# Pandora FMS の技術情報

# Pandora FMS の技術情報

Pandora FMS ドキュメント一覧に戻る

## Pandora FMS の技術詳細

この章では、いくつかの Pandora FMS の特別な機能および設計について説明します。

### Pandora FMS のデータベース設計

0.83 から 1.1 までの Pandora FMS の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。

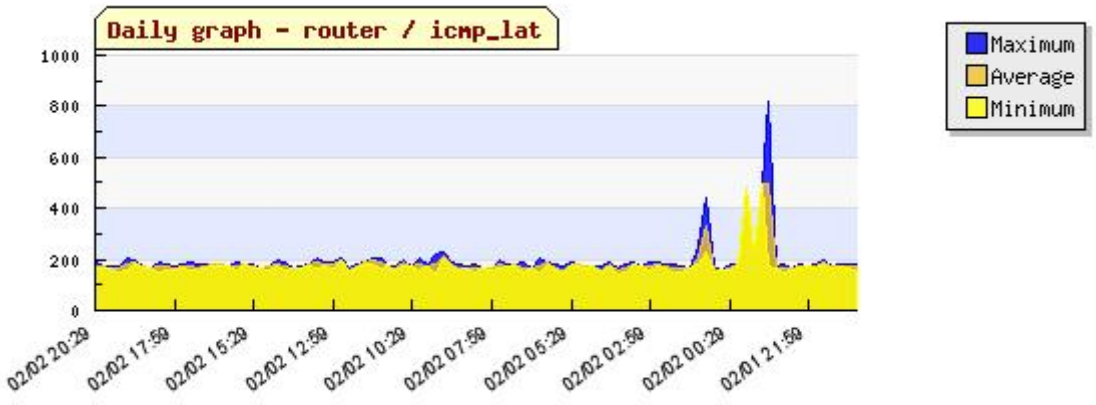
新たなデータ処理システムは、新しいデータのみを保持します。同じ値がシステムに入力された場合、それはデータベースに保存されません。これはデータベースを小さい状態に維持するのにとても便利です。これは、数値、インクリメンタルな数値、ブーリアン等のすべての Pandora FMS モジュールで動作します。ブーリアンデータでは、データの変化はまれであるため、とても高圧縮になります。それでもインデックスデータは、24時間ごとに保存されます。情報の圧縮により、参照するのに便利な最小限のデータとなります。

これは、データ量が 40%～70% となり、スケーラビリティの問題を解決します。また、スケーラビリティの問題を解決するための Pandora FMS コンポーネントを分割するという別の解もあります。これは、データファイル処理の負荷を分散させたり、ネットワークモジュールを異なるサーバで実行できるというものです。現在Pandora FMS ウェブコンソールのように、複数の Pandora FMS サーバ (ネットワークサーバ、データサーバや、SNMP サーバ) を持つことができ、また、データベースも (MySQL5 で) クラスタを組むことができます。

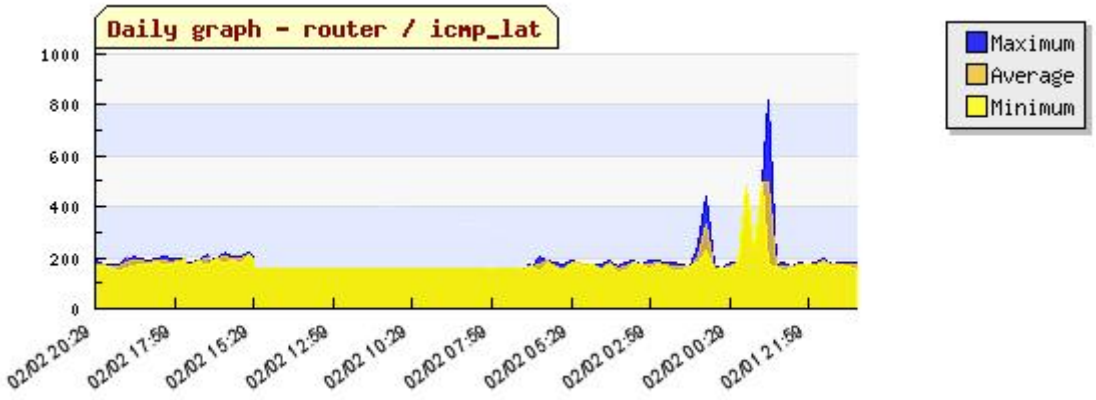
データの読み込みや処理の調整は大きな変化を意味します。我々は、新たなデータ保存モデルにおいて、すばやくデータを表示できるように、ゼロからグラフィックエンジンを設計、実装し直しました。新しいバージョンでは、エージェントが Pandora FMS と通信できずPandora FMS サーバがエージェントからデータを受け取れない場合、この欠落したデータはグラフに表示されません。そして、モジュールグラフについては、変化しません。

グラフは完全に水平線になりますPandora FMS では、新たな値を取得できない場合、それは存在せず、最後に情報が取得できた時の状態にあると判断します。例えばMRTG の処理に似ています。

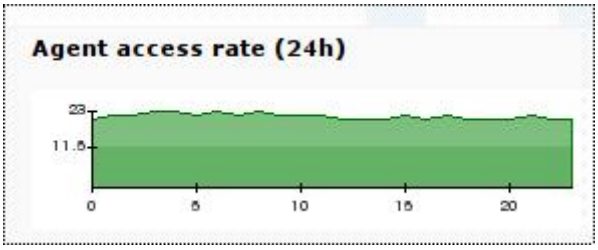
グラフのサンプルとして、以下の画像では 180 秒ごとに受け取ったデータを表示しています。




