

Zabbix における TimescaleDB の利用方法と パフォーマンス

Zabbix Conference Japan 2022

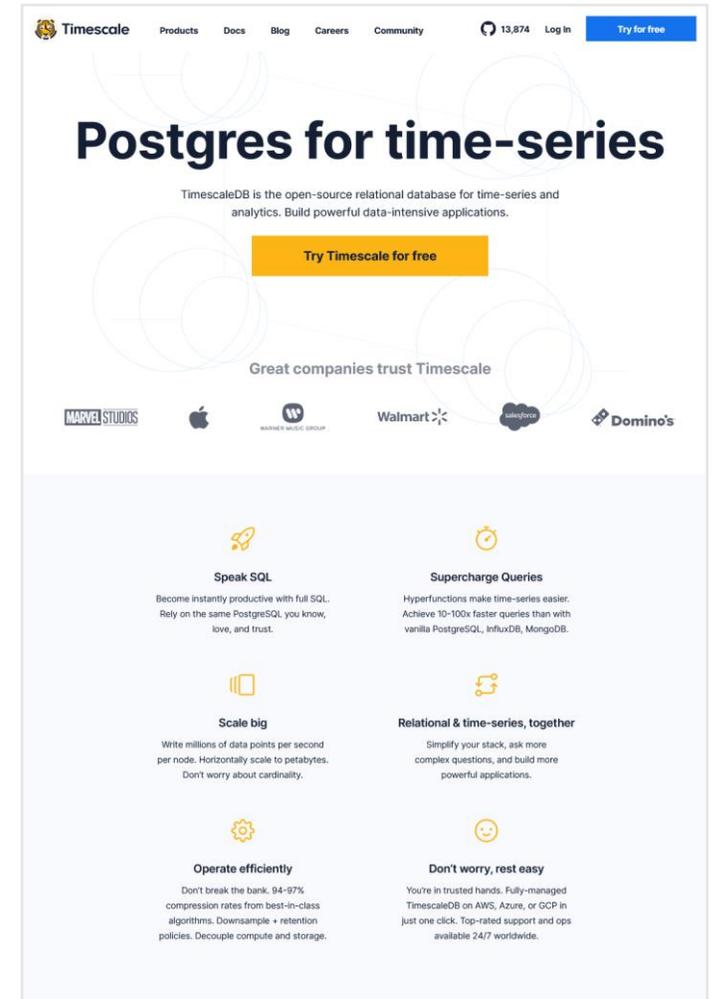
SRA OSS LLC
赤松 俊弘

- 赤松 俊弘 (Toshihiro Akamatsu)
- SRA OSS LLC (SRA OSS, Inc. 日本支社から事業移転)
- Zabbix 認定プロフェッショナル
- PostgreSQL 以外の OSS 全般の技術サポート、構築
- 主に Zabbix を担当

- TimescaleDB の基本
- Zabbix の DB として TimescaleDB を利用する際の特徴やメリット
- TimescaleDB と PostgreSQL のパフォーマンス比較

TimescaleDB とは

- 種別: 時系列データベース
- ウェブサイト: <https://www.timescale.com>
- 開発元: 米国 Timescale 社
- 初版リリース: 2017 年
- ライセンス: Apache-2 (一部TSL)
- 実装言語: C 言語
- コード量: 約 14 万行



