



Pandora FMS の技術情報



pandorafms.com/manual/!current/

ent link:

pandorafms.com/manual/!current/ja/documentation/pandorafms/complex_environments_and_optimization/09_pandorafms_engineering

3/04 21:28



Pandora FMS の技術情報

[Pandora FMS ドキュメント一覧に戻る](#)

Pandora FMS のデータベース設計

0.83 から 1.1 までの Pandora FMS **初期バージョン**では、データベースに一つのデータに対して一つの挿入を行うという、とてもシンプルな考えに基づいていました。この方法は、開発がとても簡単で、簡単な検索や挿入などの操作をプログラムしやすくなっていました。

これは、多くの利点がありましたが、スケーラビリティに関しては大きな問題をもっていました。このシステムでは、サポートできるモジュールの最大数にある上限があり、一定数のデータで負荷が増大しクラスタリングの仕組みを実装しにくく、(5万を超える項目では) 処理も早くはありませんでした。

MySQL クラスタをベースにしたソリューションは簡単ではなく、常に何らかの問題が発生していました。そして改善に長時間を要しました。

現在のバージョンの Pandora FMS では、データの挿入におけるリアルタイムでのデータ圧縮を実装しています。それは、収集間隔に基づいたデータ圧縮を行います。また、**メンテナンスタスク**にて特定の日付のデータを自動的に削除するような実装も行っています。

新たなデータ処理システムは、新しいデータのみを保持します。同じ値がシステムに入力された場合、それはデータベースに保存されません。これはデータベースを小さい状態に維持するのにとても便利です。これは、数値、インクリメンタルな数値、ブーリアン等のすべての Pandora FMS モジュールで動作します。

これらの変更により、データの読み取りと解釈に大きな変化が生じます。Pandora FMS の最新バージョンでは、新しいデータストレージモデルでデータをすばやく表示できるように、グラフィックエンジンが最初から再設計しました。

圧縮メカニズムは、データをグラフィカルに読み取って解釈するときにも **特定の影響**を及ぼします。現在、パーセントイル、リアルタイムデータ、イベントやアラートが発生した日時、その他のオプションを追加できるグラフ設定メニューがあります。

さらにPandora FMS ではコンポーネントを完全に分離できるため、異なるサーバーのデータ ファイル処理とネットワーク モジュールの実行の負荷を分散できます。