以下は、05:55頃～15:29頃の間に同じデータが続いているグラフです。接続が失敗している可能性があります。

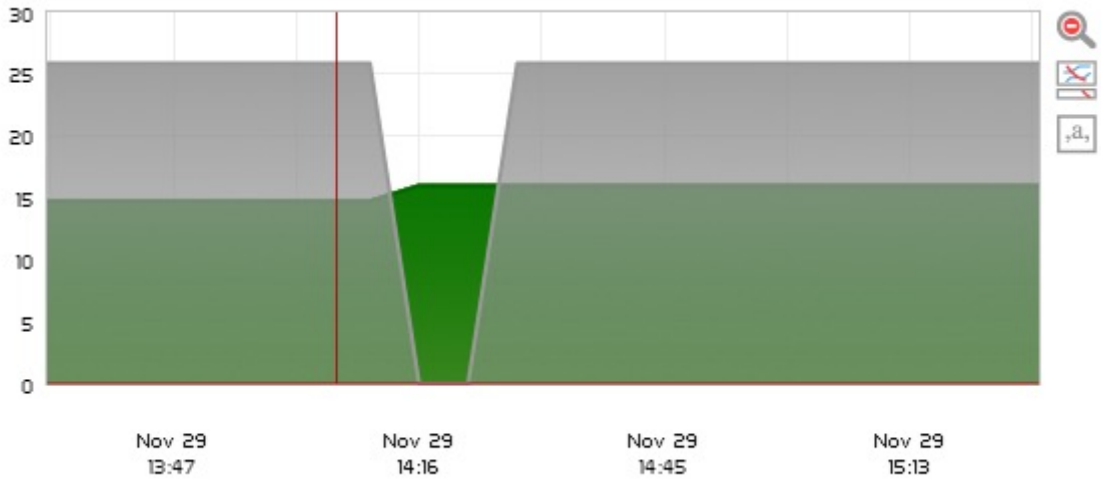



Pandora FMS 1.3 では、新たにエージェント全般のグラフが導入されています。これは、接続性を表し、エージェントモジュールからのアクセス比を示しています。このグラフは、エージェントが稼働していてデータを受け取っている時に表示される他のグラフを補完するものです。以下は、通常サーバに接続しているエージェントの例です。

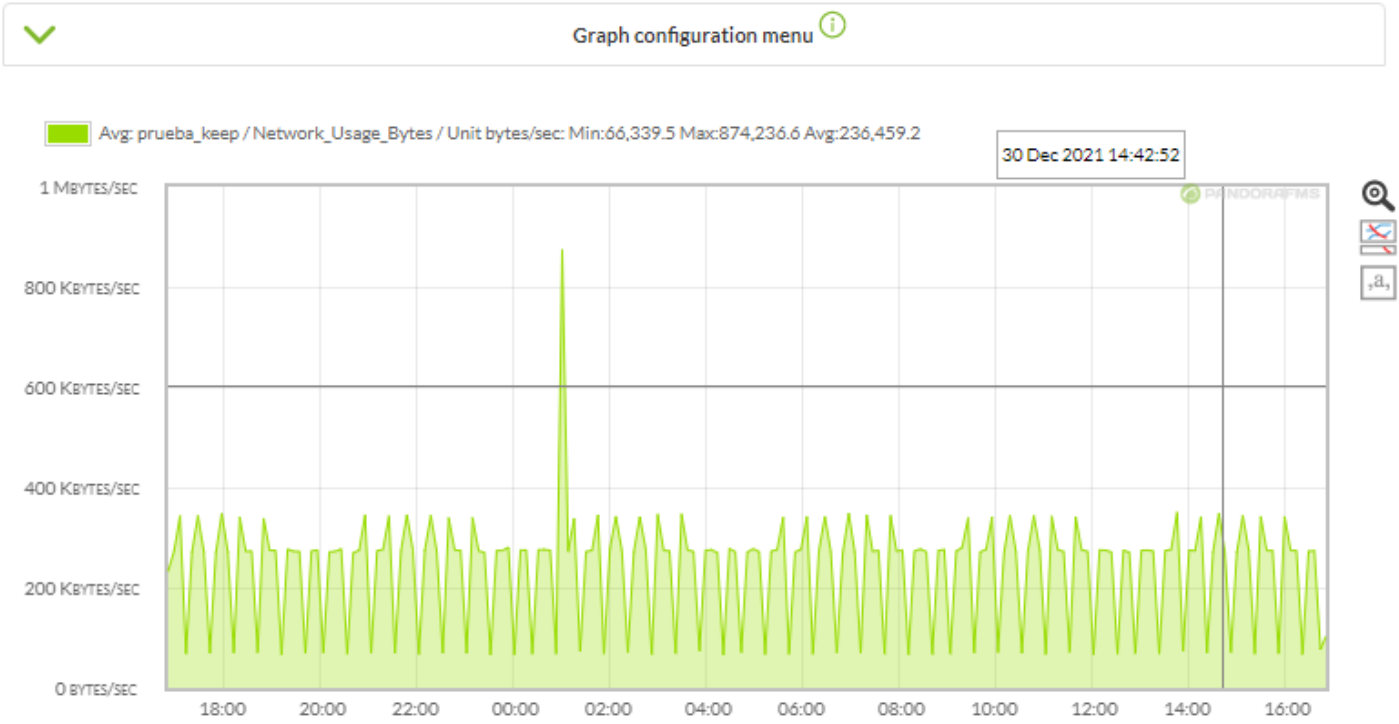



もし、グラフにピークがある場合は、Pandora FMS エージェントと Pandora FMS サーバの接続において問題があるか、遅延が発生している可能性があります。ネットワークサーバからの接続においても問題が発生している可能性があります。


Pandora のバージョン 5以降では、“ 不明モジュール ” タイプのイベントのデータをグラフに関連づけることができ、不明状態のデータをグラフに表示し、グラフを補完することができる新しい機能が導入されました。わかりやすいように以下に例を示します。

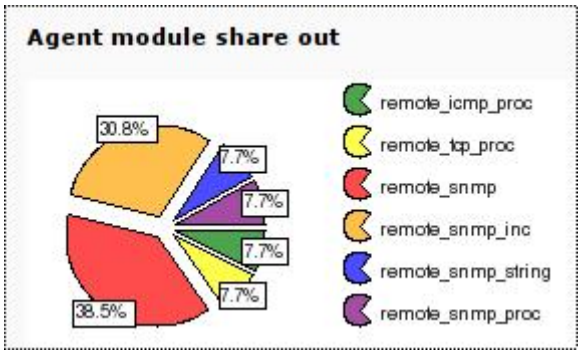



バージョン 7 NG 759には、他のオプションに加えて、イベントやアラートが発生したときに、パーセンタイル、データをリアルタイムで追加できるグラフ設定メニューがあります。




