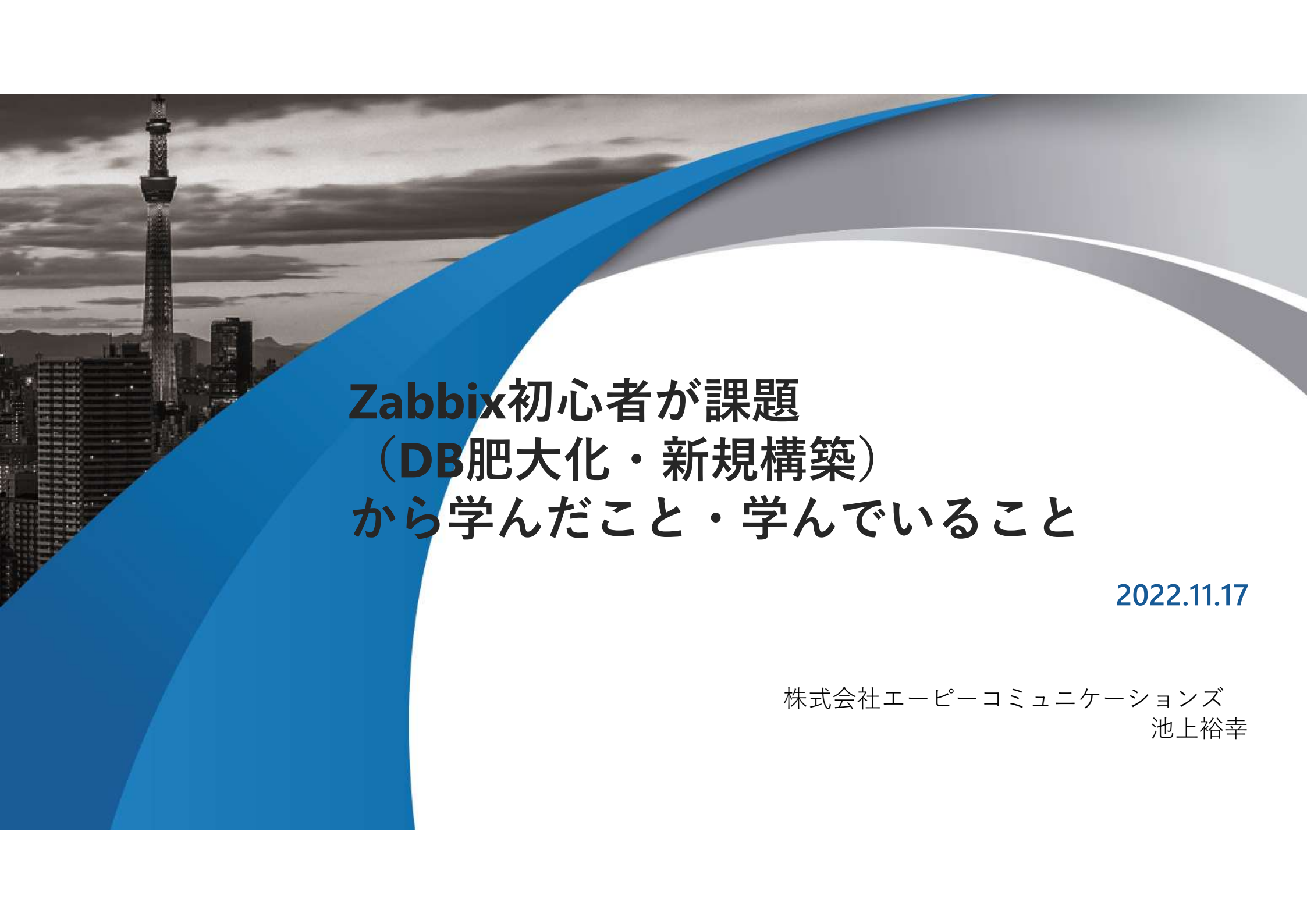




Zabbix初心者が課題 (DB肥大化・新規構築) から学んだこと・学んでいること

2022.11.17

株式会社エーピーコミュニケーションズ
池上裕幸

株式会社エーピーコミュニケーションズ
池上裕幸

サーバーインフラエンジニアとして職務に従事しております。
Zabbix経験は数カ月で知識は初心者レベルとなります。

現在、参入した業務で既存Zabbixの運用保守と
新規構築するZabbixへの移行が当面の業務となっており
その間に学んだことを今回、事例としてお話いたします。

恐らく誰もが経験された事なのでは、というお話で恐縮ですが。



このお話

Zabbix初心者が課題（DB肥大化・新規構築）に取り組んだ&取組中のお話です。

現環境での課題

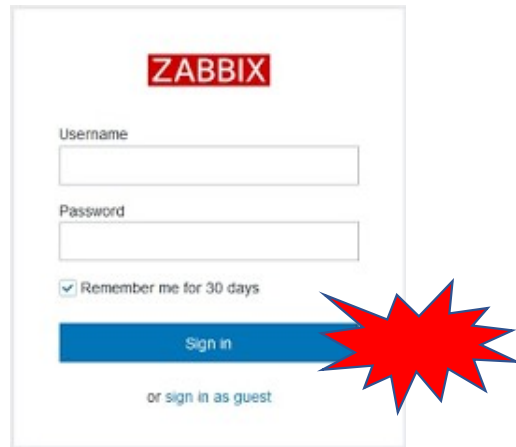
- ・ ZabbixDB肥大化
 - ・ 原因特定
 - ・ 対処（暫定、根本対処策定）

新環境での課題

- ・ 新規構築時の考慮について

事象の発生

Zabbixの動作が遅い、ログインが出来ないとの連絡を受ける。



環境情報)

Zabbixのバージョン2系

OS:CentOS系

DB:MySQL系 (ストレージエンジン : Myisam)

初動調査をすると

ZabbixDBを設置しているディスク領域が100%になっていた。

発生している事象の要因)

DBを設置している領域での書込み不能状態であるため

ログインの際の認証情報などの書き込みが出来ず、アクセスが拒否されたと思われる。

初動対応)

DB以外にログファイルを設置していたため

暫定的に別領域へ退避。空き容量を確保した。

対応効果)

ディスク領域が確保できたら、Zabbixダッシュボードへのログインが可能となった。

復旧したと思って操作すると
監視データの読込が終わらない事象を確認した。

sqlログを確認するとテーブル破損していることがわかる。

対応方針を決めて実行。

1. zabbix-serverプロセスを停止。
2. 破損テーブルのチェック(CHECK TABLE)
3. 破損テーブルの修復 (REPAIR TABLE)
4. zabbix-serverプロセスを起動。

この流れによりZabbixによる監視機能の復旧が出来た。

ZabbixのDBのディスク領域が100%になって起こる事。

※あくまでこの事例での話ですが。

1. Zabbixダッシュボードへのログインが不能になる。