自由に使える

**オープンソース
機能**
(Apache
License 2.0)

**コミュニティ
機能**
(Timescale
License)

DBaaS 提供除き
自由に使える

コミュニティ機能

- チャンクの並び替え・移動
- 分散ハイパーテーブル
- 圧縮
- 継続的集計
- データ保持
- ユーザ定義アクション
- 欠損値の補間

※[ライセンス](#)は各自で確認すること



使いやすさ

- SQLに完全対応
- PostgreSQL用アプリケーションがそのまま動作可能
- 時系列データ向け分析・管理機能あり



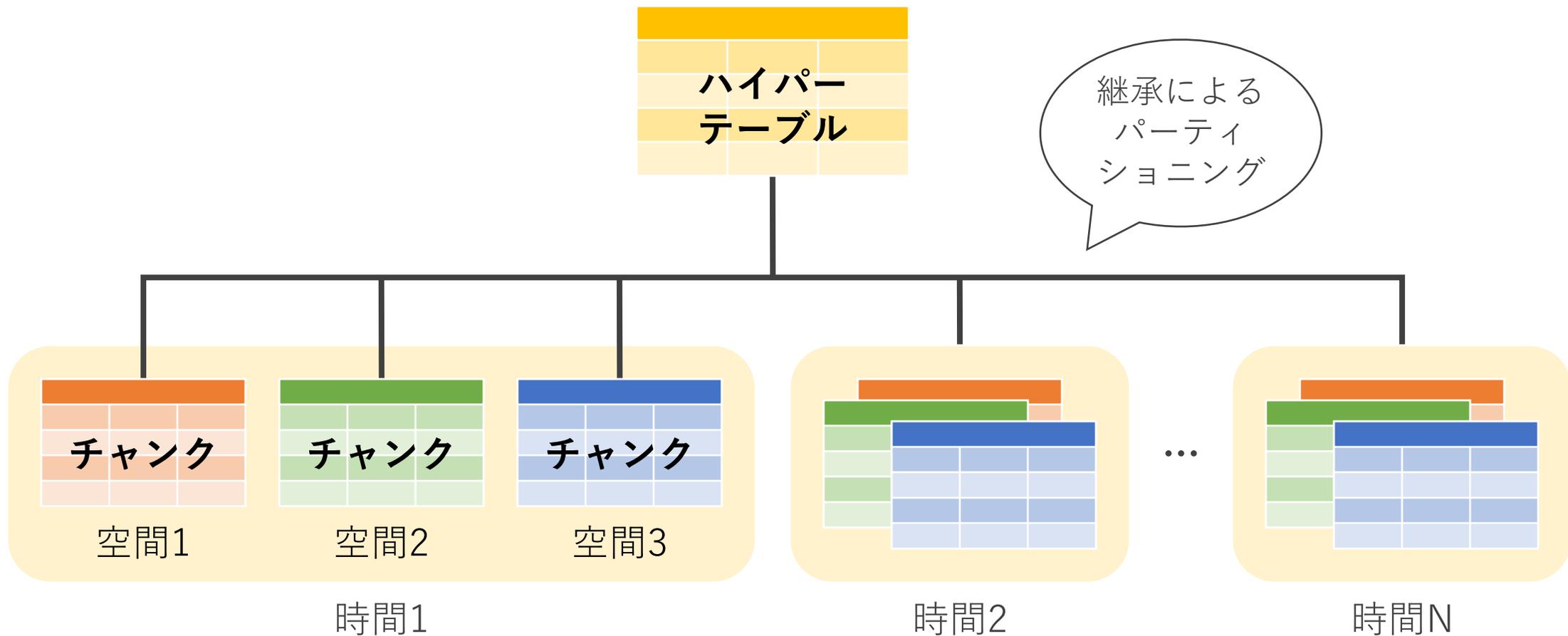
スケーラビリティ

- データを適切なサイズに分割して格納
- データ量が増えても安定して高速に書き込み可能
- 分散処理に対応



信頼性

- 長い歴史をもつPostgreSQL上で動作
- バックアップやレプリケーションなど、PostgreSQLの機能・ツールが利用可能



- Timescale Cloud を使う
 - Timescale 社提供のマネージドサービス
 - AWS と Azure、GCP に対応
 - 0.032米ドル/時間～
- 自前で構築して使う
 - Debian/Ubuntu、RHEL/CentOS、Windows、macOS 向けパッケージ、Docker イメージ、AMI が利用可能
 - ソースコードからもビルド可能

TimescaleDB/PostgreSQL バージョン対応

TimescaleDB バージョン	PostgreSQL バージョン
2.5以降	12、13、14
2.4	12、13
2.1～2.3	11、12、13
2.0	11、12
1.7	9.6、10、11、12

