PostgreSQL カンファレンス 2013 (2013.11.08)

KS Solutions

もう安心！
DB安定稼働に向けたPostgreSQL性能診断

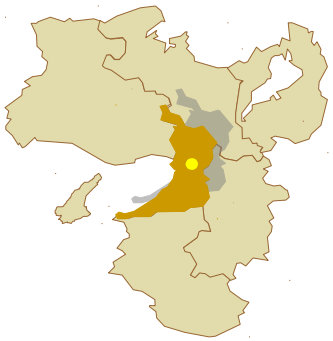
平成25年11月8日
関電システムソリューションズ株式会社
松添 隆康、今井 大嘉

会社紹介　概要

関電システムソリューションズ株式会社

設立	：　2004年10月1日 ※関電情報システム株式会社（1967年4月設立）と株式会社関西テレコムテクノロジー（1986年5月設立）は2004年10月1日に合併し関電システムソリューションズ株式会社となりました。
資本金	：　9,000万円
売上高	：　357億（2012年度実績）
株主構成	：　関西電力株式会社100%出資
従業員数	：　1,168名（2013年7月1日現在）
主な営業品目	：情報通信システムのコンサルティング、情報化戦略の立案 情報通信システムの計画、設計、構築、保守、運用管理 情報通信アプリケーションサービスの開発、提供 情報通信システム設備管理・運用のアウトソーシング
主な受注先	：　関西電力(株)、関西電力グループ会社、法人 、地方自治体
グループ会社	：　関西コンピュータサービス株式会社（KCS） 関西レコードマネジメント株式会社（KRM） 中央コンピューター株式会社（CCC）

