



NOBORI

クラウド型医療用画像システムにおける ZABBIXを利用した分散監視の事例紹介



株式会社NOBORI
2019年11月

NOBORI Ltd.

TechMatrix Group

自己紹介



山根 健志 (Takeshi Yamane)

ハンドルネーム：ふりっばあ

<https://twitter.com/fripper1214>

<https://www.facebook.com/takeshi.yamane.9>

- 医療用画像システムの開発・保守・運用に従事するエンジニア
サーバサイドのモジュール開発から、ハードウェア・ネットワーク・OS周りの
トラブル解消に至るまで、幅広い業務に携わっています
- 業務では1.6.xの頃からZABBIXを愛用中
- 以前はRAIDを伴うような自宅サーバを構築・運用したりもしていましたが
ここ数年は、市販のPCによるコンパクトな構成で運用中

株式会社NOBORI

従来はフィルム等で管理されていた画像情報の保存・管理・利用をシステム化した
医療用画像管理システム(PACS)を、自社開発して提供しています

- 1998年 テクマトリックス株式会社の医療関連事業部門として
オンプレミス型PACSシステムの提供を開始
- 2012年 クラウド型PACSサービス「NOBORI」の提供を開始
- 2018年 医療システム事業部門が、**株式会社NOBORI**として独立

みんなの明日へ、医療情報クラウド。

■ 会社概要 ■

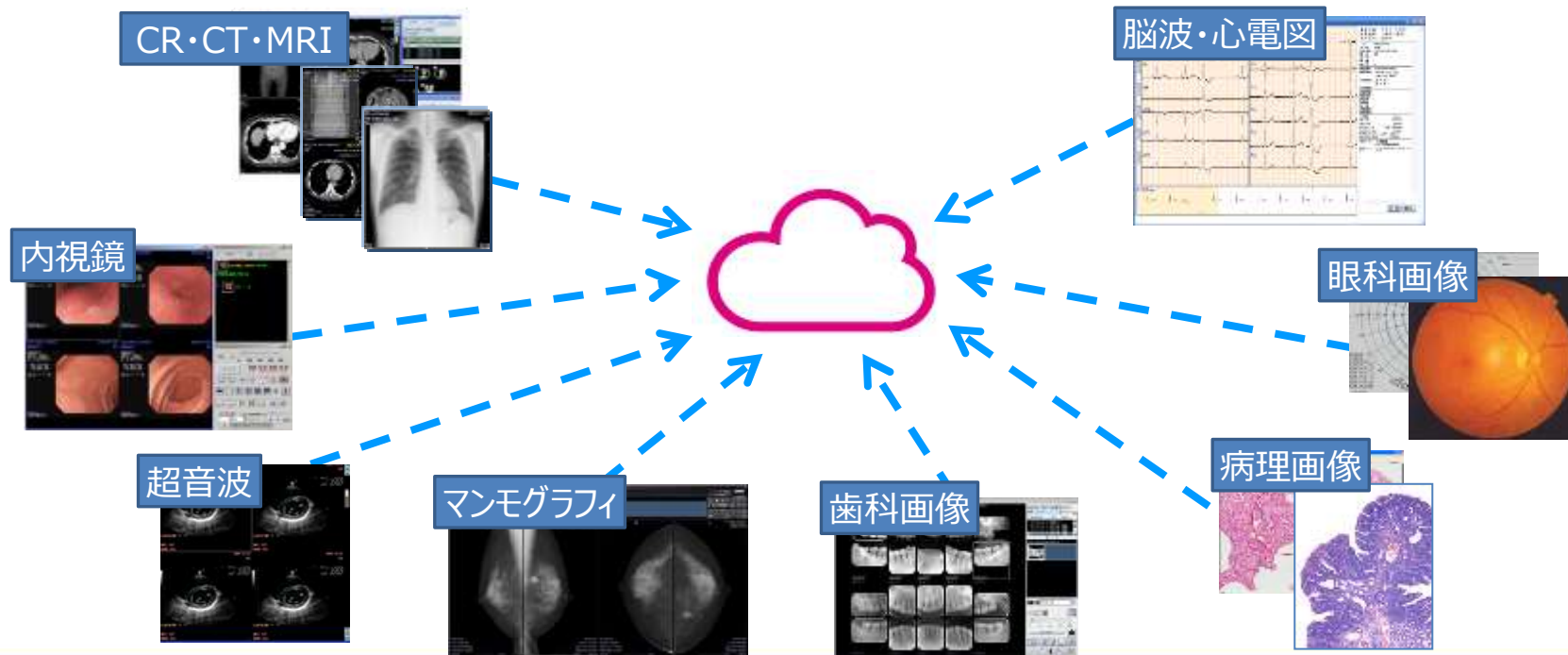
【商号】	株式会社NOBORI	
【設立】	2018年1月22日	
【所在地】	108-0073 東京都港区三田3-11-24 国際興業三田第2ビル	
【代表取締役社長】	依田 佳久	
【資本金】	11億円	
【従業員数】	120名	
【大株主及び持ち株比率】	テクマトリックス株式会社	66.66%
	三井物産株式会社	33.34%
【事業内容】	医療関連クラウドサービスの開発・提供	