※最新情報は[PostgreSQLアップグレード](#)を参照

Zabbix における TimescaleDB の特徴と利点

- Zabbix 4.2 で実験的にサポート開始
- Zabbix 4.4 で公式にサポート

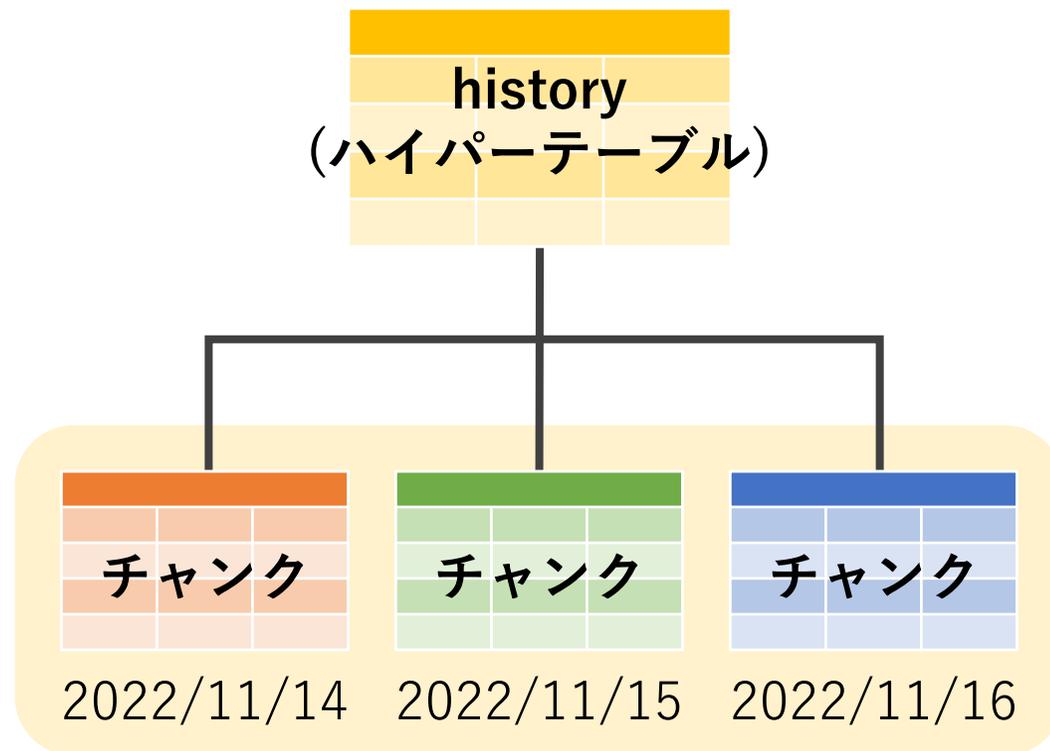
Zabbix	TimescaleDB
4.2	1.0 以降 (OSS バージョン)
4.4	1.0 以降 (OSS バージョン)
5.0 LTS	Zabbix 5.0.0 - 5.0.9: 1.x (OSS バージョン) Zabbix 5.0.10 以降: 1.x、2.0.1 から 2.7
5.2	Zabbix 5.2.0 - 5.2.5: 1.x (OSS バージョン) Zabbix 5.2.6 以降: 1.x、2.x
5.4	1.5 から 2.1
6.0 LTS	2.0.1 から 2.7 (コミュニティバージョン)
6.2	2.0.1 から 2.7 (コミュニティバージョン)

SRA OSS TimescaleDB のハイパーテーブル機能

- Zabbix DB ではハイパーテーブル機能を利用して以下のテーブルをチャンクとしてパーティショニング

- history
- history_uint
- history_log
- history_text
- history_str
- trends
- trends_uint

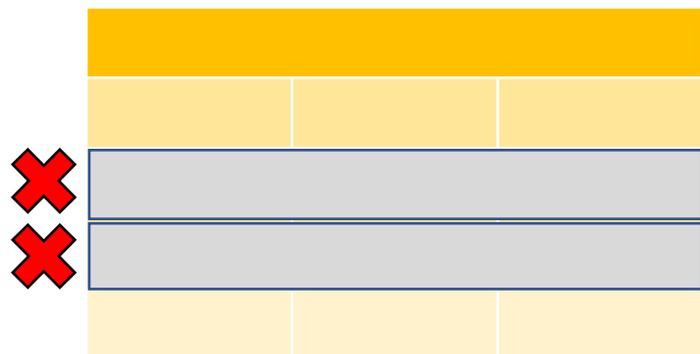
- チャンクは history 系 1 日 trends 系 30 日単位
- 保存期間はグローバル設定で上書き