2. 監視データが取得（書き込み）できなくなる。＝「監視不能になる」
3. DBのテーブル破損。（中途半端な書き込みでテーブルが破損）

当たり前ですが

Zabbixの監視機能が不全になるという事が解りました。

そもそも、そのディスク領域をひっ迫させたのは何が原因なのか？

サイズの大きいファイルを探す。

```
# find /{該当領域} -size +500M | xargs ls -lh
```

```
-rw-rw---- 1 mysql mysql 11G X月 8 09:04 2022 /{該当領域}/mysql/zabbix/history.MYD  
-rw-rw---- 1 mysql mysql 8.3G X月 8 09:04 2022 /{該当領域}/mysql/zabbix/history.MYI  
-rw-rw---- 1 mysql mysql XXXG X月 8 09:04 2022 /{該当領域}/mysql/zabbix/history_log.MYD  
～略～
```

> > **history_log.MYD**

ZabbixDBのテーブルhistory_logのデータファイルが100GBを超えるサイズとなっている。

history_logのテーブル巨大にした要因は何？

zabbix自体の該当領域の空き容量グラフをみることにする。



ある期間～事象発生の中で急激に空き容量が減っていることがわかった。

調査方針

期間がわかったので、肥大化原因のhistory_logテーブルを調べる。

何を？

特定の期間でテーブルサイズ肥大化させた監視データ。

どの監視データかは対象のホストやアイテムがわかれば良い。

history_logのテーブル巨大にした要因は何？

DBの調査を開始する。

history_logのカラム itemid を該当期間内でカウントした一覧を表示。

```
select itemid,count(itemid) from history_log  
where clock >= 開始時間 and clock < 終了時間 group by itemid order by count(itemid) desc;
```

※開始時間、終了時間はunix時間に変換して使用。

itemid	count(itemid)
123345	230000000000
154561	1000
～	～

対象アイテムのアイテム名、ホストIDを特定。

```
select itemid,hostid,name from items where iteid = "123345";
```