NOBORI
Medical information platform

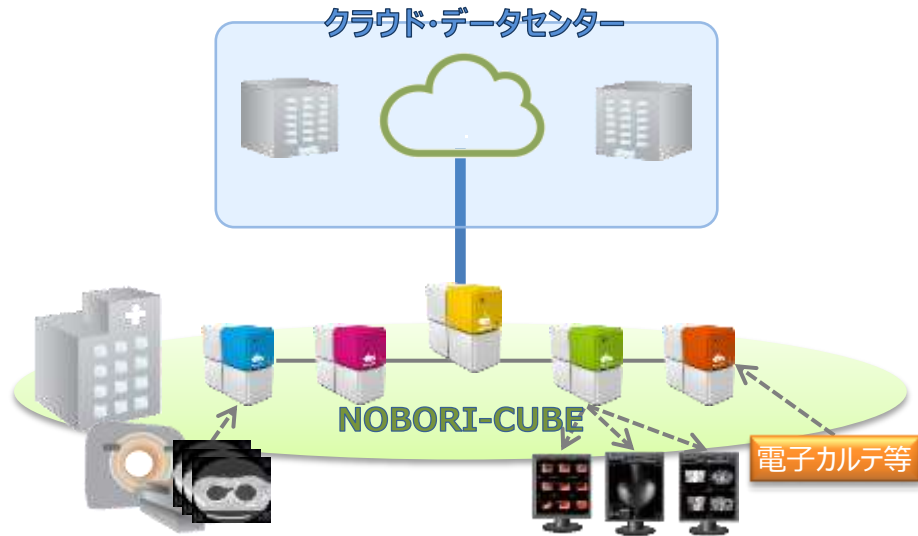
クラウド型PACSサービス「NOBORI」

医療機関内で発生する様々な画像情報を、クラウドサーバで管理



NOBORI システム構成

- ◆ 各施設には、画像データの管理・保存と参照に必要なアプリを搭載した専用の小型アプライアンス(**NOBORI-CUBE**)を設置
- ◆ データは、インターネット経由でクラウド側データセンターへ保管