**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**

Pandora FMS データベースのリレーショナルモデルにおいて、小さな拡張を実装しました。導入した変更の一つは、モジュールの種類によるインデックス設定です。これにより、データへのアクセスが早くなりました。それによりPandora FMSエージェントがモニタリングしている全データをアップロードしやすくなりました。異なるソースのデータを細かく分割して提供されますPandora FMSの新バージョンでは、多くの種類の情報を処理できる 4つの種類の新しいサーバを計画しました。



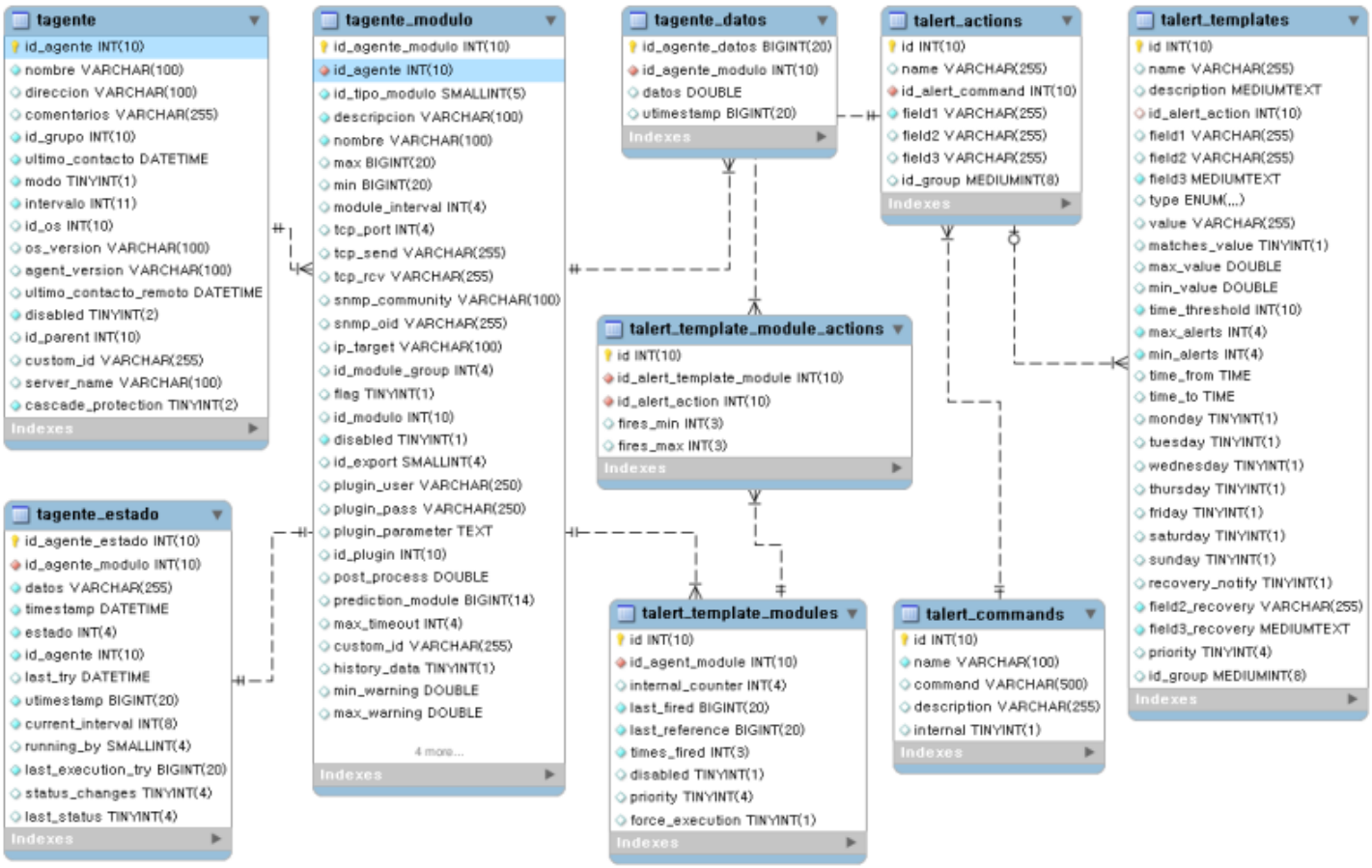

最近のバージョンでは、過去データへのアクセスパフォーマンスをさらに向上させるために、テーブルのパーティショニング(タイム・スタンプによる)を推奨します。

我々はまた、時刻表記の数値表現 (UNIX フォーマットや、1970年1月からの経過秒数) のような要素を追加しました。日付の検索や比較などが高速化しています。これは、データ挿入における検索時間の拡張と考えることもできます。

### データベースのメインテーブル

(2020年4月18日時点の) Pandora FMS データベース構造のより 詳細については、こちらのリンク から確認できます。

以下に、ER図および Pandora FMS データベースの主なテーブルの詳細説明を示します。残りのテーブルについても簡単に説明しています。




- taddress: エージェントの追加アドレス。
- taddress_agent: エージェントのアドレス(taddress/tagente に関連)
- tagente: Pandora FMS エージェントの情報。
  - *id_agente*: エージェントのユニーク ID
  - *nombre*: エージェント名。(大文字・小文字を区別します)
  - *direccion*: エージェントアドレス(traddress を通して、追加アドレスを割り当てることができます。
  - *comments*: フリーテキスト。
  - *id_grupo*: エージェントが所属するグループ ID(tgrupo に関連)
  - *ultimo_contacto*: ソフトウエアエージェントもしくは、リモートモジュールでの、エージェント接続の最終日時。
  - *modo*:エージェントの動作モード。0 は通常モード、1 は学習モードです。
  - *intervalo*:エージェント実行間隔。この間隔に応じて、エージェントはタイムアウトを表示します。
  - *id_os*: エージェントの OS ID(tconfig_os に関連)
  - *os_version*: OS バージョン。(フリーテキスト)
  - *agent_version*: エージェントバージョン。(フリーテキスト) ソフトウエアエージェントによって更新されます。

