

アプリケーション監視の最適化

イントロダクション

N層のコンピューティングアーキテクチャがあちこちで使われるようになったように、ソフトウェアアプリケーションの可用性とパフォーマンスは、ハードウェアとネットワークおよびソフトウェアのコンポーネント間の複雑なネットワーク相互作用により以上に依存し、計測と監視や管理がより難しくなっている。同時に、特に電子商取引やITサービスマネジメントにとって、上手くコスト効果高くアプリケーション監視をすることの重要性は、どんどん大きくなっている。

本論分の第一部で、アプリケーション監視についての一連の実践的なテクニックを取り上げて説明する。第二部で、価値工学のための方法として、アプリケーション監視全体の設計について述べる。

アプリケーション監視の手法

アプリケーションのパフォーマンスと可用性を監視するのに一般的に使用されているいくつかの手法を下記にリストする：

- アプリケーション監視プラグイン
- マルチ計測コンテナ
- アプリケーション管理コンソールの"廃止"
- Web サービスベースアプリケーション
- 合成トランザクションモニタ
- Web サービスのサブスクリプション
- ポートベースの"GET"

これらの方法はそれぞれ、有効性、初期コストや運用コストがまったく異なる。最もミッションクリティカルなアプリケーションは、しばしば二つ以上の手法で監視される。これらの手法は、以下の項で説明するように、ホワイトボックス監視とブラックボックス監視の二つのカテゴリに分けられる。

アプリケーション監視のためのネットワーク・スニファァの利用は以下では触れない。しかしながら、本論文の第二部に記述される設計方法は、アプリケーション監視手法全体に同等に適用される。

ホワイトボックス・テスト

ホワイトボックス・テストでは、アプリケーション内部のソフトウェアコンポーネントの可用性と、時にはパフォーマンスを計測することができる。個々の計測は、アプリケーションの可用性とパフォーマンス全体について推定するために組み合わせたり、関連付けたりできる。

アプリケーション監視プラグイン

監視プラットフォーム（HP OpenView, CA Unicenter, IBM Tivoli, BMC Patrol その他）のほとんどは、特定のアプリケーションのソフトウェアがインストールされているすべてのサーバ上に配備されるアプリケーション監視エージェントのライブラリを持っている。特定のアプリケーションのためのプラグインは、監視サーバ上のコンソールに似たアプリケーションと通信を行う。そのプラグインはエージェントからの情報を関連付け、中央コンソールを通して、アプリケーションの可用性全体を表示する。これらのコンソールから抽出できるパフォーマンス情報は、オーナーやインストールした人が自発的に構成要素を元にしてアプリケーションのパフォーマンスをモデル化し、複雑に依存関係のあるネットワークを組み立てられようにする。

この手法には高価な監視ソフトウェアと複雑なアーキテクチャ特有の関連付けモデルの作成とメンテナンスが必要なので非常にコストがかかり、そのため通常は大きな先進企業でのみ、そしてその企業の最も重要なアプリケーションのみ使用される。さらに、その手法は、あまりに専門的で複雑であるため、サービスレベル管理で使用するためには信憑性が不足する。

マルチ計測コンテナ

多くの監視ツールは、異なるモニタの任意の集合をひとつの「コンテナ」あるいは、アプリケーションの名前を付けた仮想のホストに入れることができるようにしている。このコンテナは、監視されているある特定のアプリケーションの可用性と結びつく。代表的なコンテナをいくつか以下に示す：

- Windows サービスの可用性とアプリケーションプロセスを提供する SNMP
- Windows サービスの可用性とアプリケーションプロセスを提供する Perfmon
- アプリケーション固有の MIB のための SNMP
- アプリケーションプロセス状態情報のための UNIX シェルコマンドを実行するスクリプトベースのモニタ
- プロセス可用性のためのポートベース・スキャン
- ログファイルの構文解析

ほとんどの監視ツールは、これらの手法を使うよう設定できる。その考えは、もしコンテナの中のすべてのモニタがアラームでなければ、そのアプリケーションは活着していると考えられるということである。マルチ計測コンテナからパフォーマンスを計測するのは、現実的ではない。