（参考）都市型の新データセンター



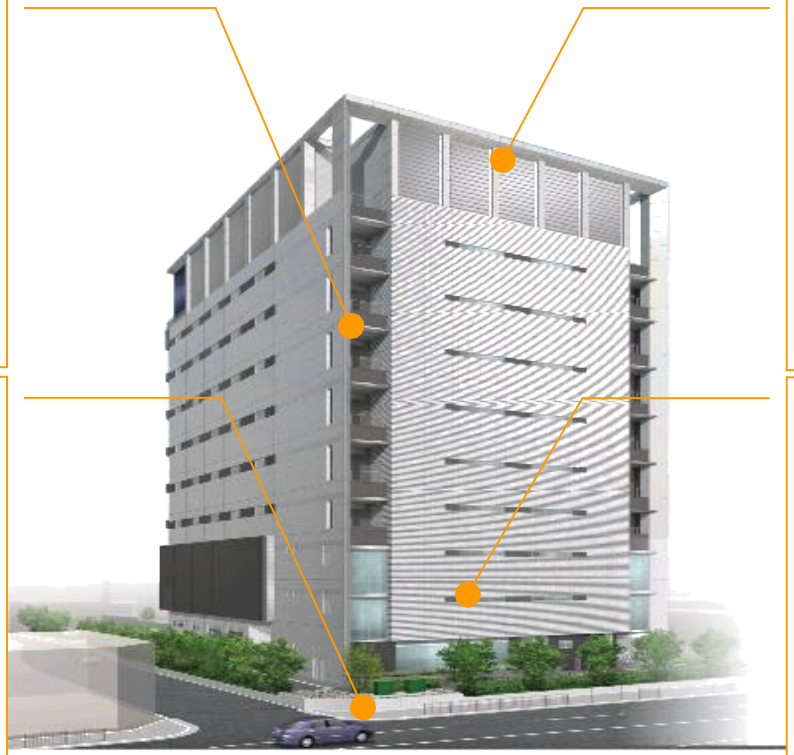
大阪第１データセンター
（大阪市北区　2001年10月～）

大阪第２データセンター
（大阪市福島区　2002年10月～）

大阪第３データセンター
（大阪市北区　2011年12月～）

監視ルーム
24時間365日お預かりしたシステムの稼働状況を有人監視。万が一のトラブルにも技術者が即座に対応いたします。

免震システム
通常の耐震基準を超える大地震でも、基準内の揺れに抑制するビル免震システムを採用。

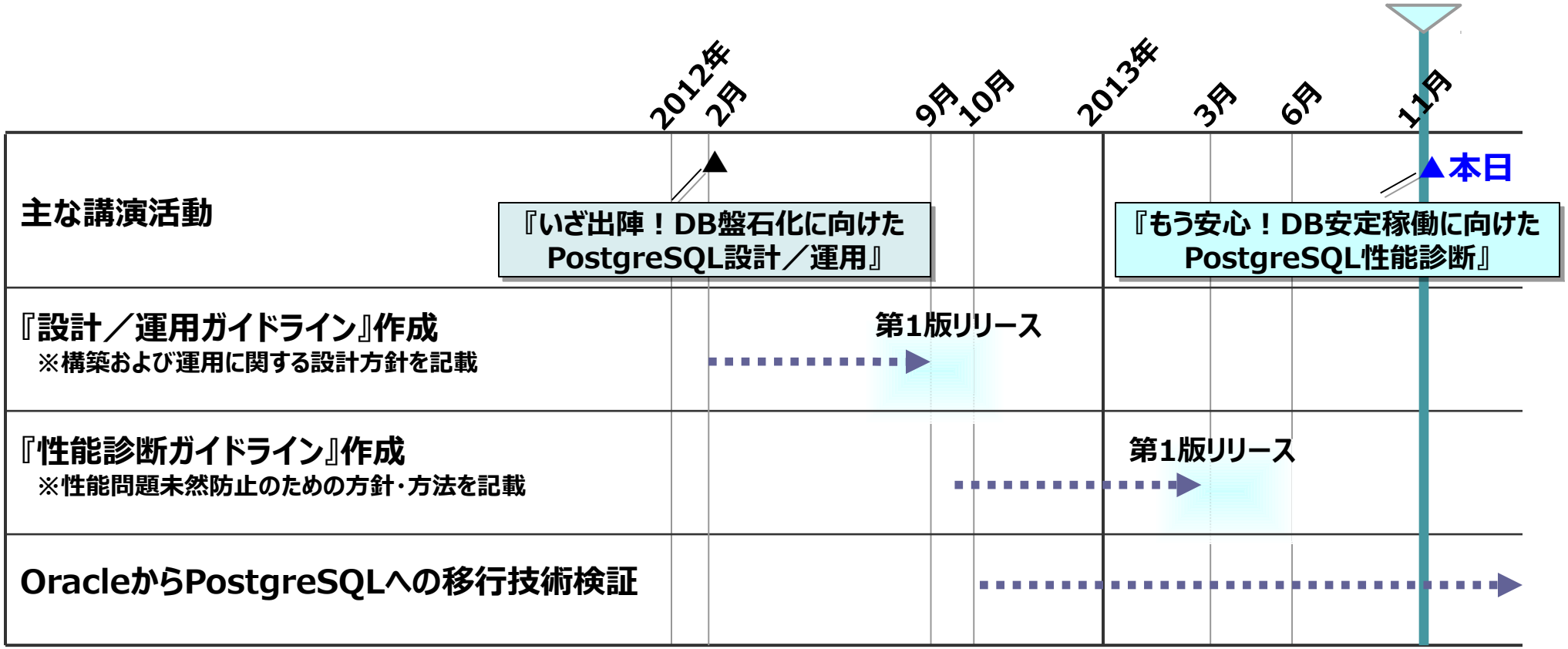


グリーンIT設備
太陽光発電や外気冷却による自然エネルギーを利用。高効率空調システムで国内最高レベルのPUEを実現。

電源設備
3スポット給電、非常用発電機(EG)、無停電電源装置(CVCF)の冗長化による止まらない受電設備。

当社のPostgreSQL取組状況

- 2012年　2月『いざ出陣！DB盤石化に向けたPostgreSQL設計／運用』 講演
- 2012年　9月『設計/運用ガイドライン』 作成
- 2012年10月『OracleからPostgreSQLへの移行技術検証』 開始（現在も継続中）
- 2013月　3月『性能診断ガイドライン』 作成