    - *`ultimo_contacto_remoto*: エージェントが接続してきた最終日時。ソフトウエアエージェントの場合、日時はエージェントから送信されます。
    - *disabled*: エージェントの状態。無効状態であれば 0、有効であれば 1 です。
    - *remote:* エージェントのリモート設定の状態です。有効であれば 1 です。
    - *id_parent*: 親エージェントの ID（tagente に関連）
    - *custom_id*: エージェントのカスタマイズ ID（他のツールとの連携に便利です。
    - *server_name*: エージェントに割り当てられたサーバ名。
    - *cascade_protection*: 関連障害検知抑制。0 で無効です。1 の場合は、親エージェントで障害が発生した場合に、該当エージェントでの障害発生を抑制します。詳細は、 アラートシステムを確認してください。
    - *safe_mode_module*: セーフオペレーションモードの(同一エージェントの)モジュール ID（(モジュールが障害状態の場合、他のすべてのモジュールは無効化されます)
- tagente_datos: モジュールから受信したデータ。同一のモジュールが最後に受信したデータと同じ値の場合は記録されません。(ただし、tagente_estado は更新されます。) インクリメンタルデータおよび、文字列タイプのデータは、別のテーブルに保持されます。
- tagente_datos_inc: インクリメンタルデータタイプ。
- tagente_datos_string: 文字列データ。
- tagente_estado: それぞれのモジュールの現在の状態。
    - *id_agente_estado*: ID（
    - *id_agente_modulo*: モジュール ID（(tagente_modulo に関連)
    - *datos*: 最新の受信データの値。
    - *timestamp*: (エージェントからの)最新のデータを受け取った日時。
    - *estado*: モジュールの状態で、0 正常、1 障害、2 警告、3 不明 です。
    - *id_agent*: モジュールに関連づけられたエージェント ID（(tagente に関連)
    - *last_try*: 最後に実行に成功したモジュールの日時。
    - *utimestamp*:UNIX フォーマットでの最後に実行したモジュールの日時。
    - *current_interval*:モジュールの実行間隔を秒で示します。
    - *running_by*: モジュールを実行するサーバ名。
    - *last_execution_try*: 実行が失敗している場合でも、最後にモジュールの実行を試みた日時を示します。
    - *status_changes*: 状態の変化が発生した回数。継続的な状態変化を避けるために利用されます。
    - *last_status*: モジュールの一つ前の状態。
- tagente_modulo: モジュールの設定。
    - *id_agente_modulo*: モジュールの ID（
    - *id_agente*: モジュールに関連付けられたエージェント ID（(tagente に関連)
    - *id_tipo_modulo*: モジュールの種類（(ttipo_modulo に関連)
    - *descripcion*: フリーテキスト。
    - *nombre*: モジュール名
    - *max*: モジュールの最大値。これより大きな値は、不正な値として認識されます。
    - *min*: モジュールの最小値。これより小さな値は、不正な値として認識されます。
    - *module_interval*: 秒単位のモジュール実行間隔。
    - *tcp_port*: ネットワークモジュールおよびプラグインの接続先 TCP ポート（WMI モジュールでは、読み込むカラム名。
    - *tcp_send*:ネットワークモジュールでの送信データ（WMI モジュールでは、ネームスペース。
    - *tcp_rcv*: ネットワークモジュールでの想定する受信データ。
    - *snmp_community*:ネットワークモジュールでの SNMP コミュニティ（WMI モジュールでのフィルタ。
    - *snmp_oid*: ネットワークモジュールでの OID（WMI モジュールでの WQL クエリ。
    - *ip_target*: ネットワーク、プラグイン、および WMI モジュールでの対象アドレス。
    - *id_module_group*:モジュールが所属するグループの ID（(tmodule_group に関連)
    - *flag*: 強制実行フラグ。1 に設定されている場合、実行間隔に関係なく実行されます。
    - *id_modulo*: id_tipo_módulo では見分けることができないモジュール ID（6 が WMI モジュール、7 が、WEB モジュール。

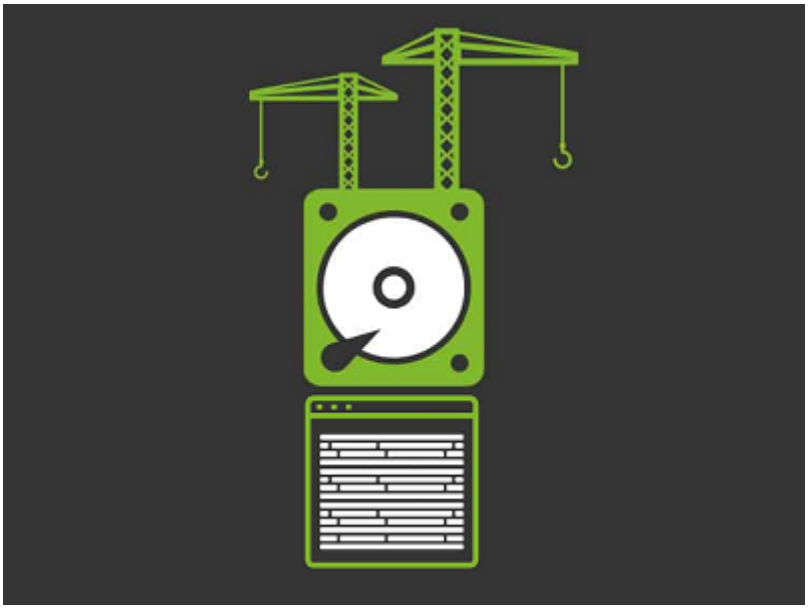
    - *disabled*: モジュールの状態。0 が有効、1 が無効。
    - *id_export*: モジュールに関連づけられた、エクスポートサーバの ID (tserver に関連)
    - *plugin_user*: プラグインおよび WMI モジュールのユーザ名。Web モジュールのユーザエージェント。
    - *plugin_pass*: プラグインおよび WMI モジュールのパスワード Web モジュールの試行回指数。
    - *plugin_parameter*: プラグインモジュールの追加パラメータ Web モジュールの Goliat タスク設定。
    - *id_plugin*: プラグインモジュールのモジュールに関連付けられた、プラグインの ID (tpluginに関連)
    - *post_process*: モジュールデータの値に対して、保存前にかける倍率。
    - *prediction_module*: 1 であれば予測モジュール。0 であればそれ以外。
    - *max_timeout*: プラグインモジュールでの、秒単位の最大タイムアウト値。
    - *custom_id*: モジュールのカスタム ID 他のツールとの連携に便利です。
    - *history_data*: 0 の場合、モジュールのデータは tagente_datos に保存されません tagente_estado のみ更新されます。
    - *min_warning*: 警告状態になる最小値。
    - *max_warning*: 警告状態になる最大値。
    - *min_critical*: 障害状態になる最小値。
    - *max_critical*: 障害状態になる最大値。
    - *min_ff_event*: 何回まで状態変化を許容するかのデータを保持します。これは tagente_estado.status_changes に関連しています。
    - *delete_pending*: 1 に設定されると、データベースのメンテナンススクリプト pandora_db.pl のメンテナンスによって削除されます。
    - *custom_integer_1*: prediction_modue = 1 の場合、このフィールドは予測に使うモジュール ID prediction_module = 2 の場合は、このフィールドはモジュールに割り当てられたサーバ ID
    - *custom_integer_2*:
    - *custom_string_1*:
    - *custom_string_2*:
    - *custom_string_3*:
- tagent_access: いずれかのサーバにエージェントからデータを受信するたびに新たな時間が追加されます。ただし、データベースの高負荷を避けるために、1分より細かい時間は記録されません pandora_server.conf ファイルで、agentaccess を 0 にすると無効になります。
- talert_snmp: SNMP アラートの設定です。
- talert_commands: エージェントに関連づけられたアクションから実行されるコマンド (send mail など) * talert_actions: アラートに関連づけられたコマンド (administrator へのメール送信など)