アプリケーション管理コンソール廃止

今日使われているエンタープライズクラス・アプリケーションの多くは、それ自体のプロセス監視コンソールを装備しており、しばしば、Web サイトやセキュア Web サイトとしてアクセスできる。その管理サブシステムには、アプリケーション開発者がそのアプリケーションのために提供したホワイトボックスの内部監視チェックのセットが入っているので、明らかにその製品ごとの有効性は著しく異なる。実際には、アプリケーションすべてに監視コンソールが提供されている訳ではない。これらのコンソールは、しばしば広い可用性と未加工のパフォーマンス監視機能を提供することができる。

アプリケーション監視のためにシステム管理コンソールを使い、それが作り出した情報を集中監視機能にインポートすることはとても便利である。これは管理コンソールに HTTP で接続して、エラー状態があるかどうかを検出するため Web ページの HTML を構文解析することで実現される。検出された時には一般的なエラーメッセージを対応要員に送ることができる。

対応要員は、調査や修復作業のためのインシデントの詳細があるアプリケーション管理コンソールに保護されたアクセスをするために監視システムのコンソールを使う。

一般的に、この手法はシステム稼働上の問題を早急に検出するのに便利である。この手法の限界はアプリケーションとネットワークコンポーネント間の相互作用について考慮したり、直接計測したりしないことである。

Web サービス監視

Web サービスを使ったアプリケーション監視は刺激的で、業界標準のアプリケーションに対して急速に発展している領域である。簡単に言えば、その手法はエンタープライズアプリケーションによって提供される Web サービスを定期アクセスする“ソフトウェアのバス”を使うことから出来上がっている。これらのアプリケーションは、SOA プロトコルを使って XML 文書形式で関連する監視情報を発行する。その手法はすべてのホワイトボックス・テストの制約のいくつかを持っているであろうが、現在世代のホワイトボックス手法よりも容易にインストールし保守することができることが約束されている。

ブラックボックス・テスト

ブラックボックス・テストでは、アプリケーション全体が、エンドユーザやインタフェースを持つアプリケーションのアクションをシミュレートする外部のプロープからテストされる。

合成トランザクションモニタ

合成トランザクションは、アプリケーションを人が使うスピードでアクションをエミュレートするスクリプトを生成するためマクロレコーダを使って作成される。アラームはシミュレートされたトランザクションにかかる時間によって検出される。合成トランザクションは、実際に行われているトランザクションと同時に処理されるので、応答時間はその合成トランザクションが実行された時点で実行されているトランザクションの量とキャパシティの影響を受ける。

このため、その測定値はアプリケーション可用性の現実的な姿だけでなく、システムの応答パフォーマンスも提供する。行動の研究によると、ソフトウェアが実際にトランザクションを処理しているかどうかに関係なく、ユーザにとってパフォーマンスの低い状態はパフォーマンスしていないのと同じである。N階層トランザクションのパフォーマンスは、しばしばその経路に依存するので、すべてのユーザのためのアプリケーション・パフォーマンスを完璧に計測するには、複数ポイントから行う必要があるであろう。

合成トランザクションのマクロレコーダには二つのタイプがある。ひとつは重いWindowsクライアントであり、もうひとつはブラウザベースの軽いクライアントである。これらの機能は、多くの標準監視ツールで利用できるが、さらに一般的には、Mercury Interactive、Software Resarch、Rational、Quest、Compuwareなどが主導するソフトウェアテスト業界から提供されている。

ポートベースのスキャン

サーバの可用性を最も簡単かつ直接的にテストする方法がある。事実上、すべての監視システムは、アプリケーションにアクセスするために使うひとつの固有ポートを使って、getコマンドを実行するよう設定することが簡単にできる。たとえば、「URLゲットと文字列比較」モニタは、DNSと他のネットワーク・ユーティリティサーバ、サーバのNICカード、固有ポートのポートデーモン、OS、および単一のモニタ中でテストしているアプリケーションを働かせる。この形態のアプリケーションモニタは、影響の少ないアプリケーションや、複数フェーズにわたる配備によって、より包括的なモニタが追加される場合の一時的な代替としては適当である。