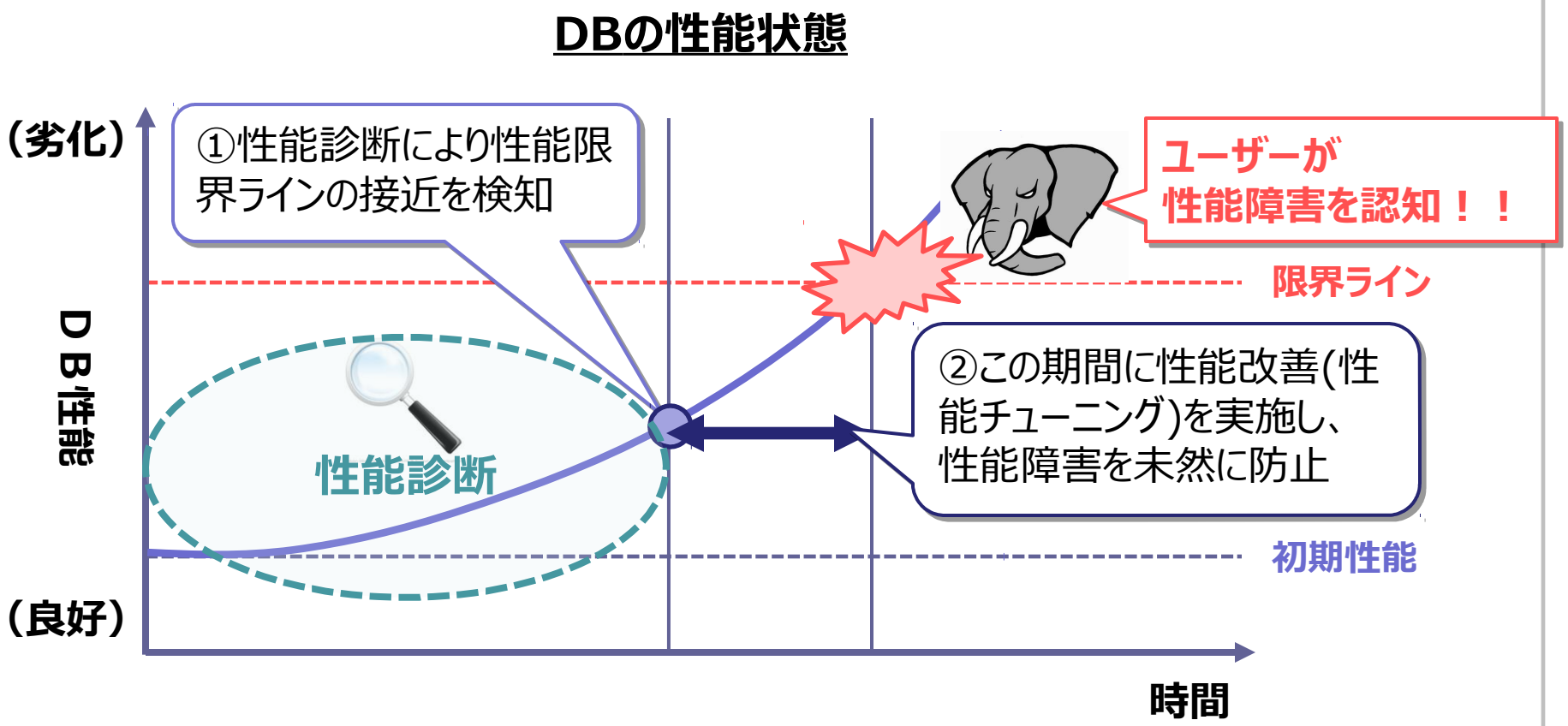


アジェンダ

1.はじめに
2.PostgreSQL性能診断実施のポイント
3.PostgreSQL性能診断の方法

はじめに

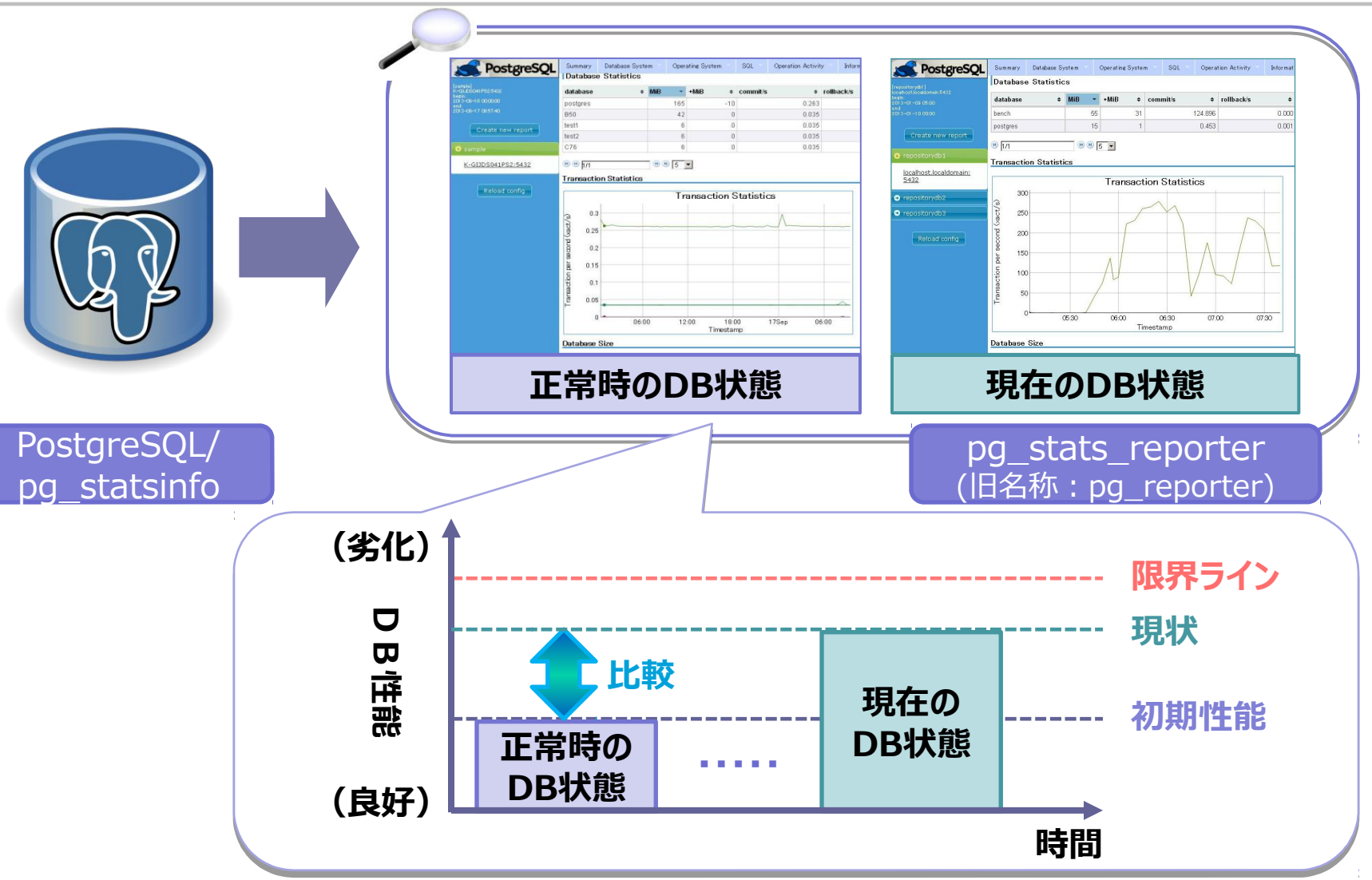
概要	性能診断は日々のDBの性能状態を把握するための作業となるため、 性能改善で実施するような深堀した性能分析は実施しない。