- talert_templates: アラートテンプレート。
    - *id*: テンプレート ID
    - *name*: テンプレート名。
    - *description*: 説明。
    - *id_alert_action*: テンプレートに関連づけられたデフォルトアクション ID
    - *field1*: カスタマイズフィールド 1 (フリーテキスト)
    - *field2*: カスタマイズフィールド 2 (フリーテキスト)
    - *field3*: カスタマイズフィールド 3 (フリーテキスト)
    - *type*: アラートの状態別の種類 ('regex', 'max_min', 'max', 'min', 'equal', 'not_equal', 'warning', 'critical')
    - *value*: regex の場合のアラートの値。(フリーテキスト)
    - *matches_value*: 1 にすると、状態を反転させます。
    - *max_value*: max_min および最大アラートの最大値。
    - *min_value*: max_min および最小アラー他の最小値。
    - *time_threshold*: アラーとの間隔。
    - *max_alerts*: 一間隔中における、アラート実行最大回数。
    - *min_alerts*: アラートが実行される一間隔中における、状態表示最小回数。

    - *time_from*: アラートが有効になる開始時間。
    - *time_to*: アラートが有効になる終了時間。
    - *monday*: 1 の場合、アラートが月曜に有効になります。
    - *tuesday*: 1 の場合、アラートが火曜に有効になります。
    - *wednesday*: 1 の場合、アラートが水曜に有効になります。
    - *thursday*: 1 の場合、アラートが木曜に有効になります。
    - *friday*: 1 の場合、アラートが金曜に有効になります。
    - *saturday*: 1 の場合、アラートが土曜に有効になります。
    - *sunday*: 1 の場合、アラートが日曜に有効になります。
    - *recovery_notify*: 1 の場合、復旧通知が有効になります。
    - *field2_recovery*: 復旧通知のカスタムフィールド 2。(フリーテキスト)
    - *field3_recovery*: 復旧通知のカスタムフィールド 3。(フリーテキスト)
    - *priority*: アラートの重要度。0 メンテナンス、1 情報、2 正常、3 警告、4 障害
    - *id_group*: テンプレートが属するグループ ID (tgrupo に関連)
- talert_template_modules: モジュールに関連付けられたアラートテンプレートのインスタンス。
    - *id*: アラートのユニーク ID
    - *id_agent_module*: アラートに関連付けられたモジュールの ID(tagente_modulo に関連)
    - *id_alert_template*: アラートに関連付けられたテンプレートの ID(talert_templates に関連)
    - *internal_counter*: アラート通知状態の発生回数。
    - *last_fired*: アラートを通知した最後の時間(Unix time)
    - *last_reference*: 現在の間隔の開始時間(Unix time)
    - *times_fired*: アラートが通知された回数(internal_counter とは異なります)
    - *disabled*: 1 の場合、アラートは無効です。
    - *priority*: アラートの重要度。0 メンテナンス、1 情報、2 正常、3 警告、4 障害
    - *force_execution*: 1 の場合、アラートが発生していなくてもアラートのアクションが実行されます。アラートの手動実行に便利です。
- talert_template_module_actions: 一つのアラートに関連付けられたアクションのインスタンス(talert_template_modules に関連)
- talert_compound: 関連付けアラートです。カラムは、talert_templates に似ています。
- talert_compound_elements: 関連付けアラートに関連付けられた、それぞれのロジックを持った単一アラートです(talert_template_modules に関連)
- talert_compound_actions: 関連付けアラートに関連付けられたアクションです(talert_compound に関連)
- tattachment: 一つのインシデントに関連付けられた添付です。
- tconfig: コンソールの設定。
- tconfig_os: Pandora FMS が扱う OS です。
- tevent: イベントエントリー。重要度の値はアラートと同じです。
- tgrup: Pandora FMS に定義されているグループ。
- tincidence: インシデントエントリー。
- tlanguage: Pandora FMS で扱える言語。
- tlink: コンソールの下の方のメニューに表示されるリンク。
- tnetwork_component: ネットワークコンポーネント。自動検出サーバで使われるネットワークプロファイルに関連付けられたモジュールです(tagente_module にエントリーが保存されると、双方のテーブルは似た状態になります。
- tnetwork_component_group: ネットワークコンポーネントを分類するグループ。
- tnetwork_profileネットワークプロファイル。自動検出サーバの検出タスクに関連付けられるネットワークコンポーネントグループです。プロファイルに関連付けられるネットワークコンポーネントは、作成されたエージェントにモジュールを定義します。
- tnetwork_profile_component: ネットワークプロファイルに関連付けられているネットワークコンポーネント(tnetwork_component/tnetwork_profile に関連)
- tnote: インシデントに関連づけられたメモ。
- tsource: インシデントの発生元リスト。
- tprofile: コンソールで定義されたユーザプロファイル。

