間が SLA 仕様内かを測定できなければなりません。また、Domino に電子メールの送受信のために充分なディスク領域が確保されていることを保証しなければなりません。このほか、パフォーマンス データをレポジトリに収集する必要があります。これにより、システムのアベイラビリティ、サーバーの接続性、応答時間などを長期にわたって追跡することができます。

SLA を満たすためには、Domino サーバーの管理を維持し自動化する際に主要な統計データを継続的に監視する必要があります。また、現行の SLA コミットメントを満たしているかを検証するためのレポートも作成する必要があります。これに加えて、動向分析やキャパシティ プランニングのためのレポートを作成しなければなりません。これにより、将来にわたる SLA 要件に追随するシステム計画を立案することができます。

分散サーバーから収集されたデータは、容易にレポート作成や分析に使えなければなりません。このことは、もう 1 つのアーキテクチャ上の要件、すなわち一元管理されたレポジトリを示唆します。これこそ、データを容易に抽出し、リアルタイムまたは時系列の形式でレポートを提供するためのアーキテクチャなのです。

AppManager はサービス レベル要件をどのように満たすか

AppManager はレポート作成を考慮して設計されており、Microsoft SQL Server ベースの一元化された中央の管理リポジトリ上に構築されています。 これにより、AppManager または多くの市販のレポート作成ツールを使って、管理データの抽出とレポート作成が容易に行えます。AppManager は Domino、Windows NT およびその他の BackOffice アプリケーションですでに用意されたレポートを提供しているため、基本的な SLA 要件を満たすことができます。また、容易に追加のレポートを作成することができ、各企業固有の SLA 要件を満たすことができます。また、AppManager は自動化されたスケジューリングと配布メカニズムを用意しており、主要レポートの作成と配布を容易にします。AppManager による SLA レポートの例を以下に示します。



4.0 まとめ

Lotus Domino は、あらゆる規模の企業にとっての強靭な統合メッセージング インフラを提供します。しかしながら、Domino を導入することはほんの第 1 歩にすぎません。Domino サーバーへの投資から大きなメリットを引き出すには、ミッション クリティカルなアプリケーションのパフォーマンスと効率性を確保するための強靭なツールと方法論が必要となります。

NetIQ AppManager for Lotus Domino は、最大規模の分散 Domino 環境の稼働状態とパフォーマンスを管理する最も有効な統合ツールです。NetIQ AppManager for Lotus Domino はネイティブの Lotus ツー

ル群を補完するもので、イベント/障害管理、自動化、リアルタイム/時系列パフォーマンス監視といった、運用管理に焦点を絞って開発されています。

これに加えて、NetIQ AppManager Suite は Windows NT や、Microsoft Internet Information Server、Microsoft SQL Server などの他のサーバー アプリケーションを監視することもできます。さらに、Compaq Insight Manager との統合も可能です。AppManager は Lotus Domino の稼働状態に影響するすべてのシステムおよびアプリケーションのための総合的なビューを提供する唯一のソリューションです。

AppManager の詳細や評価版のお申込みについては、Visit NetIQ のホームページ http://www.netiq.com にアクセスしてください。