PostgreSQL性能診断実施のポイント

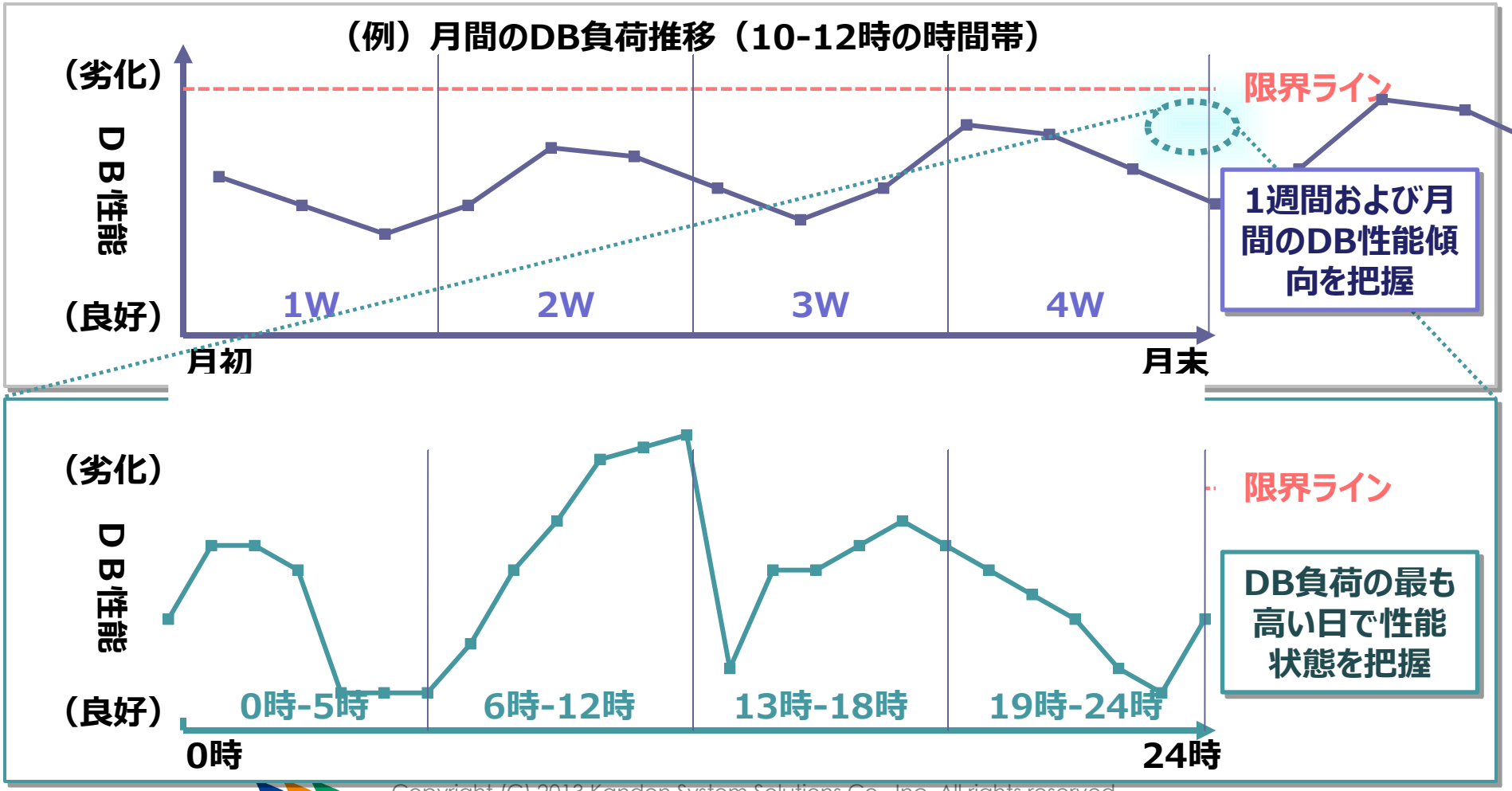
何を確認するのか？

概要	正常時のＤＢ状態(ベースライン)と現在のＤＢ状態を比較し、差異を評価する。



どのような視点で確認するのか？

概要	（DB性能診断の運用例） ➢ DB負荷の高い時間帯の性能傾向を１週間および月間で把握 ➢ DB負荷の最も高い日の性能状態を２４時間で把握



PostgreSQL性能診断の方法

性能診断項目

性能診断の対象	性能診断項目	詳細	sar	pg_stats_reporter	システムカタログ	サーバーログ
OS	システム概況診断	CPU診断	●	ー	ー	ー
		メモリ診断	●	ー	ー	ー
		ディスク診断	●	ー	ー	ー
PostgreSQL	トランザクション・SQL診断	トランザクション数	ー	●	ー	ー
		実行時間の長いトランザクション	ー	●	ー	ー
		高負荷SQL	ー	●	ー	ー
		高負荷関数	ー	●	ー	ー
		大量の一時ファイルを使用するSQL	ー	ー	ー	●
	オブジェクト診断	テーブルスペース・ディスクサイズの増加傾向	ー	●	ー	ー
		テーブル・ディスクサイズの増加傾向	ー	●	ー	ー
		オブジェクトサイズの肥大化	ー	●	ー	ー
		未使用インデックス	ー	●	ー	ー
	自動VACUUM診断	自動VACUUM概況	ー	●	ー	ー
		自動VACUUM／自動ANALYZEの実行状況	ー	ー	●	ー
	ディスク書込診断	チェックポイント処理	ー	●	ー	●

システム概況診断（CPU診断）

CPU診断 メモリ診断 ディスク診断

概要	システム全体のCPU使用率およびディスクI/O待ち状況を確認する。
診断方法	sar -u

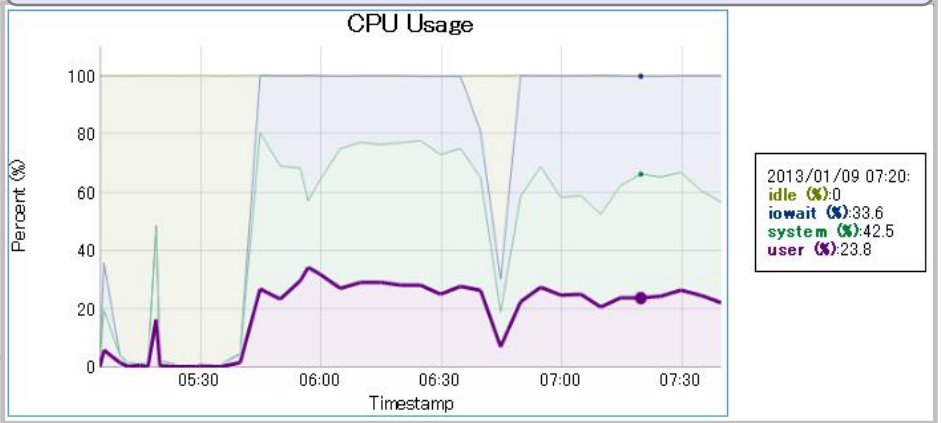
確認項目

項目名	説明
%user	通常のユーザプロセスがCPUを使っている時間の割合
%nice	優先度付きで実行されたユーザプロセスが、CPUを使っている時間の割合
%system	カーネルやシステムプロセスが、CPUを使っている時間の割合
%iowait	I/O待ち時間の割合　（CPUにとっては待ち時間）
%idle	CPUの空き時間の割合(ディスクI/O待ち時間は除く)

sarコマンド実行結果イメージ

```
# sar -u
22時00分02秒   CPU   %user   %nice   %system   %iowait   %steal   %idle
22時10分01秒   all   97.45   0.00    2.55      0.00      0.00     0.00
22時20分01秒   all   97.63   0.00    2.37      0.00      0.00     0.00
22時30分01秒   all   97.34   0.03    2.63      0.00      0.00     0.00
22時40分01秒   all   97.76   0.00    2.24      0.00      0.00     0.00
22時50分01秒   all   58.18   0.00    1.36      0.01      0.00    40.45
23時00分02秒   all    0.28   0.00    2.15     10.52      0.00    87.05
```