ホワイトボックス V.S. ブラックボックステスト

今日のエンタープライズアプリケーションにおけるN層ソフトウェアアーキテクチャは、複雑なシステムといえる。同様なシステムのコンポーネント間の相互作用があるだけでなく、特定のアプリケーションのコントロール下でないネットワークと共有ストレージを代表とする共有システム資源があるため、実際にはすべての他のシステムと相互作用が発生する。複雑なシステムでは、しばしばシステム要素と共有リソース間の繊細な相互作用が、目的とするシステムのパフォーマンスを左右する。この現実は、サービスレベル測定についてのホワイトボックステストの有効性の限界を示している。より現実的な言い方をすれば、ITの利用者は、提供されているサービスレベルの計測のために直接的なユーザ体感のシミュレーションを認める準備をしていると思われる。

他方、ホワイトボックステストは、システム障害の迅速な診断、切り分けおよび解決やパフォーマンスチューニングを行うため非常に有用な情報を提供する。多くのミッションクリティカルアプリケーションのため、ホワイトボックスとブラックボックステストの両方手法を採用することは価値があるであろう。

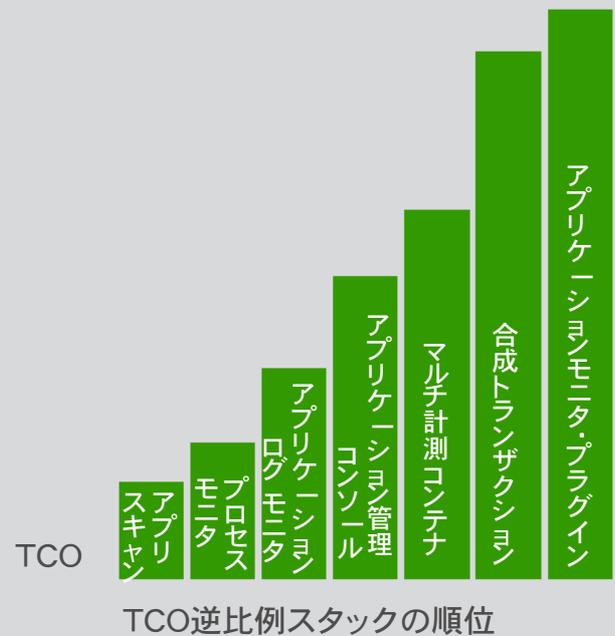
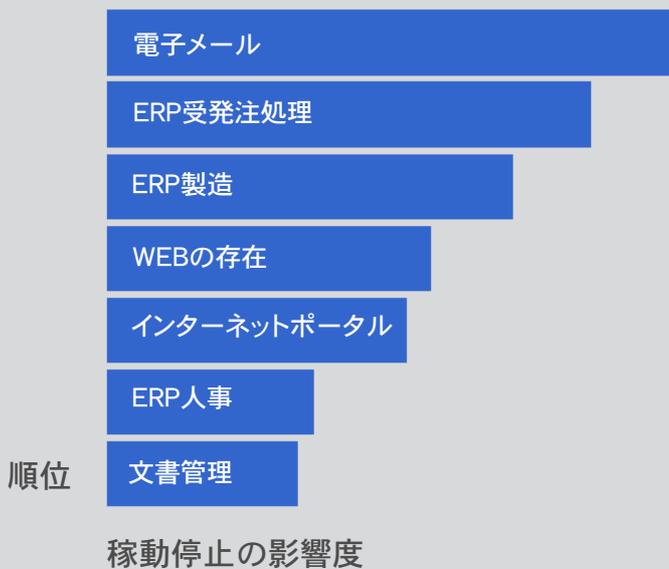
設計手法

アプリケーション監視設計の実際的な価値工学のための手順は、複雑で段階的に繰り返してアプローチするものである。このアプローチは、限界分析の概念を利用し、初期設置コストと同様に監視アプリケーションを維持するライフサイクルのコストを明確に盛り込む。アプリケーション設計手法の最初のステップは、すべての監視されるべき重要なアプリケーションを特定することである。このとき、そのアプリケーションを機能停止によるビジネスへの影響度合いでランク付ける。下図にそれを図解する：

このプロセスの次ステップでは、すでに配備されている、あるいは設計過程で購入すべきとされた監視ツールのために考慮できる手法のセットを選ぶ。このプロセスの一部でさまざまなアプリケーション監視方式を提供するために必要な新しいツールのパフォーマンスと購入コストを見積もる。概念的に、その結果は下記の図に示したように、相対的なコストに基づいたアプリケーション監視ツールのランク付けをした水平方向のスタックとなる。

