

大規模時系列データの省資源監視のための 動的ポーリング制御

藤山 健一郎 今井 照之 喜田 弘司 中村 暢達

サーバ管理のために CPU 負荷等の時系列情報を監視することが求められている。監視方式の一つに、毎回、監視対象サーバに接続して監視コマンドを発行してポーリングする毎回接続という方式がある。しかし、毎回接続方式は接続処理が重く低速なため、大規模で高品質な監視が行えない。一方、コネクションプーリングを用いて接続負荷を軽減する接続維持という方式がある。しかし、接続維持方式は高速だが、多くの資源を消費する。そこで本稿では、限られた資源で大規模・高品質な監視を実現する動的ポーリング制御方式を提案する。提案方式は、監視対象の状況に応じて監視頻度を制御し、さらに監視周期に応じて監視対象への接続方法を高速だが高資源消費の毎回接続と低速だが低資源消費の接続維持とで切り替える。このように必要な状況のときのみ高頻度に監視するようにすることで、省資源と大規模・高品質な監視を両立させた。

A Dynamic Polling Method with Small Memory for Large-scale Server Performance Monitoring

Ken-ichiro FUJIYAMA Teruyuki IMAI
Koji KIDA Nobutatsu NAKAMURA

For a server management, it is required to monitor time series data such as CPU load. There is a sequential connecting method that connects to a target server, starts monitoring commands and receives polling data periodically. This method cannot achieve large-scale and high-quality monitoring because common connection processes are slow. On the other hand, there is a connection pooling method that is fast because it decreases a connection process load. However, it consumes a large amount of memory. We propose a dynamic polling technology for large-scale and high-quality monitoring with a restricted memory. At first, it controls a monitoring frequency according to a state of the target server. Then it switches the connecting method to the target server by the sequential connecting method or the connection pooling method according to the monitoring frequency. As above, the proposed technology frequently monitors target servers only when they should be monitored. Therefore, we achieved a balance between quality and memory size on a large-scale server system monitoring.

1. はじめに

大規模なデータセンターでは、サーバの集中管理等のために、大量のサーバの性能情報を収集、監視したいというニーズがある。性能情報とは CPU 使用率、メモリ消費量といった、時々刻々と変化する時系列データであり、監視の主な目的はこれら時系列データの異常の検出（アラート）や状態の記録（ロギング）である。監視方式は、大きくポーリング方式とメッセージドリブン方式の二通りに分けられる[1]。

- ・ **ポーリング方式**： 監視サーバから監視対象サーバに対しアクセスし、性能情報を収集する。SNMP[2]によるポーリングや、SSH[3]等でサーバに接続して監視コマンドを発行する方法等がある。

- ・ **メッセージドリブン方式**： 監視対象サーバに監視エージェントを配置し、特定イベント発生時に性能情報を監視サーバに送信する。SNMP のトラップ等が相当。

既存のデータセンターに監視機能を追加する場合、監視対象サーバでは既に何らかのサービスが複数稼動しており、そこに新たに監視エージェントを追加すると、余計な負荷をかけたたり、既存サービスに影響を与える可能性がある。また、監視対象サーバの数が多く監視エージェントの導入そのものが高コストのため、メッセージドリブン方式はデータセンター事業者等からは一般的に好ましく思われていない。そこで我々はエージェントを用いないポーリング方式、特に SSH で様々な監視コマンドを発行することで柔軟な監視を行える SSH ポーリング監視の研究を行っている。

SSH ポーリング監視では、監視を実行する都度に監視対象サーバに SSH で接続し、監視コマンドを発行し、切断するという処理を行うが、接続、切断の負荷、特に接続時の認証負荷が大きい。そのため、従来の監視スループットは毎分数百台が限界であった。今後、データセンターの大規模化が進み監視対象サーバの数がさらに増えた場合、監視スループットが低いと全体を監視するのに時間がかかり、素早いアラートやきめ細かいロギング、すなわち高品質な監視ができなくなるという問題がある。

そこで本稿では、ポーリング時の監視周期や接続方法を動的に制御することで、大規模な時系列データを効率よく収集、監視する方式について述べる。以下、2 章では、大規模監視の既存技術とその課題について説明する。3 章では課題を解決する方式を提案する。4 章では提案方式を評価する。最後に 5 章で結論を述べる。