(参考)pg_stats_reporter レポート出力結果イメージ

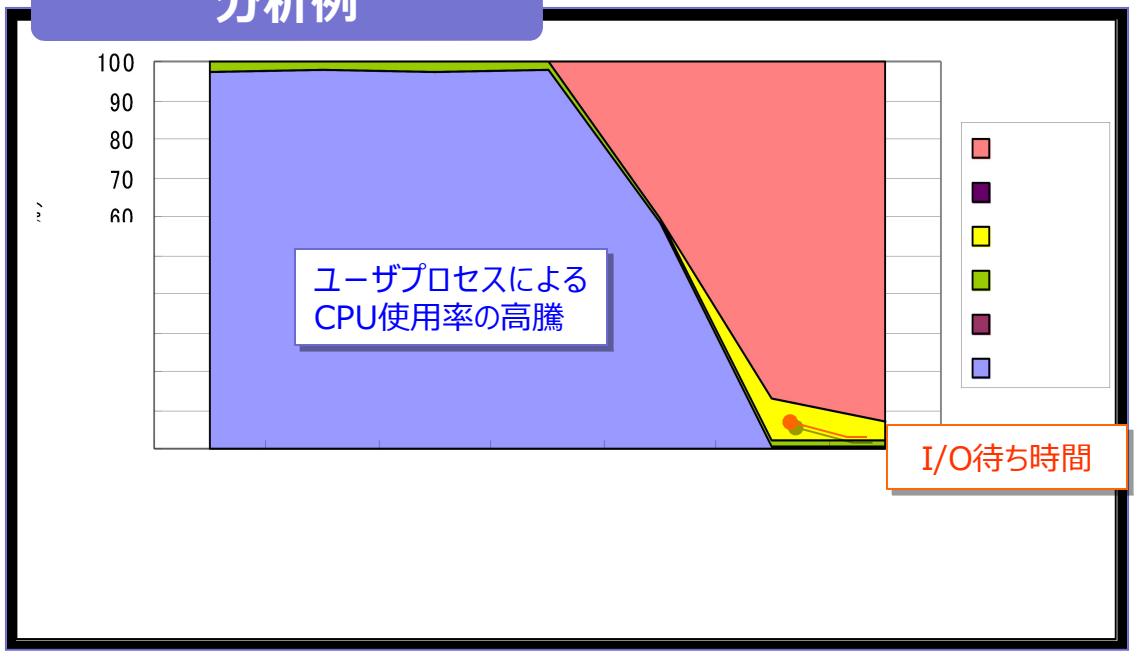


システム概況診断（CPU診断）

分析観点

- ベースラインとの間に大きな差異があるかどうか
- 「%user + %systemで50%を超える」状態が想定ピーク時以外でも断続的に1時間以上続く場合、CPU高騰の可能性が高い
- %iowaitの比率が高い場合、ディスクがボトルネックである可能性が高い

分析例



グラフの前半部では %user が大部分を占めているため、**ユーザプロセスによりCPU使用率が高騰**していることが分かる。

このサーバーは PostgreSQL のみ稼動しているため、この時間帯に **PostgreSQL で非効率な処理が行われていた**可能性が推測できる。

グラフの後半部では %iowait が大きくなっているため、この時間帯に **I/O がボトルネックとなるような処理が行われていた**可能性がある。

システム概況診断（メモリ診断）

CPU診断 | メモリ診断 | ディスク診断

概要	システム全体のスワップ発生状況を確認する。
診断方法	sar -W

確認項目

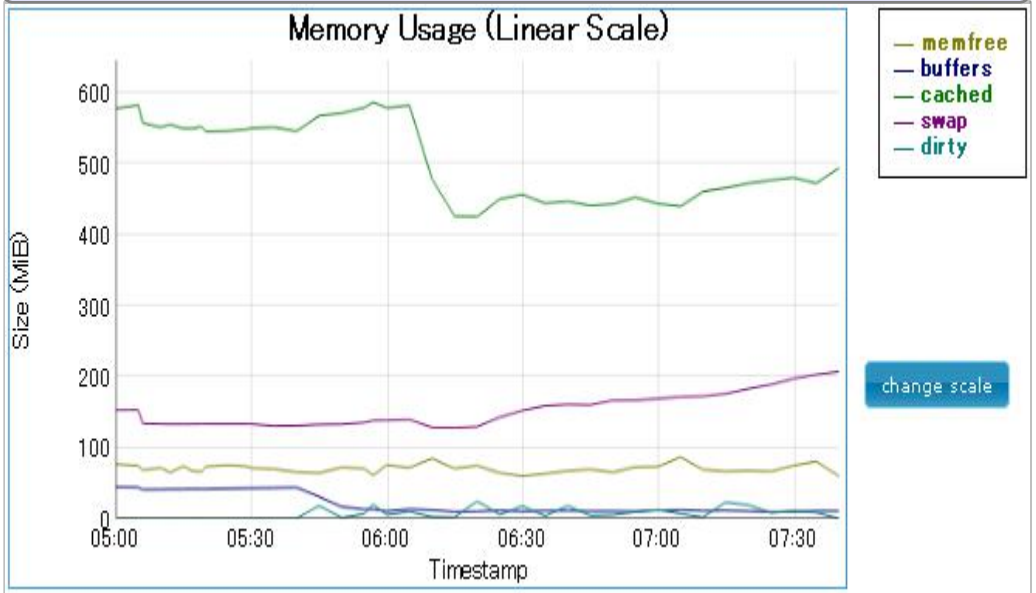
項目名	説明
pswpin/s	ディスク(swap) からメモリに送られたページ数（スワップイン）
pswpout/s	メモリからディスク(swap) に送られたページ数（スワップアウト）