図 1. アプリケーション機能停止のビジネスへの影響度

図 2. アプリケーション監視ツールのTCO



合成トランザクションのための監視ソリューションは、商用の監視ツールの「組み込み」マクロレコーダよりも、中央の監視システムのコントロール下で運用された無償の固定のマクロレコーダを使う方が良いことが経験上分かっている。それ故に、GroundWorkにおいて、このやり方のコストが、幾つかの監視システムに含まれているマクロレコーダを使うコスト見積もりよりも良いと評価する。次にインストールされたツールが求められるキャパシティを持っているかどうか、この設計選択肢を再考する。

合成トランザクションモニタの有効性は、実運用環境と同じ環境がリリース前テストおよび負荷テストで利用できる場合に大幅に向上する。ライセンス費用を節約することに加え、実運用監視からレポートされる診断情報は、QAと開発要員の両方にとって身近である。

ホワイトボックス監視ツールが組み込まれているプラットフォーム製品がすでにインストールされていて、アプリケーションを実現する「層」の数が比較的少ない場合は、ホワイトボックス監視ツールの方ががはるかに選ばれやすい。このことは、アプリケーション監視のために必要なネットワーク相互関係のインストールとメンテナンスを容易にする。アプリケーションが単一のサーバに含まれている場合が、マルチ計測コンテナが使われやすい。たとえば、マイクロソフトの Exchange 監視であればこの方法で行えるが、SAPインフラストラクチャに対してこの方法を試みるのは、恐れを知らない場合のみである。監視システムの最終設計は、さまざまな監視強度を持つ複数の監視手法を割り当てることに基準を置く。

最終設計において、最も影響を与えるアプリケーションでは(もしすでにライセンスが有れば)、HP Open View や CA Unicenter のような監視システムからホワイトボックスのプラグインと複数のテストトランザクションを持つ合成トランザクションと固有ポートへのポートベーススキャンを使うであろう。これに対して、中位の影響度のあるアプリケーションでは、マルチ計測コンテナに、単一の合成トランザクションとポートベーススキャンとを使うであろう。

最後に、さらに低い影響しかないアプリケーションでは、応答フィールドの文字列比較を行うポートベーススキャンのみを使うであろう。それらのアプリケーションのため、通常管理コンソールの提案と、エラーメッセージのためのコンソールの廃止が推奨される。アプリケーション監視手法の適用結果は、コストと有用性との最適な関係を提供することで明らかになるであろう。

結論

何十ものアプリケーション監視設計の研究によって、しばしば、簡単な合成トランザクションモニタと、簡単で重要でないアプリケーションのためのマルチ計測コンテナを組み合わせることによって、最適なソリューションが実現されることが分かった。このソリューションの費用対効果は、N階層のアプリケーションのリリーステスト、負荷テストの監視で同じツールが使われた場合に実質的に大幅向上する。我々は、複雑なホワイトボックステスト手法は主として診断に有用であることに気付いた。

よく管理されたプロジェクトに加えて、これらの原則をアプリケーション監視に適用することで、著しいビジネスメリットを勝ち取ることができる。これらには、高いアプリケーション稼働率と従業員生産性、および資産と運用コストの低下が含まれる。高価な監視ソフトウェアを使うかオープンソースを使うかに関わりなく、監視システムの価値は、その設計と実施計画および、その結果を使うビジネスプロセスに由来する。ソフトウェアの機能ではない。

多くの企業にとって、これは良いニュースである。

GROUNDWORK について

GroundWork Open Source, Inc. は、ネットワークとシステム監視、サービスデスク管理およびITダッシュボードのような、オープンソースを基盤としたITインフラストラクチャソリューションを提供します。

GroundWork のソリューションは、IT マネージメントにおいて、柔軟で低コストのオープンソースツールを活用することにより、商用ソフトウェアのほんの一部のコストでエンタープライズレベルの可用性、パフォーマンスおよび運用効率性を達成します

コンタクト情報

866.899.4342

info@groundworkopensource.com

www.groundworkopensource.com

GroundWork Open Source, Inc.

139 Townsend Street, Suite 100

San Francisco, CA 94107