2. 既存技術とその課題

SSH ポーリング監視による接続時の認証負荷を軽減し、高い監視スループットによる高品質監視を実現する方法としてコネクションプーリング[4]やキープアライブ[5]の技術を応用した方法が考えられる。通常の SSH ポーリングでは、監視を行う都度に、

監視対象サーバに接続 監視コマンド発行 切断という処理を繰り返す。これを毎回接続監視という。一方、コネクションプリーングでは、監視対象サーバに一度接続したら以降切断せず、その接続を維持し続ける。そして維持した接続を用いて適宜監視コマンドの発行を繰り返すことで、すなわち、接続 監視コマンド発行 監視コマンド発行...とすることで、接続時の認証負荷を、最初の1回のみ削減する。これを接続維持監視という。接続維持監視により高い監視スループットを実現する。しかし、コネクションプリーングは、監視対象サーバ毎に接続を維持し続けるために監視サーバのメモリ等の資源を消費する。そのため1台の監視サーバが同時に監視できる監視数に上限があるという問題がある。

この問題を解決する方法として、階層化監視[6]という方式がある。図1に示すように、階層化監視のシステムは実際に監視を行う複数のブランチサーバとブランチサーバの監視結果を取りまとめるルートサーバから構成される。

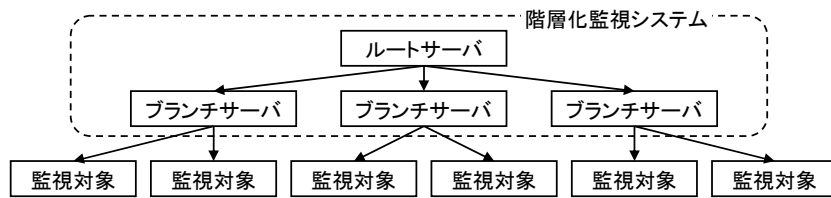


図1 階層化監視

1ブランチサーバ辺りの監視数には上限があるが、ルートサーバには複数のブランチサーバを接続できるため、システム全体ではサーバ単体の監視数上限を超えてコネクションプリーングによる高品質監視を実現できる。しかし階層化監視は、ブランチサーバとして複数の物理サーバを必要とする。本来、データセンター等にとって監視は副次的な機能であり、多くの資源は割けない。そこで我々は、限られた資源で大規模かつ高品質な監視を実現する、動的ポーリング制御による省資源監視を提案する。

3. 方式提案

動的ポーリング制御は、省資源で高品質な監視を実現する動的監視周期制御と、省資源で大規模な監視を実現する動的接続方法制御の2つから成る。

3.1 動的監視周期制御

前述のとおり、監視の主な目的は時系列データである性能情報のアラートとロギングである。従来の監視では性能情報を一定周期で監視していた。高品質な監視を行う

ためには、より高頻度、すなわち短周期で監視を行う。しかし、アラート目的なら異常が発生しそうな時のみ短周期で監視すればよく、ログ目的なら状態が変動しそうな時のみ短周期で監視すればよい。図2に示すように、アラート目的の場合、性能情報の異常を判定する閾値を超えるような異常に対しては、異常が発生する近辺のみ短周期に監視しておけば、常に短周期に監視する場合と同程度の速さで異常を検出できる。また、ロギング目的の場合も、変動が激しい場合のみ短周期に監視すれば、常に短周期に監視する場合と、さほど変わらないグラフが得られる。むしろ、必要なときのみ、より短周期に監視することで、常時ある程度短周期に監視するよりも高速に異常を検出したり、監視誤差が少なく高精度に記録したりと、より高品質な監視を行える。

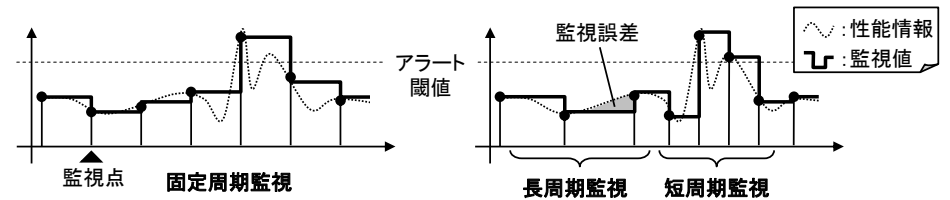


図2 監視周期

動的監視周期制御で重要となるのは、性能情報を、どれ位の周期で監視すべきか判定することである。今回はアラートとロギングという二つの目的に合わせ、以下の2つの場合に、特に短周期で監視する必要があると考えた。