sarコマンド実行結果イメージ

```
# sar -W
11時00分01秒  pswpin/s pswpout/s
11時10分01秒      0.00      0.00
11時20分01秒      0.00      0.00
11時30分01秒      0.00      0.00
11時40分01秒      0.00      0.00
11時50分01秒      0.00      0.00
12時00分01秒      0.00      0.00
12時10分01秒      0.00      0.00
12時20分01秒      0.00      0.00
12時30分01秒      0.00      0.00
12時40分01秒      0.00      0.00
12時50分01秒      0.00      0.00
13時00分01秒      0.00      0.00
13時10分12秒     10.35   1065.21
13時20分02秒     96.88    657.67
13時30分01秒     26.05      0.00
```

メモリに負荷をかけた
※メモリリークするプログラムを実行し、スワップを強制的に発生させている

(参考)pg_stats_reporter レポート出力結果イメージ

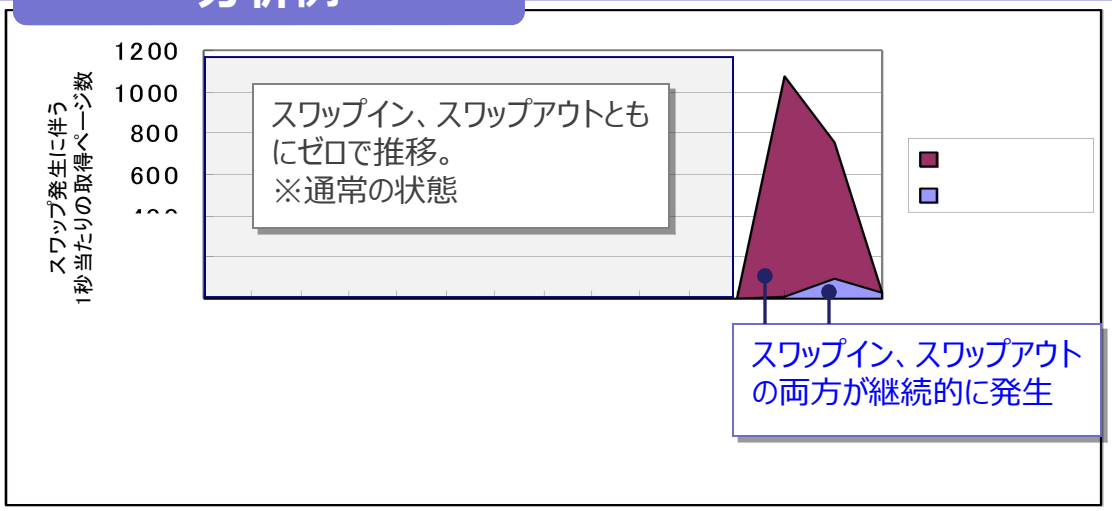


システム概況診断（メモリ診断）

分析観点

- ベースラインとの間に大きな差異があるかどうか
- pswpin/s と pswpout/s の双方が0より大きい状態が継続する場合、メモリ不足の可能性が高い

分析例



グラフの後半部ではスワップイン(pswpin/s)とスワップアウト(pswpout/s)の双方の値が、0より大きい数値で継続的に表示されているため、
メモリ不足の可能性が高いことが考えられる。

システム概況診断（ディスク診断）

概要	システム全体のディスクのビジー率を確認する。
診断方法	sar –d -p

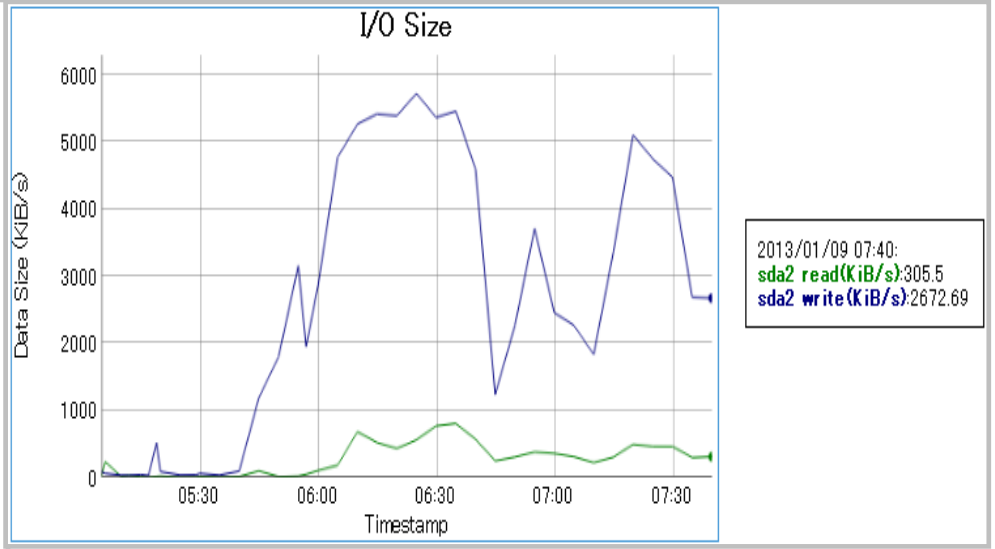
確認項目

項目名	説明
rd_sec/s	1秒当たり読み込みI/Oリクエストのセクタ単位（512 バイト）のデータ量
wr_sec/s	1秒当たり書き込みI/Oリクエストのセクタ単位（512 バイト）のデータ量
%util	ディスクのビジー率