NOBORI-CUBE

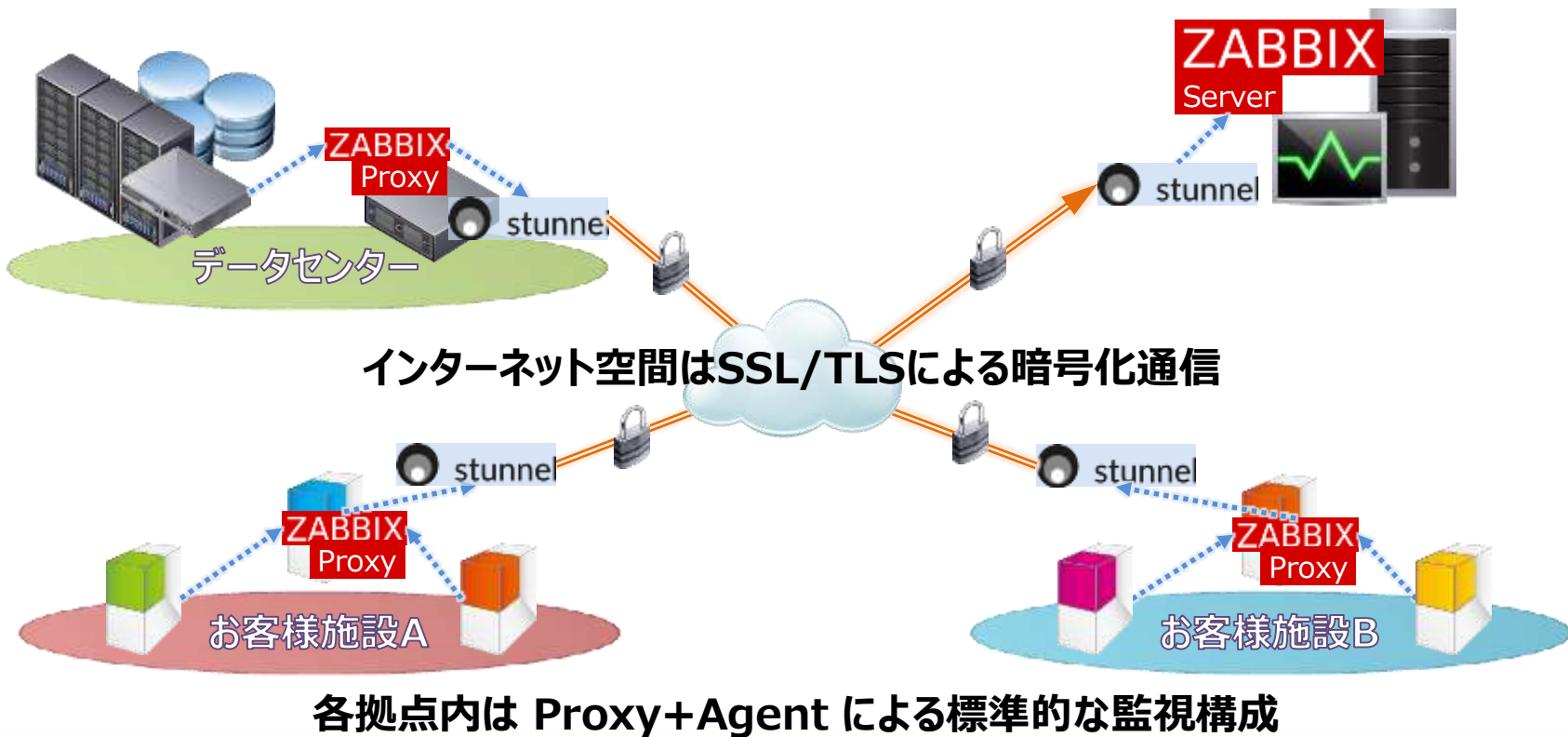
- ◆ Linux系OSを搭載した小型アプライアンス
- ◆ 導入施設の規模に応じ、1台～複数台構成など、柔軟に導入・構成可能
- ◆ クラウド側で集中管理された施設毎の設定情報を受信するモジュールを搭載しており定期的に通信を実施
- ◆ 画像サーバ、データベース、参照用モジュール等、アプリケーションは全てコンテナ化し施設毎の要件・構成に応じて配信
 - ✓ 故障・障害の発生時には CUBE 交換で迅速に復旧
 - ✓ 複数台構成の場合には、リモート操作で隣接CUBEの空きリソースへコンテナを移動すれば、サービス・業務が継続可能



個々のシステム障害が検知できないとダメ

➡ **ZABBIX** で集中監視を実現!!!

システム構成・分散拠点への対応



システム構成・分散拠点への対応

- ◆ 拠点間でやり取りされるZABBIX通信データ
 - 3.0以降はSSL/TLSによる暗号化通信に対応しているが2.x以前では非対応だったため、以下の対策を実施
 - ✓ **stunnel**を利用してSSL/TLS暗号化し、秘匿性・安全性を確保
 - ✓ 各施設へはZABBIX Proxyのコンテナに同梱して配信して稼働

ZABBIXによる監視・運用規模 (2019年11月現在)

- ホスト数 2,700 ホスト
 - うち、テンプレートホスト数 110 テンプレート
 - アイテム数 449,500 項目
 - トリガー数 365,000 項目
 - ユーザアカウント数 1,500 ユーザ
 - **1秒あたりの監視項目数 591.07 件**
- 個々のお客様施設ごとに、ホストグループ・ユーザグループ設定を定義して当該施設の情報に関するアクセス権限を付与することで、障害発生時には担当営業・担当SI・サポートスタッフなど、必要なメンバに通知が飛ぶよう設定