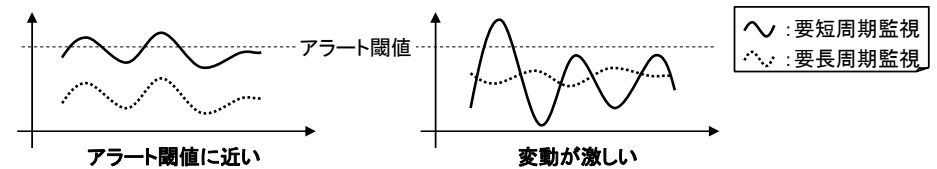


図3 短周期で監視する必要がある場合

・アラート閾値に近い場合：

性能情報がアラート閾値に近い場合、閾値を超えて異常となる可能性が高いため、より短周期に監視する。アラート閾値との近さは、これまで一定期間の性能情報の平均値がアラート閾値にどれだけ近いかで判定する。具体的には以下のような指標 Idx_1 を用いて判定する。一定期間内の n 個の性能情報の時系列データを V_1, V_2, \dots, V_n 、性能情報の最大値を V_{max} 、アラート閾値を $A (0 < A < V_{max})$ とする。各性能情報、およびアラート閾値を V_{max} で正規化すると $V1/V_{max}, V2/V_{max}, \dots, Vn/V_{max}$ 、および A/V_{max} となる。

n 個の性能情報の時系列データの平均を $V_{ave} = (V_1 + V_2 + \dots + V_n) \div n$ すると、正規化した平均値とアラート値の差分 $= A / V_{max} - (V_1 / V_{max} + V_2 / V_{max} + \dots + V_n / V_{max}) \div n = (A - V_{ave}) \div V_{max}$ となる。この値が小さいほど性能情報の平均値がアラート閾値に近く、短周期に監視する必要がある。後述する別の指標と揃えるため、値が大きい程短周期に監視するように $1 -$ を指標 Idx_1 とする。

・変動が激しい場合：

性能情報の変動が激しい場合、閾値を超える可能性も高いうえ、その変動をより高精度に記録する必要があるため、より短周期に監視する。変動の激しさは、これまで一定期間の性能情報の標準偏差がどれだけ大きいかで判定する。具体的には以下のような指標 Idx_2 を用いて判定する。一定期間内の n 個の性能情報の時系列データを V_1, V_2, \dots, V_n 、性能情報の最大値を V_{max} 、平均を V_{ave} とすると、 V_{max} で正規化した性能情報の分散は $\sigma^2 = 1/n \cdot (V_{ave}/V_{max} - V_i/V_{max})^2 = 1/n V_{max}^2 \cdot (V_{ave} - V_i)^2$ となる。この分散 σ^2 の平方根である標準偏差 σ を指標 Idx_2 とする。この値が大きい程性能情報の変動が大きく、短周期に監視する必要がある。

上記の指標を元に、今回は監視周期 T を以下のように決定する。なお、 I_{max} は指標 Idx_1, Idx_2 のうちより大きい方、 T_{min} は最短監視周期、 T_{max} は最長監視周期、 α は指標の監視周期への影響度の重みである。

$$T = T_{max} \times (1 - I_{max} \times \alpha)$$

$$\text{if } T < T_{min} \text{ then } T = T_{min}$$

$$\text{if } T > T_{max} \text{ then } T = T_{max}$$

動的監視周期制御により、性能情報の状態に応じた適切な周期で監視を行うため、監視の品質を落とすことなく不要な監視による資源の浪費を削減できる。

3.2 動的接続方法制御

動的接続方法制御は、監視実行時の監視対象サーバへの接続方法を、毎回接続と接続維持とで随時切り替える。各接続方法による監視は、以下のような特徴がある。

- ・**毎回接続による監視**：接続時の認証処理負荷等のため監視スループットが低く低速だが、監視後には接続を切断するため、監視サーバの資源を消費し続けることが無く省資源であり、多くのサーバを監視できる。
- ・**接続維持による監視**：接続時の認証処理負荷等が少ないため監視スループットが高く高速だが、監視後に接続を維持するために資源を消費するため、多くのサーバを監視できない。