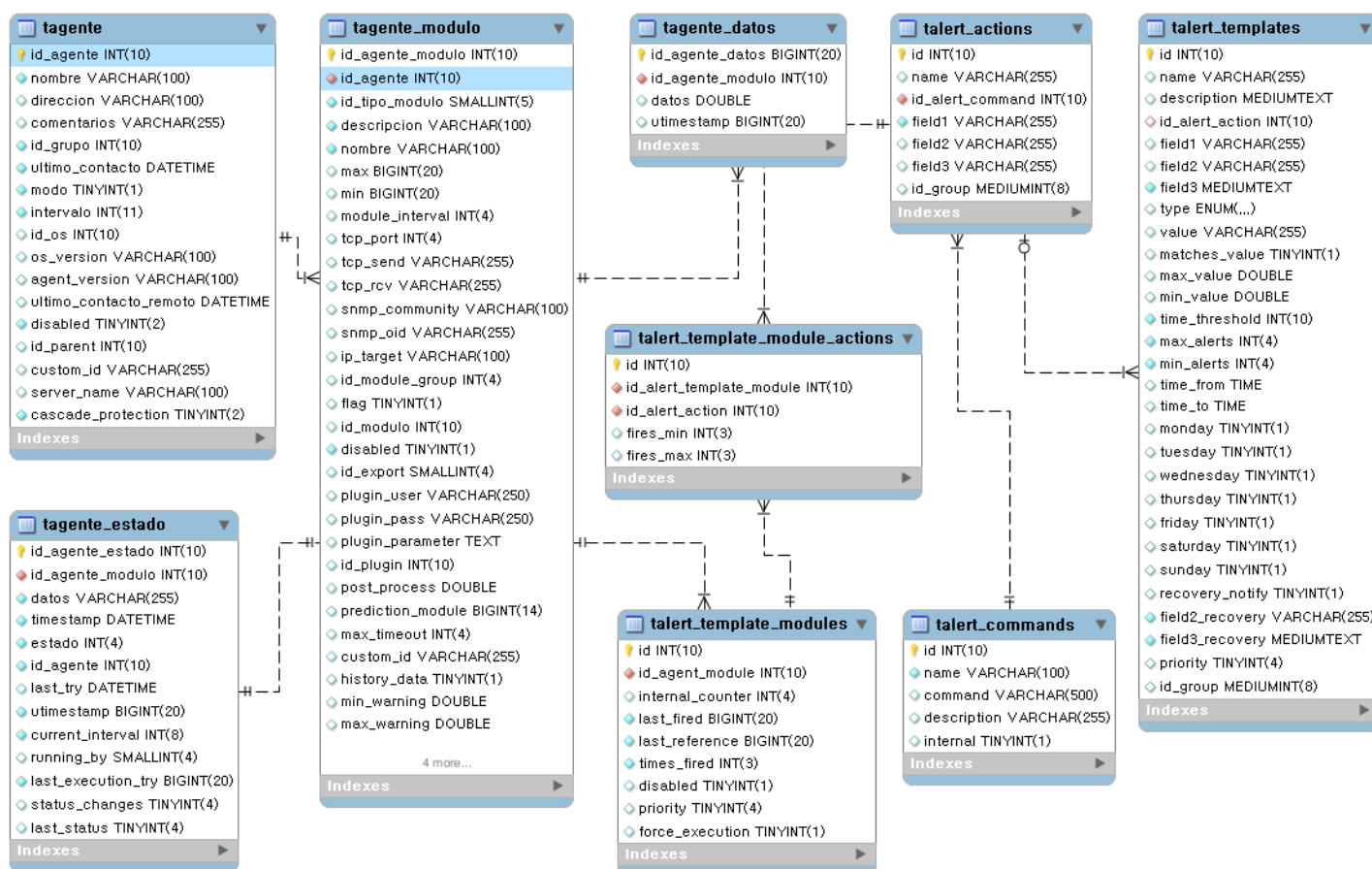
データベースにおけるインデックスの改良とその他技術的側面

ソフトウェアアップデートを通じてPandora FMS データベースのリレーショナルモデルに改善が実装されました。導入された変更の1つは、異なるタイプのモジュールに基づいて情報をインデックス化することです。これにより、情報が異なるテーブルに分散されるためPandora FMS は情報に非常に速くアクセスできます。

テーブルを（タイムスタンプ別に）パーティション分割して、履歴データへのアクセスのパフォーマンスをさらに向上させることが可能です。

さらに、タイムスタンプの数値表現 (UNIX 形式 `_timestamp_`) などの要素により、日付範囲の検索、日付の比較などが高速化されます。この作業により、検索時間と挿入が大幅に改善されました。