履歴データ削除 (housekeeper) の違い

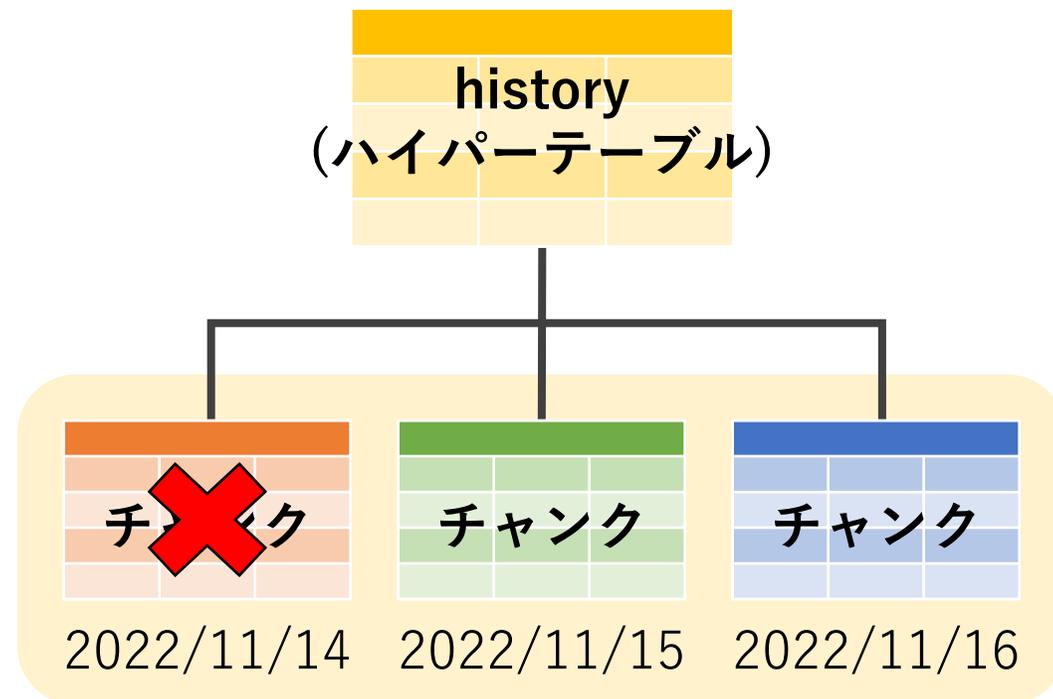
PostgreSQL

該当行を検索して削除
不要領域回収に要 VACUUM



TimescaleDB

チャンクごと削除
VACUUM 不要



- Zabbix 5.0 より TimescaleDB の圧縮機能をサポート
 - PostgreSQL 10.2 以降かつ TimescaleDB 1.5 以降のコミュニティバージョンが必要
 - 過去の履歴データを圧縮形式で保存 (最短 7 日より古いデータ)
- ストレージの節約に有用
- 「管理」→「一般設定」→「データの保存期間」

ヒストリとトレンドの圧縮

圧縮の有効化

* 次よりも古いものはレコードを圧縮

TimescaleDBのインストール

- RHEL に PostgreSQL インストール済み、サーバ稼働中の状態が前提
- TimescaleDB の Yum リポジトリを登録

```
$ sudo tee /etc/yum.repos.d/timescale_timescaledb.repo <<EOF
[timescale_timescaledb]
name=timescale_timescaledb
baseurl=https://packagecloud.io/timescale/timescaledb/el/¥$releasever/¥$basearch
repo_gpgcheck=1
gpgcheck=0
enabled=1
gpgkey=https://packagecloud.io/timescale/timescaledb/gpgkey
sslverify=1
sslcacert=/etc/pki/tls/certs/ca-bundle.crt
metadata_expire=300
EOF
```

- TimescaleDB の RPM パッケージをインストール
 - コミュニティ機能を含まない OSS パッケージは timescaledb-oss

```
$ sudo yum -y install timescaledb-2-postgresql-14-2.7.2
```

(省略)

Installed:

```
timescaledb-2-loader-postgresql-14-2.7.2-0.el8.x86_64 ← ライブラリローダ  
timescaledb-2-postgresql-14-2.7.2-0.el8.x86_64 ← 本体  
timescaledb-tools-0.14.1-0.el8.x86_64 ← クライアントツール
```

Complete!

- 最新バージョンだと **Zabbix** のサポートバージョン外の場合があるためバージョン明示推奨

- TimescaleDB 向けに PostgreSQL の設定を行う

```
$ sudo timescaledb-tune --pg-config=/usr/pgsql-14/bin/pg_config
Using postgresql.conf at this path:
/var/lib/pgsql/14/data/postgresql.conf

Is this correct? [(y)es/(n)o]: y
Writing backup to:
/tmp/timescaledb_tune.backup202210171441

shared_preload_libraries needs to be updated
Current:
#shared_preload_libraries = ''
Recommended:
shared_preload_libraries = 'timescaledb'
Is this okay? [(y)es/(n)o]: y
success: shared_preload_libraries will be updated

Tune memory/parallelism/WAL and other settings? [(y)es/(n)o]: y
(省略)
Saving changes to: /var/lib/pgsql/14/data/postgresql.conf
```

- `timescaledb-tune`はCPU数やメモリ量をもとに対話的に設定を行うツール
- 最低限必要な設定はカンマ区切りで`shared_preload_libraries`に`timescaledb`を追加するだけ

- 通常通り PostgreSQL で Zabbix DB を構築

```
$ sudo -u postgres createuser --pwprompt zabbix
$ sudo -u postgres createdb -O zabbix zabbix
$ zcat /usr/share/zabbix-sql-scripts/postgresql/server.sql.gz | sudo -u zabbix psql zabbix
(省略)
COMMIT
```