- trecon_task: 自動検出サーバの自動検出タスク。
- tserver: 登録されているサーバ。
- tsesion: ユーザの操作記録。
- ttype_modulo: 取得元およびデータの種類に応じた、モジュールの種類。
- ttrap: SNMP コンソールで受信したトラップ。
- tusuario: コンソールで登録されているユーザ。
- tusuario_perfil: ユーザに関連付けられたプロファイル (tusuario/tperfil に関連)
- tnews: コンソールに表示されるニュース。
- tgraph: コンソールで作成されたカスタムグラフ。
- tgraph_source: グラフに関連付けられたモジュール(tgraph/tagente_modulo に関連)
- treport: コンソールで作成されたカスタムレポート。
- treport_content: 一つのレポートに関連付けられた要素。
- treport_content_sla_combined: 一つのレポートに関連付けられた SLA のコンポーネント。
- tlayout: コンソールで作成されたカスタムマップ。
- tlayout_data: マップに関連付けられた要素。
- tplugin: プラグインサーバのためのプラグイン定義。
- tmodule: モジュールの種類。(ネットワーク、プラグイン、WMI...).
- tserver_export: エクスポートサーバの宛先設定。
- tserver_export_data: 宛先に関連付けられた、エクスポートするデータ。
- tplanned_downtime: 計画停止。
- tplanned_downtime_agents: 計画停止に関連付けられたエージェント(tplanned_downtime/tagente に関連)

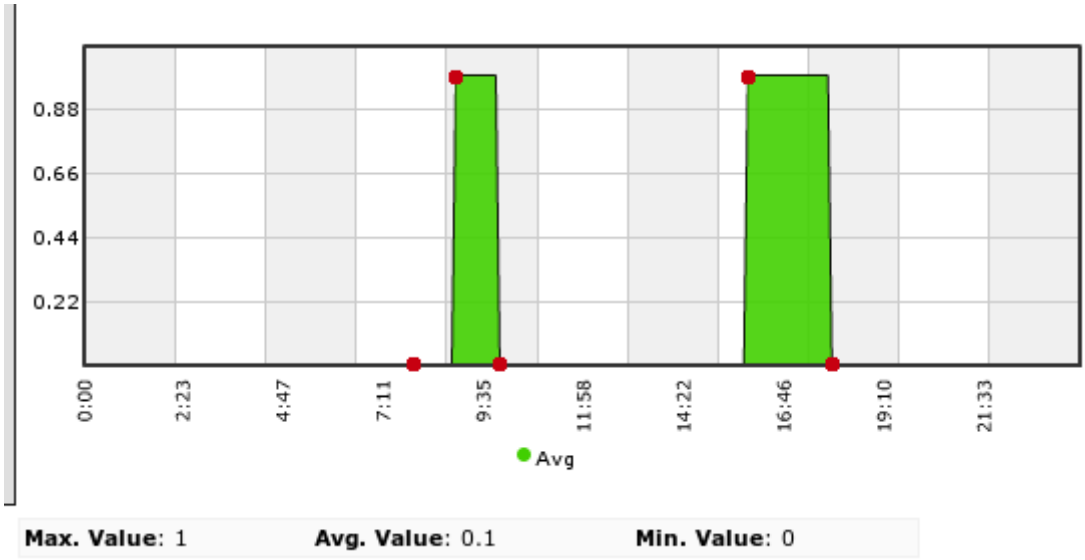
**リアルタイムでのデータ圧縮**



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 0,1,0,0,0,0,0,0,1,1,0,0 というデータがあった場合、データベースには、0,1,0,1,0 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。

以下のグラフでは、上記の例のデータを示しています。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になります。
- 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。


毎日実行される Pandora FMS データベースのメンテナンスで、古いデータを検索し削減します。この削減は、単純な線形補完を用いて行われます。つまり、一日に10,000個の情報があったとすると、

その処理により 1000個に置き換えられます。

これは、明らかに情報が欠落します。なぜならば線形補完だからです。しかし、データベース容量を節約し、長期間(月次、年次)のグラフについては同じように見れます。大きなデータベースでは、これでもデータベースパフォーマンスが必要です。その場合はこれを使わずに、代わりにヒストリデータベースモデルを利用してください。