対象アイテムのホスト名を特定。

```
select itemid,hosted,name from hosts where hosted = "XXXX";
```

history_logのテーブル巨大にした要因は何？

⇒ ログ監視のアイテムによるヒストリログが要因だった。

何故その監視データが肥大化したか？

- ・ネットワークの利用状況の変化により、監視データが増加していた。

簡単な数値でいうと

1日約200万件のヒストリログデータの半分以上をそのアイテムによるデータが占めていた。

特定できたアイテムを調査すると
引き続き膨大な監視データをヒストリログに書き込みは継続しているため
暫定対処で開放した領域をひっ迫させることは予想できた。

根本対応

アイテムの無効化

効果

テーブルサイズの肥大化が緩やかになった。

課題

肥大化したhistory_logテーブルが
いつ他の監視データで急激な増大によるひっ迫を起こすかわからない状況は変わらず。

肥大化したhistory_logテーブルのサイズ縮小化の検討。

保存期限を超えたヒストリログデータ定期削除処理の調整で縮小化できないか試す。

[対応]

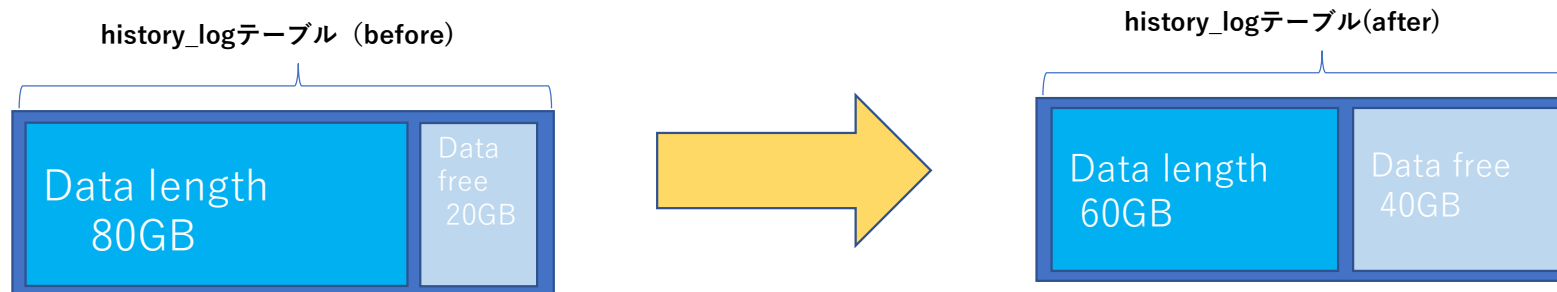
- ・ ヒストリの保存期限の見直し。
- ・ Housekeeping処理(/1h)の削除レコード数を増やす。

/etc/Zabbix/Zabbix_server.conf 編集

MaxHousekeeperDelete=5000 ⇒ 20000

[効果]

- ・ テーブルサイズのダウンサイズは見られなかった。
- ・ テーブル内にある「Data_free」という未利用領域が増加していることを確認。



目的としていたhistory_logテーブルのシュリンクは図れなかった。

状況整理のため、一旦DBの機能を調べることにする。

使用しているDBの仕様として一旦、テーブルが記憶領域として確保した領域を開放することがなくデータ削除などをすると「Data free」という未使用領域が増加することがわかった。

「Data free」の領域を解放するにはテーブルの再構築を行う必要があることに行きつく。

課題

現環境DBのエンジンがMyisamであり

再構築をおこなうためにはオフラインでの実行が必要。

>> 監視を止める必要がある(監視停止はNG・所要時間が不明)

テーブルの再構築を行う場合、テーブルと同サイズの以上の空きが必要。

>> そもそも空き容量に余裕はない

実行手順を用意した上で、テーブル再構築によるディスク領域の開放は断念した。

対応策

DBの領域を別領域へ移設する。

⇒DBをコピーするのに時間を要してそのあいだ監視が停止してしまうため却下となる。

⇒**最悪の事態を想定して、移設手順のみ用意することとした。**

該当領域のDBと関係のないファイルを別領域へ機能移転する。

⇒**ログファイル保存する領域としても利用されており、数十GBの領域開放できた。**

結果

残念ながら根本的なサイズ縮小はできなかった。

平行して運用しながら新規構築での監視機能の移行を進めることとなる。

移行先のZabbixサーバの構築について

バージョンとしては最新安定版6.0LTSとした。

OS: Alma Linux

DB: MariaDB (ストレージエンジン : innoDB)