データベースのメインテーブル

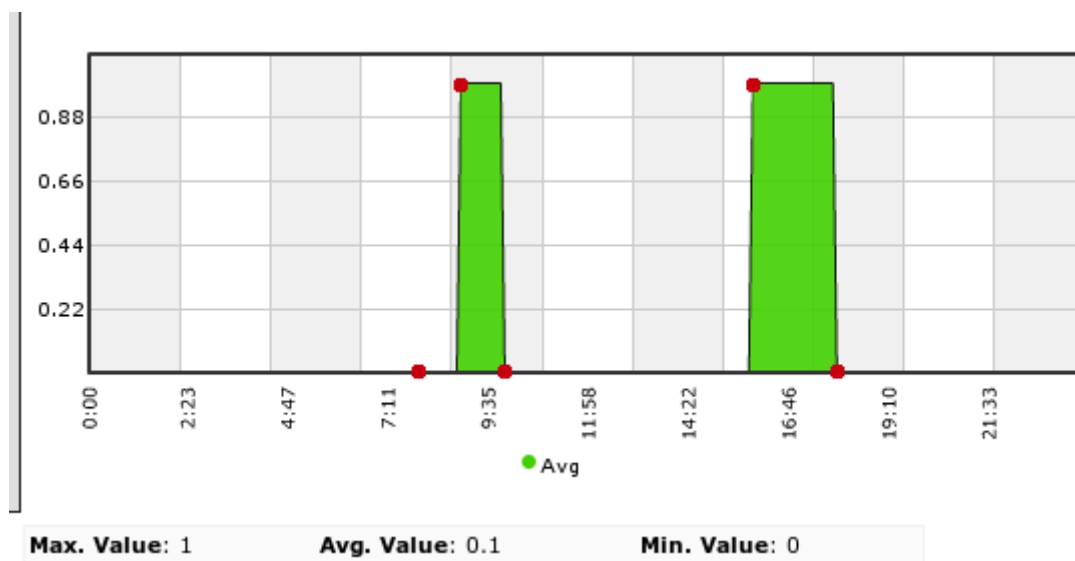


リアルタイムでのデータ圧縮

データベースの高負荷を避けるために、データ挿入時にサーバは簡単な圧縮を行います。データが一つ前のデータと同じ場合や、24時間以内に変化が内場合は、データベースに保存されません。

例えば、だいたい1時間の間隔で 0,1,0,0,0,0,0,0,0,0,1,1,0,0,0 というデータがあった場合、データベースには、0,1,0,1,1,0 と記録されます。24時間経過しないと、連続した0は記録されません。

赤で示したデータのみがデータベースに保存されます。



圧縮は、データ処理のアルゴリズムに影響を与えます。グラフ描画では、圧縮により発生した空白のデータを埋める必要があるということを認識することが重要です。

以上を理解して、あるモジュールの実行間隔と最初のデータを使って、そのモジュールのデータを求めるためには、次のような手順を踏む必要があります。

- 指定した間隔や日時よりも古いデータを検索します。見つかった場合は、それが現在の値になった最初のデータです。見つからなかった場合は、過去にデータは存在しません。
- モジュールの実行タイミングと同じ日時まで、指定した間隔や日時よりも新しいデータを検索します。見つかった場合は、それが値が変化したタイミングのデータとなります。もし、見つからなかった場合は、最後に記録された値が現時点まで続いていることとなります。
- あるデータを特定するためには、それが可能となる他のデータを全て確認する必要があります。

データの削減

Pandora FMS は、データベースに保存する情報を削減するシステムを持っています。これは中小規模のシステム(250-500エージェント、100,000未満のモジュール数)で、精度を落として長期間のデータ保存を想定したものです。

1時間ごとに実行される Pandora FMS データベースのメンテナンスでは、他のクリーニングタスクとともに、古いデータの圧縮を実行できます。この圧縮は、補間であり、単純な線形補間が使用されます。このため、この情報の詳細は失われますが、レポートや月次、年次グラフなどを生成するには十分な情報となります。

大規模なデータベースでは、この動作はパフォーマンスの面で非常にコストがかかる可能性があるため、無効にする必要があります。代わりに、ヒストリデータベースモデルを選択することをお勧めします。

ヒストリデータベース

ヒストリデータベース は、1 か月以上前のデータなど、最近の表示では使用されない過去のすべての情報を保存するために使用される機能です。このデータは、メインデータベースとは異なるストレージ (ディスク) を持つ別のサーバ上にある必要があります別のデータベースに自動的に移行されます。

古いデータを含むグラフやレポートを表示する場合、Pandora FMS は最初の数日間はメインデータベースを検索し、ヒストリデータベースを参照する必要がある時点で達するとヒストリデータベースを検索します。これにより、システムに大量の情報が蓄積されている場合でも、パフォーマンスが最大化されます。

詳細設定

Pandora FMS のデフォルト設定では、文字列 テキスト文字列タイプのデータはヒストリデータベースに転送されませんが、この設定が変更され、ヒストリデータベースがこのタイプの情報を受信している場合は、消去を設定することが不可欠です。そうしないと、占有量が大きくなり、大きな問題が発生し、パフォーマンスに悪影響が及ぶことになります。

このパラメータを設定するには、データベース内で直接クエリを実行して、この情報が消去される日数を決定する必要があります。問題のテーブルは `tconfig` とフィールド `string_purge` です。このタイプの情報の消去を 30 日間に設定するには、次のクエリを履歴データベース内で直接実行します。

```
UPDATE tconfig SET VALUE = 30 WHERE token = "string_purge";
```