監視システムの計画・導入

- ✓ 監視対象システムの障害検出や、パフォーマンス管理に必要な情報を見極めて、効率の良い監視・収集ができるように計画・導入する
 - アイテムの収集間隔は必要以上に短くしすぎない
 - 余分なデータを収集しても無駄になるだけで、負荷も上昇してしまう
 - ユーザ定義アイテム等において、数値形式で収集できる情報は可能な限り、文字列よりも数値形式のデータとして収集する
 - グラフ表示で値の推移が参照できるなど、活用しやすい
 - トリガーにおける障害判定条件も単純化できるので、負荷の上昇が抑えられる
 - 標準テンプレートは素晴らしいが、実システムでそのまま使うのではなく対象システムにおいて意味のある部分を重点的に監視する
 - 標準テンプレートには監視のヒントが盛り沢山。めいっぱい参考にして活用する

ZABBIX Server/Proxy の負荷対策

- ✓ **内部チェックのアイテム項目(zabbix[xxxxx])を活用**して ZABBIX Server/Proxy のパフォーマンス・チューニングを実施する
 - ZABBIX Server/Proxy 自身は、CPU・メモリ・ストレージ等単なるサーバリソースの監視だけでは不十分
 - 高負荷になっている部分・リソース占有率の高い部分の特定まではできてもボトルネックの原因となっている理由が十分に判断できない
 - 各役割別プロセスの負荷・稼働状況や、キャッシュ・バッファ周りの利用状況、遅延キューの発生状況など、ZABBIX の内部動作を確認できる
 - バッファ量やプロセス数など、各々の設定値についてチューニングを実施して適切・十分なリソースを割り振ることで、無駄な負荷の上昇が抑えられ全般的な監視の動作がかなり改善できる

データベースのパフォーマンス・チューニング

- ✓ DB 関連処理がボトルネックにならないよう、DB周りのリソースやパフォーマンスの推移を監視して、適切にチューニングを実施する
 - DB用の監視テンプレートを活用し、同時接続数やキャッシュヒット率、スロークエリの発生など、DB内部のパフォーマンス情報について監視する
 - 同時接続数やバッファメモリの割当量など、設定値を見直しチューニングする
 - ZABBIX/DB が日々処理し続けなければいけない監視データの目安量は、Web 画面の「1秒あたりの監視項目数」から確認できる
 - 監視ホストやアイテムの追加でデータ流入量は日々増加する
 - パフォーマンス低下の原因が、CPUの演算スペック不足なのか、ストレージのI/Oスペック不足なのか・原因の分析が重要
 - 流入データを常時安定して処理し続けられるだけのサーバスペック・システムリソースを計画的に確認し、事前に確保するよう対処する

データベース周りの負荷対策

- ✓ 日々の動作において、DB 関連処理は流動的に増減する
不定期に発生するピークを見極め、耐えられるような構成にする
 - 流入データを**24時間安定して処理し続けられる構成**はあくまで**最低限**
以下のような事象も十分に考慮して、チューニングや強化を実施しておく
 - ログ監視・SNMPトラップ監視などによる大量データの瞬間的なバースト流入
 - Web画面の操作による監視履歴データの検索や、設定の更新操作による負荷上昇
 - 障害判定後のアラート通知周り動作など、ZABBIXの内部処理による負荷上昇
 - 古くなった履歴・トレンド値を削除するhousekeeperモジュールによる負荷上昇
 - 定期的なDBバックアップ等、一時的なパフォーマンス低下

監視対象となる施設・ホスト情報登録の自動化

- ✓ 個々のお客様施設・分散拠点の増減や、監視対象ホスト(CUBE)の台数・設置状況の変更のような変化は、日常的には発生しない
- ✓ **日常的ではないからこそ自動化・スクリプト化することでミスを防止**
 - 施設情報管理用のサーバにおいて、分散拠点毎の設定情報を集中管理
CUBEの配置・IPアドレス・アプリケーションコンテナの配置状況 等…
 - 監視対象のシステム全体で、いかに同じ方式で設定情報を管理してそれらの設定情報を、常に最新の状態に維持し続けるか、が重要
 - 設定情報を基にして、**ZABBIX-APIで登録・更新**
 - Azure/AWS/GCP 等のようなクラウド環境ならば、API・コンソール等からZABBIX に監視登録する際に必要な情報・設定値を取得することができる
 - ホストグループ・ホスト・テンプレート・ユーザグループ・ユーザ・Proxy の配置など施設毎の監視に必要な情報をAPIで登録・更新