- TimescaleDB 用のスクリプトを実行

```
$ cat /usr/share/zabbix-sql-scripts/postgresql/timescaledb.sql | sudo -u zabbix psql zabbix  
(省略)  
TimescaleDB is configured successfully
```

- history 系および trends 系
テーブルのハイパーテーブル化
- アイテムのヒストリ/トレンドの
保存期間設定を上書きを有効化
- 圧縮を 7 日で有効化

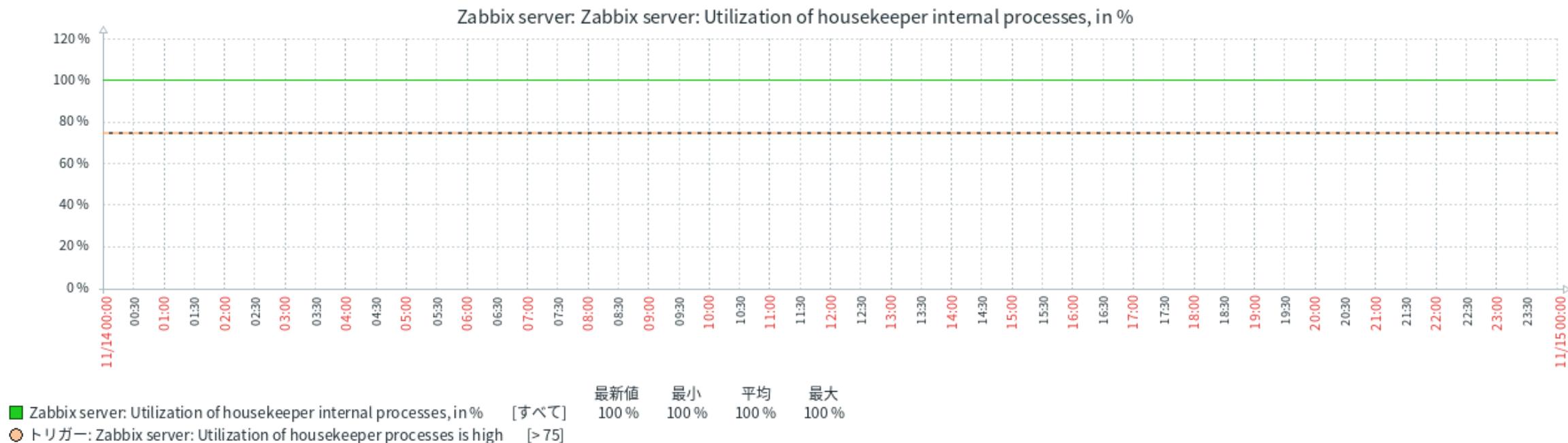
- 既存の PostgreSQL から移行する場合にはこの手順のみで OK

TimescaleDBのパフォーマンス

- OS: CentOS 8.6
- リソース
 - CPU: 2 コア
 - メモリ: 4GB
- ソフトウェアバージョン
 - Zabbix: 6.0.9
 - PostgreSQL: 14.5
 - TimescaleDB: 2.7.2
- 監視設定
 - ホスト数: 500
 - アイテム数: 約 35,000
 - NVPS: 約 400

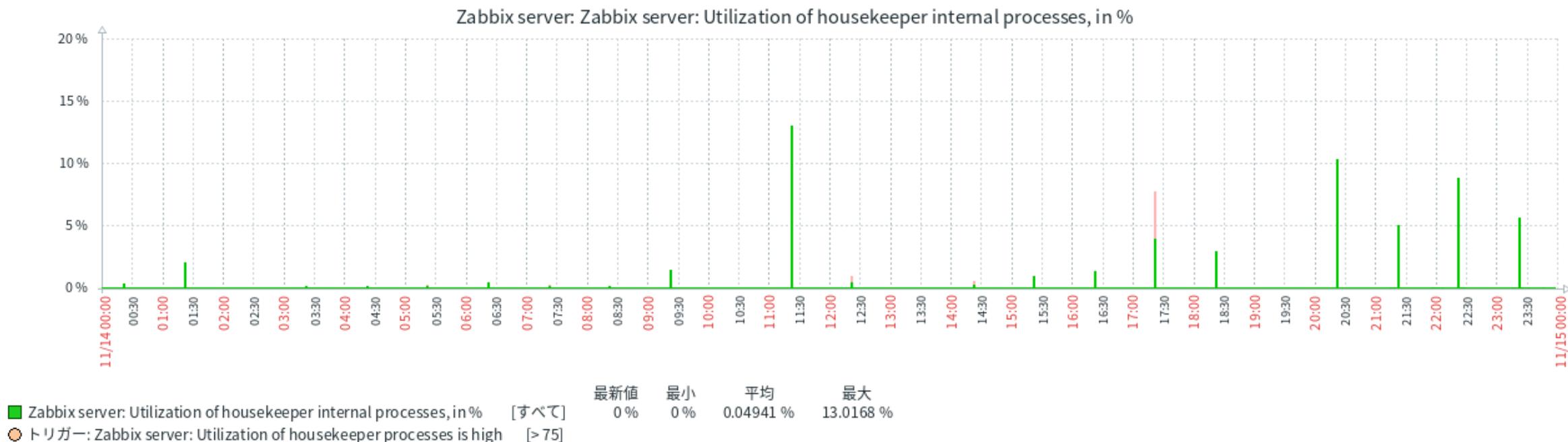
- PostgreSQL と TimescaleDB を Zabbix DB で利用し
データ削除と更新における Zabbix およびリソースの負荷を
プロセスの busy 率と CPU 負荷で比較
- データベースのチューニングは timescaledb-tune の結果を
両データベースに適用
- Zabbix は各キャッシュとデータ収集プロセス (poller) の数を
適切にチューニング