今回の対応でDBのディスク領域の急激な増大をさせない対策として以下を組み込むこととしています。

Zabbix自体のチューニング

hosekeeperの最大削除数の調整。

ヒストリログの保存期限の管理。

⇒ 根本的なサイズ縮小には貢献しないが

開放するためのData freeのサイズで関節的な貢献が出来る。

DBに関して

以前の環境と違いエンジンはMyisamからinnoDBとなっており

オンラインでテーブル再構築が可能であるため

定期的にData freeの領域を解消させる処理を追加することしました。

innoDBについて

Zabbixで使用しているDBのストレージエンジンがinnoDBでも同様にMyisamでも発生していた断片化(未使用領域の増加)が発生する。

一度、DBの割当領域に書き込まれたデータは断片化により永久に使用領域が増加する。



↑イメージ図で、枠がディスクの割当領域で、●がデータとなる。

データが散らばって整理されていないように書き込まれていると空き領域 (Data free)があったとしても、その分が使用領域として確保されてしまう。

対策として、定期的に断片化が発生しているテーブルの再構築 (デフラグ) にてデータの整理が必要。

↓整理したデータのイメージ



テーブル再構築について

```
ALTER TABLE table_name ENGINE=InnoDB
```

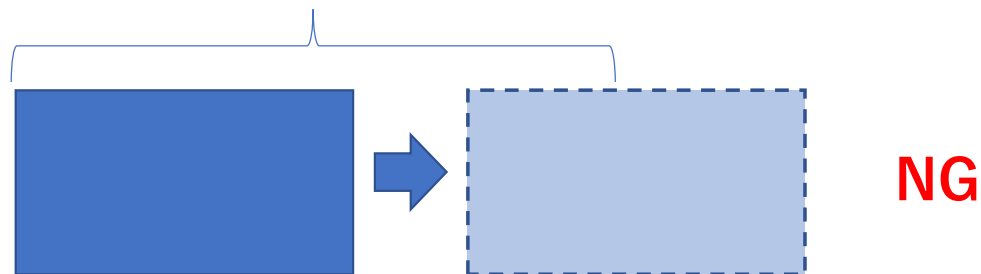
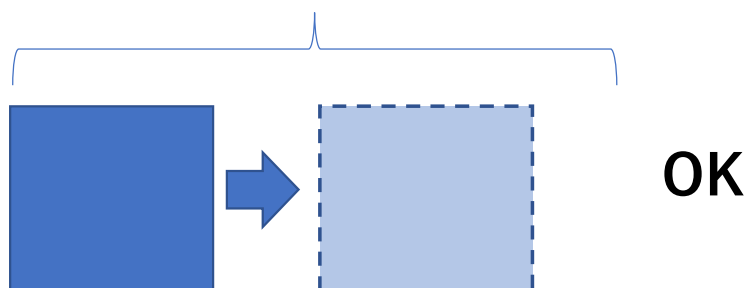
オンラインでの操作が可能、瞬間的に処理前後でロックがかかり注意が必要だが
実機での検証を何度かやっているが、監視機能への影響はほぼなかった。

ディスク領域がひっ迫した際に懸念されることについて

テーブル再構築をおこなうには

そのテーブルサイズと同サイズ以上の**空き容量**が必要であること。

実際に空き容量がギリギリのサイズで検証したらディスク領域が100%になり再構築処理が停止する。



適度な空き領域を確保するように
未使用領域があるテーブルの再構築を行うスクリプトを作成して定期運用を行うこととしました。

ディスク領域の管理や定期的なメンテナンスも
Zabbixサーバの運用上重要であることを実感したお話でした。

わたしのごと

現在、新規構築したZabbixへの移行作業に
手を付けだした所なのですが、新たな課題に直面しています。

課題

新・旧環境差分を補う知識。
Zabbixを本当の意味で監視ツールとして扱う基礎知識及びノウハウ。
(監視設定、条件式etc、API)

解決策

地道な学習...。
今回のようなカンファレンスで有識者の知識に触れる。
そのため、今回Zabbixのカンファレンスに参加させていただきました。

本日は、拙い内容にお時間を頂きありがとうございました。