sarコマンド実行結果イメージ

```
# sar -d -p
03時00分01秒   DEV       tps  rd_sec/s  wr_sec/s   ・・・   %util
03時10分01秒   sda    285.31     31.56   2977.34   ・・・   19.28
03時10分01秒   sdb      0.00      0.00      0.00   ・・・    0.00
03時20分01秒   sda   1160.91      0.01  10886.73   ・・・   77.78
03時20分01秒   sdb      0.00      0.00      0.00   ・・・    0.00
03時30分01秒   sda    829.74      0.81   7671.02   ・・・   54.71
03時30分01秒   sdb      0.00      0.12      0.00   ・・・    0.00
03時40分01秒   sda   1189.52    158.63  11035.10   ・・・   81.75
03時40分01秒   sdb      0.01      0.00      0.05   ・・・    0.00
03時50分01秒   sda    451.09    135.24   4207.14   ・・・   35.93
03時50分01秒   sdb      0.12      0.00      0.96   ・・・    0.00
04時00分01秒   sda    897.84     94.17   8226.84   ・・・   63.69
04時00分01秒   sdb      0.11      0.28      1.04   ・・・    0.04
```

(参考)pg_stats_reporter レポート出力結果イメージ



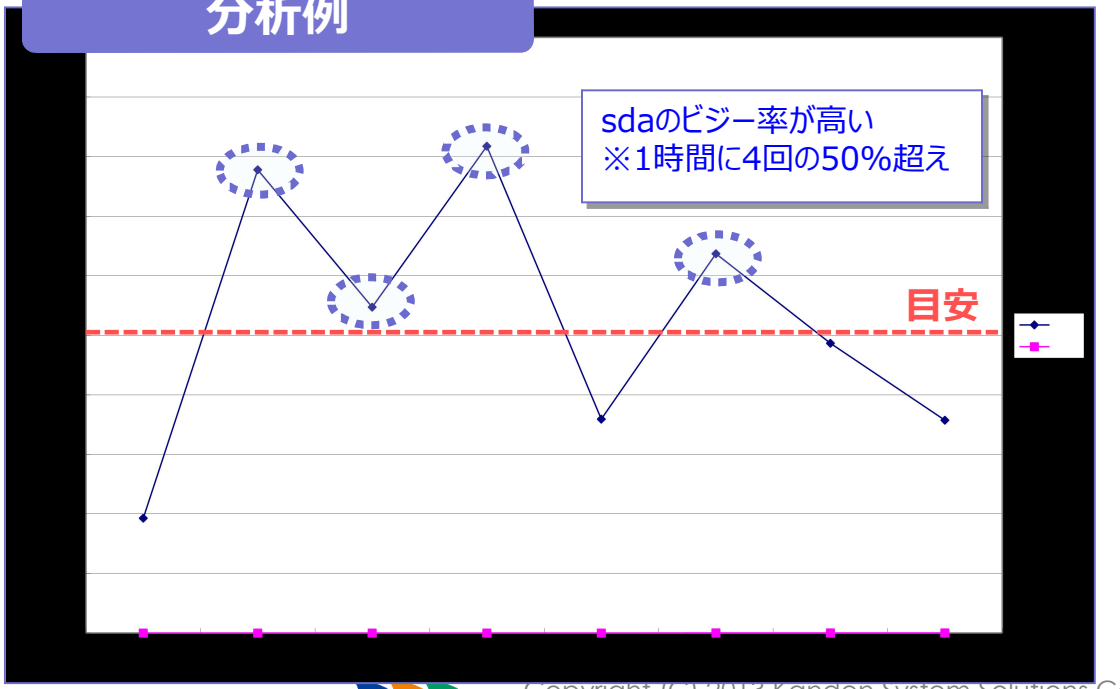
システム概況診断（ディスク診断）

CPU診断　メモリ診断　ディスク診断

分析観点

- CPU診断でI/O待ちが見受けられる場合 (%iowait)、ディスクがボトルネックである可能性が考えられるため、ディスク診断で詳細を確認する
- ベースラインとの間に大きな差異があるかどうか
- ディスクビジー率（%util）が50%を超える状態が、想定ピーク時以外でも1時間に1回以上観測される場合、ディスクがボトルネックである可能性が高い

分析例



５０%を超えるディスクビジー率が４回発生しているため、**ディスク側がボトルネック**である可能性が高いと考えられる。

ボトルネックとなっているディスクの確認が取れた場合、読込み／書込み　いずれの問題であるかを切り分けるため、「読込データ量(rd_sec)」および「書込データ量(wr_sec)」の値を確認する。

トランザクション・SQL診断（トランザクション数）

KS Solutions

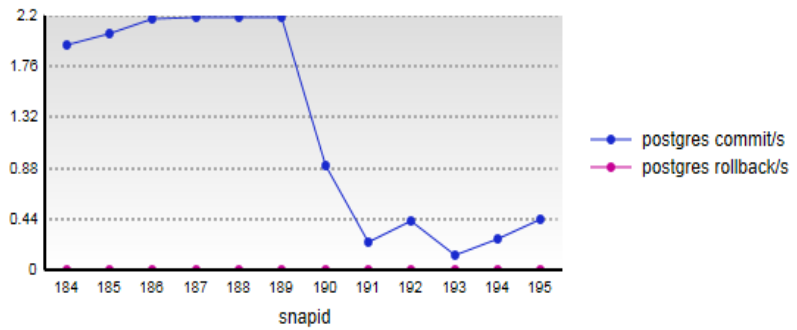
概要	PostgreSQLの処理量が増加傾向にあるかどうかを確認する。
診断方法	pg_reporter ＞ Transaction Statistics