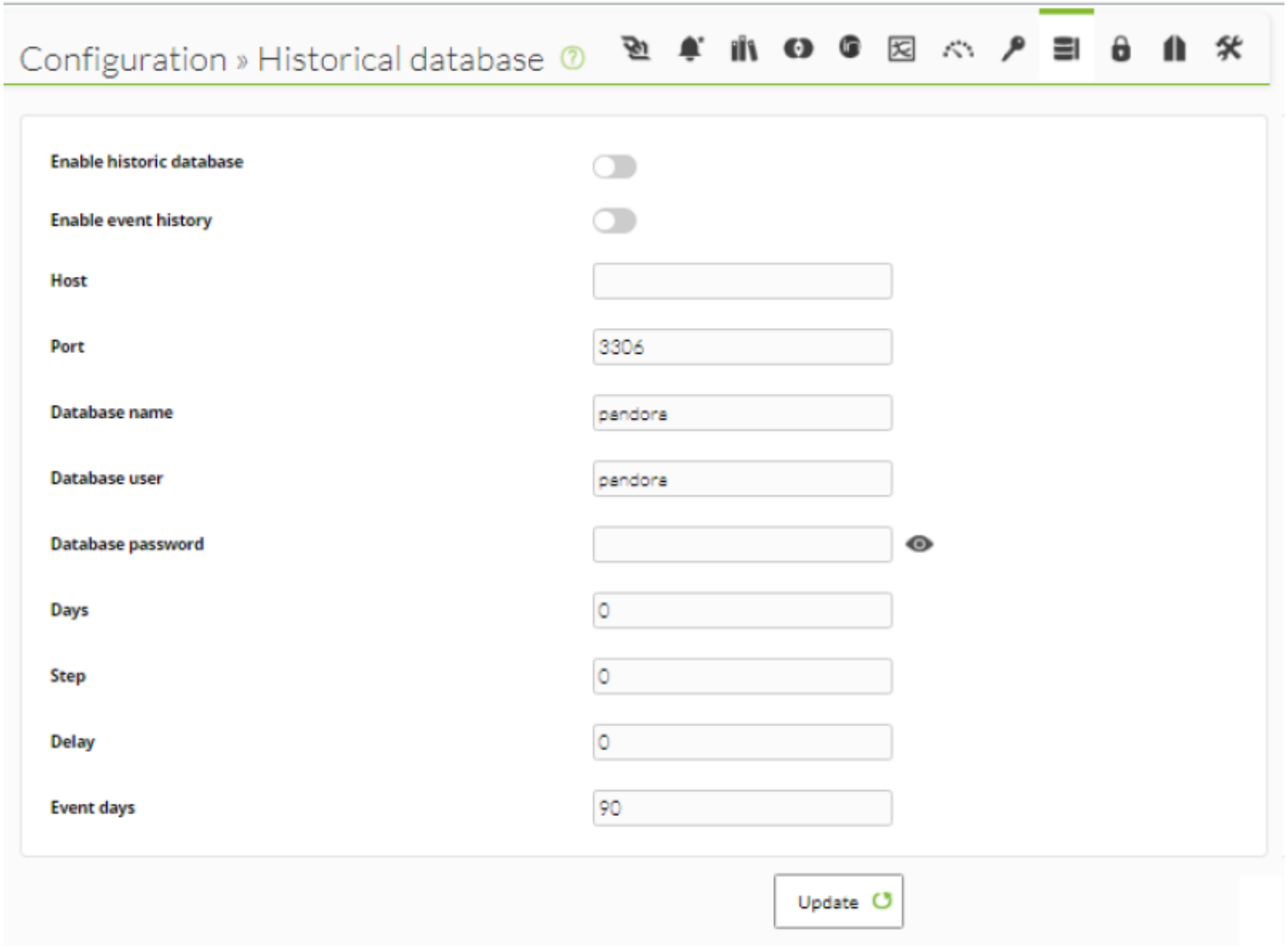
### ヒストリデータベース

これは、日々の参照には使われない古い情報を格納するために使われます。例えば、1ヶ月以上前のデータです。これらのデータは自動的に異なるデータベースにマイグレーションされます。このデータベースは、メインのデータベースとは異なるハードディスク、異なるサーバ上にあるべきです。

古いデータを含むグラフやレポートを表示しようとした時、Pandora FMS は、自動的に直近のデータをメインのデータベースから取得し、それより古い情報をヒストリデータベースから取得します。これにより、大量のデータを保存している場合において、パフォーマンスの問題を避けることができます。

これを設定するには、他のサーバにてヒストリデータベースの設定(Pandora FMS のデータベーススキーマをインポートし、データは入れる必要ありません)とメインの Pandora FMS サーバからのアクセスができるような設定を手動で行う必要があります。

セットアップ(Setup) → セットアップ(Setup) → ヒストリデータベース(Historical database) へ行き、そこでヒストリデータベースへアクセスするための設定を行います。



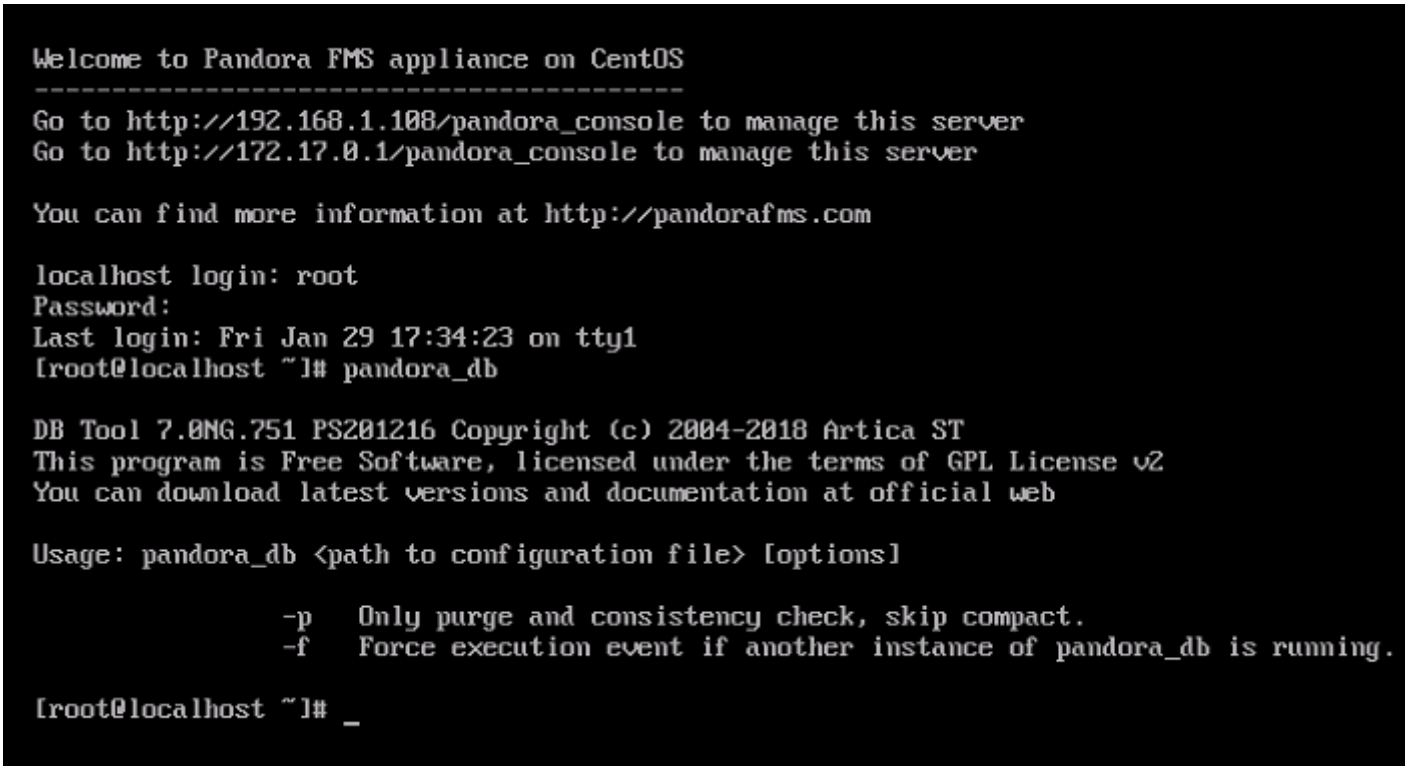

**詳細設定**

Pandora FMS のデフォルト設定では、文字列タイプデータをヒストリデータベースへ転送しません。しかしながら、この設定を変更してヒストリデータベースにこのタイプの情報が送られている場合は、 パフォーマンスに悪影響を与えるだけでなく、大きな問題を引き起こすことになります。