- Utilization of housekeeper internal processes, in %



104627:20221108:053713.876 housekeeper [deleted 5915630 hist/trends, 0 items/triggers, 0 events, 610 problems, 0 sessions, 0 alarms, 0 audit, 0 records in **71897.577989** sec, idle for 1 hour(s)]

- Utilization of housekeeper internal processes, in %



```
2058:20221114:112351.350 housekeeper [deleted 0 hist/trends, 0 items/triggers, 0 events, 2064 problems, 0 sessions, 0 alarms, 0 audit, 0 records in 6.984659 sec, idle for 1 hour(s)]
```

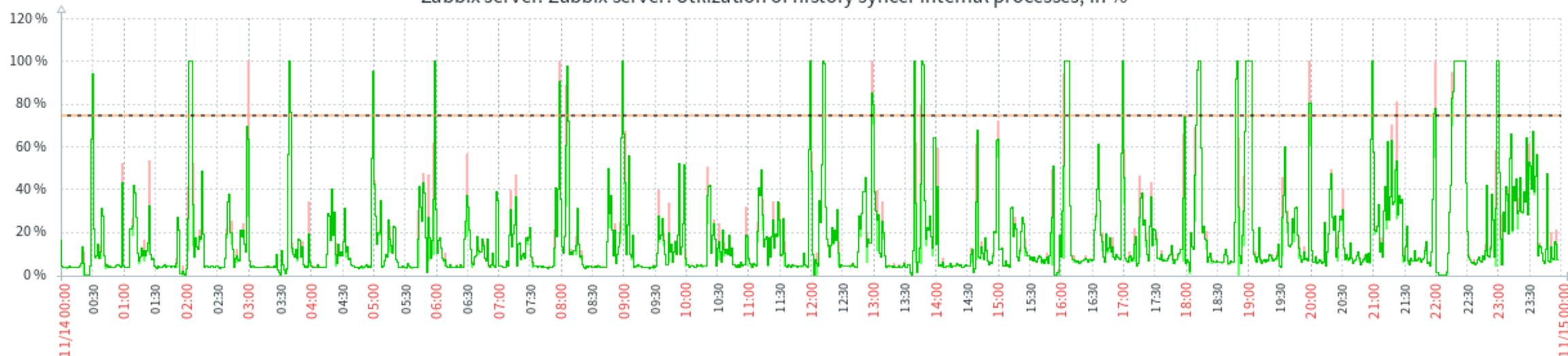
- Utilization of housekeeper internal processes, in %

	最小値	平均値	最大値	最大継続時間
PostgreSQL	100%	100%	100%	71897.577989 sec
TimescaleDB	0%	0.04941%	13.0168%	6.984659 sec
比較結果	変化なし	大幅負荷ダウン	大幅負荷ダウン	大幅負荷ダウン