そこで、動的監視周期制御により決定された監視周期がある閾値より短ければ、高速な監視が必要と判定し、接続維持による監視を行うようにする。逆に監視周期が周期閾値より長ければ監視速度が遅くても良いと判定し、毎回接続による監視を行う。動的接続方法制御により、監視周期に応じた適切な接続方法で監視を行うため、必要な監視スループットを確保しつつ、不要な資源の浪費を削減できる。

以上の2つの制御を組み合わせることで、監視の品質を落とすことなく、省資源で十分な監視スループットを保つ、すなわち高品質で大規模な監視を行うことができる。

4. 評価

動的ポーリング制御を構成する動的監視周期制御と動的接続方法制御の2つについて、それぞれ評価を行う。

4.1 動的監視周期制御の評価

4.1.1 評価目的

動的監視周期制御は、監視周期を変えることで、監視の品質を落とすことなく省資源で監視できることを特徴とする。そこで、ある性能情報を監視する際に、監視周期を固定した場合とで、監視誤差と監視回数を比較する。監視誤差とは、図2に示す監視対象である性能情報の実際の値と、監視値の差分であり、小さいほど監視品質が高い。監視回数は監視を行った回数であり、少ないほど資源消費が少ない。

4.1.2 評価方法

事前に収集した12種類のサーバのCPU使用率の変動に対し、固定周期による監視と動的監視周期制御による監視とで監視誤差と監視回数をそれぞれ測定、比較する。固定周期による監視は、監視周期を60秒、90秒、120秒に固定した3パターンで測定した。動的監視周期制御による監視は、最大監視周期や指標の監視周期への影響度の重みを様々に変更した複数のパターンで測定した。動的監視周期制御による監視を複数のパターンで測定した理由は、固定周期による監視の各パターンと比較する際に、監視誤差を比較するなら監視回数が似たパターンで、監視回数を比較するなら監視誤差が似たパターンで比較するためである。

4.1.3 評価結果

監視誤差と監視数の測定結果を示す。図4は監視回数が近いパターンで監視誤差を比較、図5は監視誤差が近いパターンで監視回数を比較したものである。

動的監視周期制御による監視は図4に示すとおり、監視回数、すなわち資源消費が同程度であれば監視誤差は3~8%少ない、すなわち高品質である。また、図5に示すとおり、監視誤差、すなわち監視品質が同程度であれば監視回数が13~25%少ない、すなわち省資源である。したがって、高品質と省資源を両立させている。

固定周期		動的監視周期制御			差分比	
周期	誤差	回数	周期、重み	誤差		回数
60秒	2.31	1432	30-120秒, $\alpha=10$	2.13	1381	-7.8%
90秒	2.58	955	30-120秒, $\alpha=3$	2.39	930	-7.5%
120秒	2.70	716	30-150秒, $\alpha=2$	2.63	703	-2.7%

図4 監視誤差の比較

固定周期			動的監視周期制御			差分比
周期	誤差	回数	周期、重み	誤差	回数	
60秒	2.31	1432	30-120秒, $\alpha=5$	2.31	1068	-25.4%
90秒	2.58	955	30-180秒, $\alpha=5$	2.61	735	-23.0%
120秒	2.70	716	30-180秒, $\alpha=3$	2.69	623	-13.0%

図 5 監視回数の比較

4.2 動的接続方式制御の評価

4.2.1 評価目的

動的接続方法制御は、毎回接続と接続維持を切り変えることで、省資源で大規模な監視ができることを特徴とする。そこで、大量のサーバを監視する際に、すべて毎回接続の場合、およびすべて接続維持の場合と、資源消費量と監視時間を比較する。資源消費量が少ないほど省資源であり、監視時間が短いほどより大規模な監視ができる。