データベースは `pandora_db.pl` というスクリプトによってメンテナンスされます。データベースのメンテナンスが正しく実行されているかどうかを確認するために、メンテナンス スクリプトを手動で実行します。

```
/usr/share/pandora_server/util/pandora_db.pl /etc/pandora/pandora_server.conf
```

エラーは報告されません。別のインスタンスがデータベースを使用している場合は、実行を強制する `-f` オプションを使用することもできます。 `-p` パラメーターを使用すると、データは圧縮されません。これは、スクリプトがコンポーネントに対して必要な手順を正しい順序と方法で確実に実行するため、ヒストリデータベースを備えた HA 環境で特に役立ちます。

Pandora FMS におけるモジュールの状態

状態はいつ変更されるのか

- それぞれのモジュールには、警告 や 障害 状態のしきい値の設定があります。
 - これらのしきい値は、データの値によってそれぞれの状態になるということを定義します。
 - モジュールのデータがこれらのしきい値の範囲から外れれば、正常状態と認識されます。
- それぞれのモジュールには、データを収集する間隔の設定もあります。
 - データ収集の間隔はコンソールで確認されます。
 - もし、モジュールが収集間隔の 2 倍の期間データを収集できなかった場合、そのモジュールは不明状態となります。
- 最後に、モジュールにアラート設定があり、アラートが発生し承諾されていなければ、モジュールはアラート発生状態となります。

伝播と優先度

Pandora の仕組みにおいては、いくつかの要素は他に依存しています。たとえば、一つのエージェントにおけるモジュールや、一つのグループにおけるエージェントです。これらはまた Pandora FMS ポリシーに適用することができます。ポリシーは、個々のエージェントに割り当てられたもののように、いくつかのエージェントおよびモジュールを関連付けることができます。

この構造は、モジュールの状態を簡単に評価するには特に便利です。これにより、エージェント、グループ、およびポリシーの状態を仕組みの中で伝播していくことができます。

エージェントがとりうる状態

エージェントの状態は、モジュールの状態の中で最も悪い状態を採用します。グループの状態は、それに属するエージェントの中で最も悪い状態を採用します。そして、ポリシーの状態も同様に、割り当てられたエージェントの最も悪い状態を採用します。

これにより、たとえば、障害状態の一つのグループを見ることによって、そこに属する少なくとも一つのエージェントがそれと同じ状態であることがわかります。それを特定したら、上位レベルへ障害状態を伝播させる原因となったモジュールへと、細かいレベルへの情報へ掘り下げていくことができます。

状態の優先順位

状態のうち **最も重要な状態** が伝播されるため、どの状態が他の状態よりも優先されるかが明確である必要があります。

1. アラート発生
2. 障害状態
3. 警告状態
4. 不明状態

5. 正常状態

アラートが発生した場合、その状態は他よりも優先順位が高く、エージェントもこの状態となり、またこのエージェントが属するグループもこの状態となります。

言い換えると、例えば一つのグループが正常状態であれば、そのグループの全エージェントは正常状態であり、全てのモジュールが正常状態であるといえます。

カラーコード

- アラート発生
- 障害状態
- 警告状態
- 不明状態
- 正常状態

Pandora FMS グラフ

グラフは、Pandora FMS の実装において最も複雑なもののひとつです。なぜならDB からリアルタイムで情報を収集していて(rrdtool などの)外部のツールは利用していないためです。

グラフにはいくつかの動きがあり、それはデータのタイプに依存します。

- 非同期モジュール. データ圧縮が無いことを想定しますDB に保存されているデータは、すべて実際に収集されたデータです(圧縮がないため)。抜けなく正確なグラフが生成されます。
- 文字列モジュール. 収集したデータのレートを表示します。
- 数値モジュール. ほとんどのモジュールはそのデータを表示します。
- Boolean モジュール. ping のチェックやインタフェースの状態などの *PROC モジュールでは、数値データになります。0 は、障害を、1 は正常を意味します。

[Pandora FMS ドキュメント一覧に戻る](#)