- 結論
 - 全体的に大幅に負荷は軽減
 - 特にヒストリデータの削除にかかる時間は劇的に減少

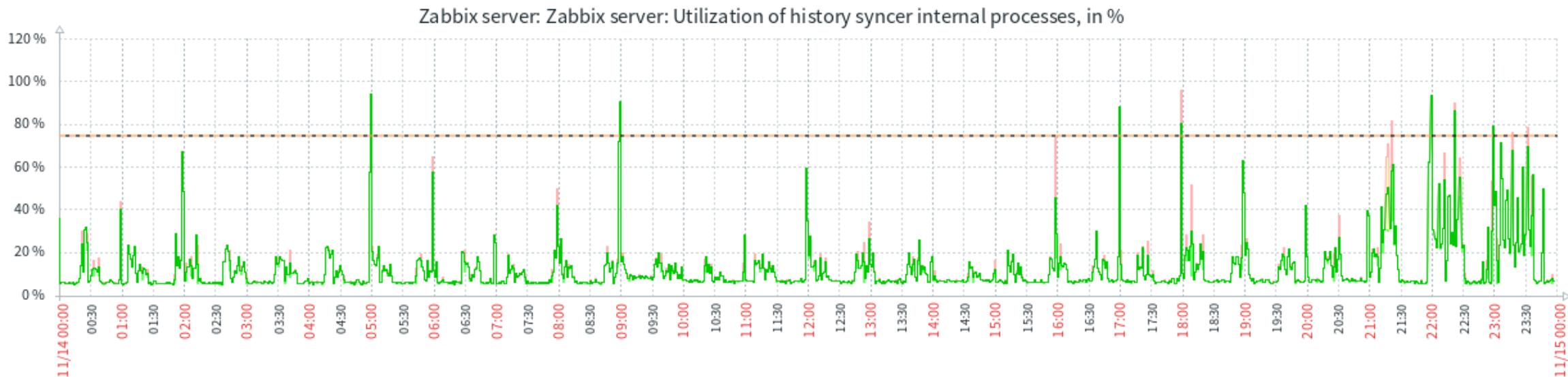
- Utilization of history syncer internal processes, in %

Zabbix server: Zabbix server: Utilization of history syncer internal processes, in %



■ Zabbix server: Utilization of history syncer internal processes, in %	[すべて]	最新値	最小	平均	最大
● トリガー: Zabbix server: Utilization of history syncer processes is high	[> 75]	6.9238 %	0 %	17.3844 %	100 %

- Utilization of history syncer internal processes, in %



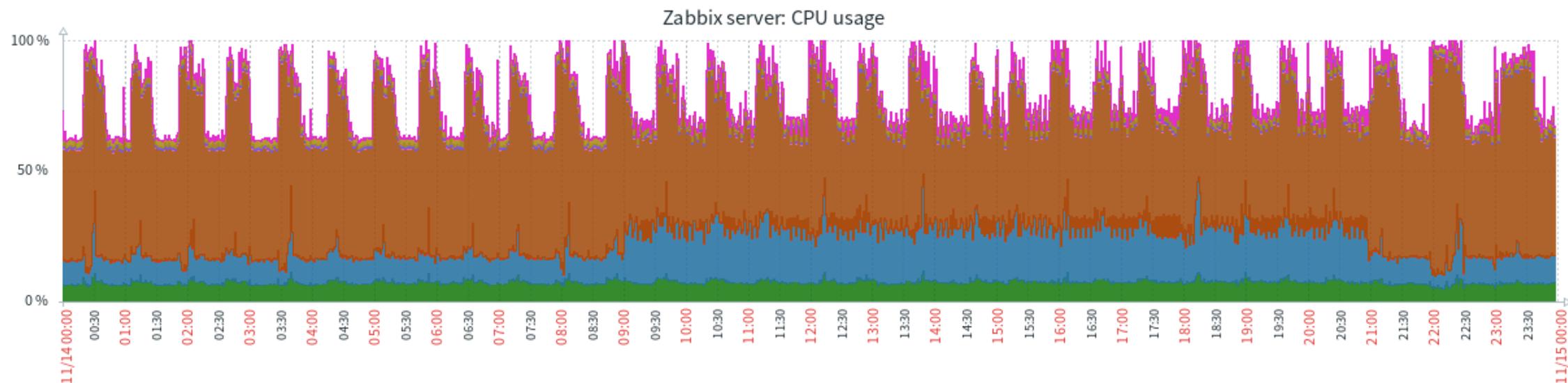
	最新値	最小	平均	最大
Zabbix server: Utilization of history syncer internal processes, in % [すべて]	6.6848 %	4.7242 %	11.7819 %	96.1848 %
トリガー: Zabbix server: Utilization of history syncer processes is high [> 75]				

- Utilization of history syncer internal processes, in %

	最小値	平均値	最大値
PostgreSQL	0%	17.3844%	100%
TimescaleDB	4.7242%	11.7819%	96.1848%
比較結果	多少負荷アップ	多少負荷ダウン	ほぼ変化なし