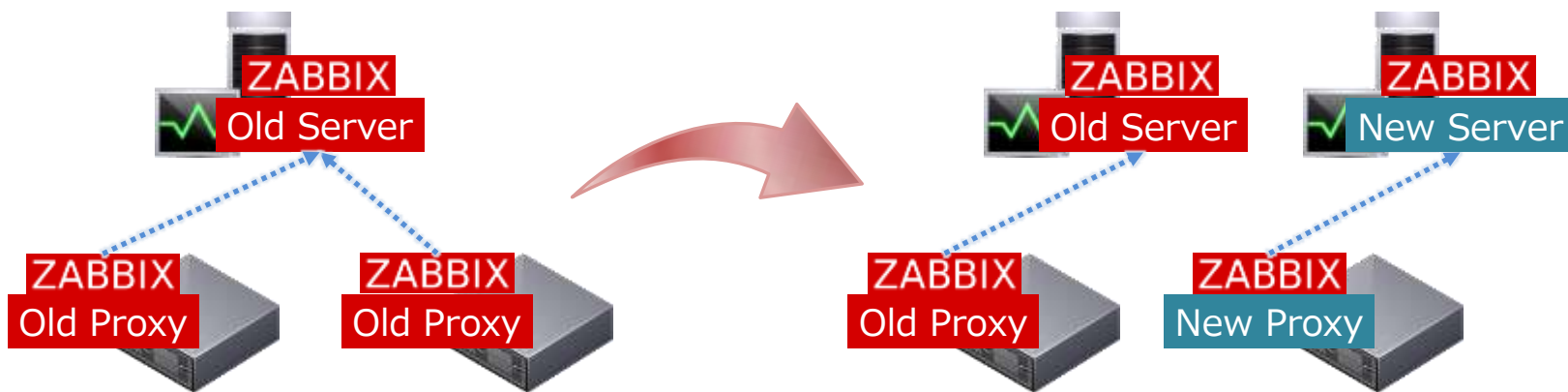
監視対象毎に固有の監視項目登録の自動化

- ✓ 個々のお客様に固有の設定や、コンテナ配置の構成、連携する外部システムの構成など、細かな変化は、日常的に発生する
- ✓ **日常的だからこそ、Low Level Discovery で自動化**
 - ZABBIXの標準機能でLLD対応可能なネットワーク・ストレージ等の構成に限らず、コンテナの配置状況や、ファン回転数・温度等のハードウェア関連、NASやUPS等の周辺機器についても、独自スクリプトでLLDに対応
 - テンプレート側に設定するディスカバリルールと、応答のJSON表現、アイテム収集用の独自スクリプトを工夫することで、**多くの監視項目をLLDによる動的監視に対応可能**

NOBORI 監視における今のテーマ

◆ Proxy を含む ZABBIX システムのバージョンアップ

- 監視対象の設定や次回収集のタイミング等、内部情報を相互で通信しているため、Server/Proxy は**同じメジャーバージョン**で稼働している必要がある
 - 分散拠点・Proxy 多用構成の場合、バージョンアップ・移行は大変



新しい Server を別に用意し、1拠点ずつ順番に移行するのが安全かつ確実

NOBORI 監視における今のテーマ

◆ Proxy を含む ZABBIX システムのバージョンアップ

- Agentd については、無理に更新しなくても概ね動作するが、アイテムキーの表現や取り扱いが異なっていたり、ログ監視周りの動作が異なっていたりするなど、バージョンアップによって改善され続けている
 - 新旧バージョンの Agentd が混在すると、テンプレートの設定で困ることも起こりうる
 - 余力がある限り、新しいバージョンへと更新しておくほうが良い
- ZABBIX モジュール全般の移行・バージョンアップがやりやすくなるように、初期の設計・導入の時点から、**パッケージやコンテナといった構成で構築・運用**しておくのが良い

ZABBIXによる監視を採用・導入してみたて…

- ◆ お客様システムにおける障害発生をいち早く検知することにより、お客様からの障害問い合わせに先立って、**保守スタッフからお客様へ連絡**しての対応が可能となった
 - ✓ 対応の迅速化
 - ✓ お客様満足度の向上
- ◆ オープンソースで無償利用できるうえ、情報も豊富
- ◆ これまで7年間にわたって運用してきた中でメンテナンスを除く**システム障害は発生しておらず非常に安定している**

ZABBIX 監視のメリット!



みんなの明日へ、医療情報クラウド。



NOBORI
Medical information platform

株式会社NOBORI

東京都港区三田3-11-24 国際興業三田第2ビル 5F

TEL : 03-4405-7834

E-mail : sales-office@nobori.ltd