このパラメータを設定するには、データベースで直接クエリを実行して、この情報を削除する日数を決定する必要があります。関係するのは、テーブル *tconfig* とフィールド *string_purge*です。 たとえば、このタイプの情報の保存期間を 30日を設定したい場合、ヒストリデータベースで直接次のクエリを実行します。

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

データベースは、スクリプト pandora_db.pl でメンテナンスされます。




これをテストするには、データベースメンテナンススクリプトを手動実行します。

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

エラーが出ないことが正常です。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、 `-f` オプションを使用して強制的に実行できます。 `-p` パラメータを使用するとデータを圧縮しません。これは、ヒストリデータベースがある HA 環境で特に役立ちます。スクリプトは、これらのコンポーネントに必要な手順が正しい順序とモードで実行されることを確認するためです。

## Pandora FMS におけるモジュールの状態

Pandora FMS では、モジュールは、不明、正常、警告、障害や、アラート発生といった状態を持ちます。

### 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い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で、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。

### ステータスポリシー

正常ではない状態が複数発生している場合、どれが最も重要かを判断する必要があります。そのために優先順位のリストがあり、1番目の状態が他よりも最も優先順位が高く、最後のものが優先順位が低くなっています。すべての要素でいずれかが表示されます。

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ることがわかります。

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

それぞれの状態には、何かが正常に動作していないことをネットワークマップで簡単に表示できるように、色が設定されています。

- 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. 取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。

### 圧縮

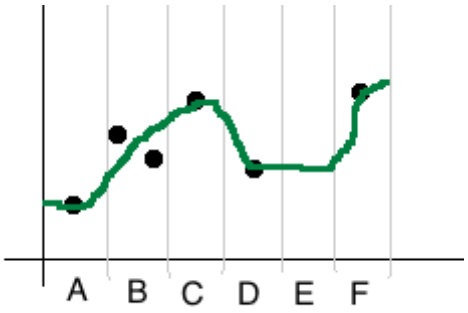
圧縮は、グラフがどう表示されるかに影響します。同じ値の 2つのデータを受信した場合、Pandora は新しい方のデータは保存しません。しかし、他の値が存在しない場合は、最後の値をその時間のデータとして使います。グラフを描画するとき、グラフの開始時点で参照する値がない場合は、Pandora はリファレンスとして参照できる値を 48時間までさかのぼって検索します。何もみつからない場合は、0 で開始します。

非同期モジュールでは、圧縮はされていませんが、同じようにさかのぼって検索する動作を行います。

**補間**

グラフを構成するとき、Pandora は 50xN サンプルを利用します。N はグラフの解像度(この値は設定で定義できます)です。データを 300秒(5分)間隔で収集している場合、1時間に 12のデータがあり、1日では 12×24 = 288 のデータがあります。1日のグラフを表示する場合、288個の値を表示するわけではありません。圧縮や補間をして、50×3 = 150 のデータを利用します(デフォルトでは、Pandora のグラフ解像度は 3 です)。

これは、精度といくつかのデータを失うことを意味します。そして、保持しているサンプルが多ければ多いほど、より多く失うことになります。 しかし、これは、異なる間隔または拡大率でグラフィックを作成することにより回避できます。

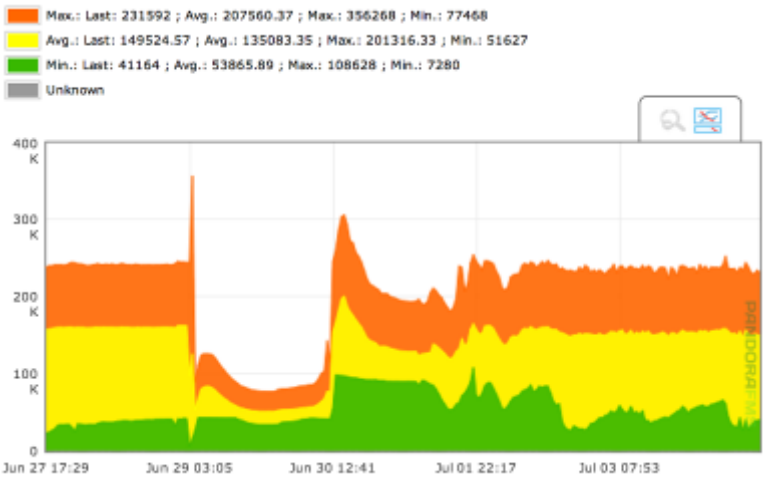



通常のグラフでは、補間は簡単な方法で実装されています。間隔内に 2つのデータがある場合(例のB)は、平均をとってそれを表示します。

boolean グラフでは、複数データ(1または0のみ)がある場合、悲観的なアプローチをとり 0を表示します。これにより、間隔内の障害状態をわかりやすくし、正常状態よりも障害状態の表示を優先します。

両方の場合において、データが存在しない場合(圧縮されているか取得できていない場合)は、その前のタイミングでの最新の収集データを表示に利用します。上記の例の E の ような場合です。

**平均/最大/最小**




**デフォルトでグラフは、**平均、最大**および**最小値を表示**します。サンプル（“**補間**”を参照）は**複数のデータを持つことができるので、データの**最大値や最小値の平均値を示**しています。**長い期間**の多くの**データを**表示**する場合は**より多くの**補間が必要です。しかし、補間レベルが高くなると、**最大と最小の値の差は大きくなります。短い範囲(1時間など)のグラフでは、補間は無いか最小限になります。そのため、**実際の解像度で見ることができます。そして、3つの値が同じになります。