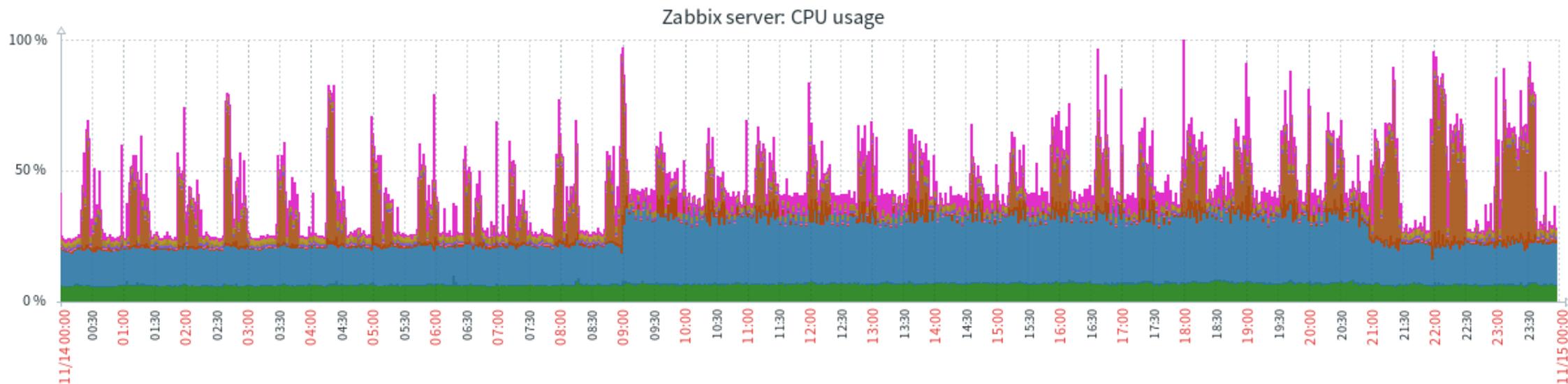
- 結論**
 - 平均的に負荷は多少軽減
 - 特に負荷のかかる頻度は軽減

- CPU usage



		最新値	最小	平均	最大
CPU guest nice time	[平均]	0 %	0 %	0 %	0 %
CPU guest time	[平均]	0 %	0 %	0 %	0 %
CPU softirq time	[平均]	2.9121 %	2.1606 %	2.6499 %	4.0659 %
CPU interrupt time	[平均]	0.9632 %	0.7954 %	1.0139 %	1.3595 %
CPU steal time	[平均]	0 %	0 %	0 %	0 %
CPU iowait time	[平均]	45.4622 %	31.6402 %	51.8623 %	86.2277 %
CPU nice time	[平均]	0 %	0 %	0.01117 %	1.7747 %
CPU user time	[平均]	10.416 %	4.2833 %	16.3526 %	52.3483 %
CPU system time	[平均]	6.9572 %	4.448 %	7.0239 %	13.8979 %

- CPU usage



		最新値	最小	平均	最大
CPU guest nice time	[平均]	0 %	0 %	0 %	0 %
CPU guest time	[平均]	0 %	0 %	0 %	0 %
CPU softirq time	[平均]	2.6034 %	2.2792 %	2.5682 %	3.0524 %
CPU interrupt time	[平均]	1.0049 %	0.7861 %	0.9696 %	1.3826 %
CPU steal time	[平均]	0 %	0 %	0 %	0 %
CPU iowait time	[平均]	1.3144 %	0.2006 %	11.1856 %	78.7239 %
CPU nice time	[平均]	0 %	0 %	0.01315 %	1.7754 %
CPU user time	[平均]	16.6261 %	10.6069 %	21.4753 %	35.7818 %
CPU system time	[平均]	5.8632 %	4.9757 %	6.2684 %	9.2242 %

- CPU iowait time

	最小値	平均値	最大値
PostgreSQL	31.6402%	51.8623%	86.2277%
TimescaleDB	0.2006%	11.1856%	78.7239%
比較結果	大幅負荷ダウン	大幅負荷ダウン	ほぼ変化なし

- **結論**

- 全体的に I/O 負荷が軽減

- TimescaleDBはPostgreSQLベースの時系列データベース
 - DBaaS 提供をしなければ、すべての機能を自由に利用可能
 - NoSQL とリレーショナルデータベースの特長をあわせもつ
 - データは時間で分割されたチャンクに格納され、ハイパーテーブルを通じてアクセス
- Zabbix DB に利用する場合の特徴と利点
 - 履歴系テーブルのパーティショニングによる削除処理の負荷軽減
 - 圧縮機能によるストレージの節約
- PostgreSQL とのパフォーマンス比較結果
 - 履歴データについて削除は大幅に、データ更新は比較的負荷軽減を観測



 www.sraoss.co.jp  sales@sraoss.co.jp  [03-5979-2701](tel:03-5979-2701)