4.2.2 評価方法

監視周期要件の異なる 5 種類の監視対象サーバに対して、異なる 3 種類の接続方法で監視を行い、資源消費量（今回はメモリ消費量）と監視時間を測定、比較する。監視対象サーバは図 6 に示す 5 種類で、各種 300 台ずつ、合計 1500 台である。各監視対象サーバは、その性能情報の変動から動的監視周期制御により 60 秒～300 秒の間でそれぞれ適切な監視周期が判定されているとする。一方、接続方法は図 7 に示す 3 種類で、事前実験により、それぞれの 1 台あたりの監視実行に要する時間、およびメモリ消費量を測定してある。なお、本評価実験環境においては、総メモリ消費量が約 400MB を超えると、監視サーバ（物理メモリ 512MB）がスワップを起こし、極端に反応が遅くなり、場合によってはフリーズして監視を実行できなくなる。

監視対象	監視周期	台数
サーバ1	60秒	300台
サーバ2	120秒	300台
サーバ4	180秒	300台
サーバ5	240秒	300台
サーバ6	300秒	300台

図 6 監視対象サーバ

接続方法	監視時間	メモリ消費量
毎回接続	0.231秒/台	113KB/台
接続維持	0.007秒/台	351KB/台
動的接続方式制御	上記いずれか	上記いずれか

図 7 接続方法

4.2.3 評価結果

すべて毎回接続で監視した場合、全サーバを監視するのに必要なメモリ消費量は

113KB/台 × 1500 台 = 169.5MB、監視時間は 0.231 秒/台 × 1500 台 = 347 秒となり、メモリ消費量は問題ないが、監視時間が間に合わない。

すべて接続維持で監視した場合、全サーバを監視するのに必要なメモリ消費量は 351KB/台 × 1500 台 = 556.5MB、監視時間は 0.007 秒/台 × 1500 台 = 10.5 秒となり、監視時間は間に合うが、メモリ消費量が大きく、監視を実行できない。

動的接続方法制御の場合、サーバ 1～3 は監視周期が短いため接続維持による監視、サーバ 4～5 は監視周期が長い毎回接続による監視を行うと仮定する。すると、全サーバを監視するのに必要なメモリ消費量はサーバ 1～3 は接続維持のため 351KB/台 × 900 台 = 351.9M、サーバ 4～5 は毎回接続のため 113KB/台 × 600 台 = 67.8MB となり、合計 383.7MB のため問題ない。一方、監視時間はサーバ 1～3 は接続維持のため 0.007 秒/台 × 900 台 = 6.3 秒となり、最も監視周期の短いサーバ 1 でも十分間に合う。また、サーバ 4～5 は毎回接続のため 0.231 秒/台 × 600 台 = 138 秒となり、サーバ 4～5 では十分間に合う。したがって、省資源と大規模監視を両立させている。

5. まとめ

本稿では、限られた資源で高品質かつ大規模な監視を実現する動的ポーリング制御による省資源監視について述べた。本手法は、監視対象サーバの性能情報の変動に応じてポーリングの監視周期、接続方法を適切に制御することを特徴とする。本手法により監視周期要件の異なる 1500 台の監視対象サーバに対し、各監視周期要件を満たす十分な監視品質を維持しつつ、従来手法に比べメモリ消費量を約 30% 削減した省資源監視を実現した。

なお、全ての監視周期要件が短周期の場合、資源不足か監視周期要件を満たせなくなる。その場合、監視対象に応じて優先度を付けて、限られた資源内でベストエフォートの監視する等の対応が考えられる。今後、これらの手法について検討を進める。

参考文献

- 1) 宮元洋輔: Hinemos によるネットワーク統合管理のすべて, Software Design 208 号, 技術評論社 (2008).
- 2) J.D.Case, et al.: A simple network management protocol(SNMP), RFC 1157 (1990).
- 3) D. J. Barrett: SSH The SecureShell, O'Reilly Media Inc. (2001).
- 4) 山本秀樹: 最新 Servlet&JSP 徹底入門 Chapter9 データベースとの接続, OPEN DESIGN, 2001 年 9 月号, CQ 出版社 (2001).
- 5) 杉本章義 他: リクエスト待機間隔を考慮したウェブサーバの leap-alive 時間の自動設定, コンピュータソフトウェア 24(2), 日本ソフトウェア科学会 (2007).
- 6) 安達惇 他: 無線ライフライン情報監視システム, 電子情報通信学会 通信ソサエティ大会, B-5-161 (1998).