確認項目

項目名	説明
commit/s	1秒あたりのコミット回数
rollback/s	1秒あたりのロールバック回数

pg_reporterレポート出力結果イメージ

Transaction Statistics



分析観点

- ベースラインとの間に大きな差異があるかどうか
- 1か月で10%ずつトランザクション数が増加するなど、中期的な増加傾向がみられるかどうか

⇒普段と違う状態を示している場合、
それが何故起こったのかを
サービスの使用状況やDB運用の観点から
調査を行う。

トランザクション・SQL診断(実行時間の長いトランザクション)

KS Solutions

概要	長時間を要しているトランザクションが存在するかどうかを確認する。
診断方法	pg_reporter ＞ Long Transactions

確認項目

項目名	説明
client address	トランザクションを開始したクライアントのアドレス(サーバ上でトランザクションを実行した場合は空白)
when to start	トランザクションが開始された日時
duration(sec)	トランザクションの実行時間（秒）　※継続中の場合、取得時点での経過時間
query	トランザクション中のSQLのうち、スナップショット取得時点で実行されていたSQL本文

pg_reporterレポート出力結果イメージ

Long Transactions

ID	pid	client address	when to start	duration (sec)	query
1	2545	10.32.0.199	2013-01-15 20:23:40	9379.731	<IDLE> in transaction
2	763		2013-01-15 16:56:15	2975.443	/*LONG SELECT*/ select count(*) from pg_stat_all_tables a,pg_stat_all_tables b,pg_stat_all_tables c,pg_stat_all_tables d,pg_stat_all_tables e;
3	12567		2013-01-15 18:05:52	133.131	select count(*) from test;

分析観点

- ベースラインとの間に大きな差異があるかどうか
- リストされているトランザクションは、ロックを保持したままで長時間何も(コミット、ロールバック)行っていない可能性がある
- リストされているトランザクションは、ロック待ちに陥っている可能性があるため、ボトルネックの根本原因まで把握できない場合がある
- リストに表示されるPostgreSQLの内部処理(VACUUM、COPYなど)は無視する

トランザクション・SQL診断（高負荷SQL）

KS Solutions

概要	ユーザ作成SQLの内、実行時間の長いSQLを確認する。
診断方法	pg_reporter ＞ QueryActivity ＞ Statements

確認項目

項目名	説明
time/call(sec)	SQL１実行あたりの所要時間（秒）

pg_reporterレポート出力結果イメージ

Statements

ID	user	database	query	calls	total time (sec)	time/call (sec)
1	postgre			10814	267.796	0.025
2	postgre		as A left	4	246.012	61.503
3	postgres	postgres	T */ select count(*) from pg_stat_all_tables a,pg_stat_all_tables b,pg_stat_all_tables c,pg_stat_all_tables d) TBL	3	140.932	46.977
4	postgres	postgres	/* FROM CLIENT */ select count(*) from pg_stat_all_tables a,pg_stat_all_tables b,pg_stat_all_tables c,pg_stat_all_tables d	2	92.496	46.248
5	postgre		test as A	1	59.754	59.754
6	postgre		d;	1	48.630	48.630
7	postgres	postgres	insert into test select i,current_timestamp from generate_series(1,500000) as i ;	1	20.915	20.915

１実行あたりの実行時間は短め
※チューニングの余地は限られる

SQL実行時間の合計だけでなく、
1実行あたりの実行時間も長い

分析観点

- ベースラインとの間に大きな差異があるかどうか
- SQL実行に要した合計時間（秒）（total time(sec)）の降順に表示されるので、上位のものから確認する
- 1実行あたり500ミリ秒に満たないSQLはチューニングの余地が限られるため、1実行あたりの時間(time/call(sec))が0.5秒以上で、かつ実行回数が多いSQLをリストアップする
  ※ただし、遅い機能を特定できている場合については、1実行あたり500ミリ秒に満たないSQLであってもリストアップの対象とする
- リストに表示されるPostgreSQLの内部処理（VACUUM、COPY、statsrepoなど）は無視する

トランザクション・SQL診断（高負荷関数）

KS Solutions

概要	ユーザ作成関数の内、実行時間の長い関数を確認する。
診断方法	pg_reporter ＞ QueryActivity ＞ Functions

確認項目

項目名	説明
time/call(ms)	関数１実行あたりの実行時間（ミリ秒）

pg_reporterレポート出力結果イメージ

Functions

ID	Database	Schema	Function	calls	total time (ms)	self time (ms)	time/call (ms)
1	postgres	statsrepo	partition_insert	8493	6355	6355	0.748
2	postgres	statsrepo	alert	12	4890	4890	407.500
3	postgres	public	isvaliddept	180113	3900	3900	0.022
4	postgres	public	isvaliddept2	70	1095	1095	15.643
5	postgres	statsrepo	get_snap_date	8493	661	661	0.078
6	postgres	statsrepo	create_partiti			17	12.750

PostgreSQL内部処理

ユーザ作成関数

1実行あたりの所要時間に大きな差がある。
※isvaliddept関数のチューニング余地は小さい。

分析観点

- ベースラインとの間に大きな差異があるかどうか
- 関数の実行に要した合計実行時間(total time(ms))の降順に表示されるので、上位のものから確認する
- 1実行あたり500ミリ秒に満たない関数はチューニングの余地が限られるため、1実行あたりの時間(time/call(ms))が0.5秒以上で、かつ実行回数が多い関数をリストアップする
  ※ただし、遅い機能を特定できている場合については、1実行あたり500ミリ秒に満たない関数であってもリストアップの対象とする
- 関数全体の実行時間(total time(ms))と、本関数の純粋な実行時間(self time(ms))との差異が大きい場合、本関数に含まれる子関数が多くの時間を使用している可能性がある

トランザクション・SQL診断(大量の一時ファイルを使用するSQL)

KS Solutions

概要	一時ファイルを多く使用するSQLを確認する。
診断方法	PostgreSQLのサーバーログ ※postgresql.confの「log_temp_files」設定値以上の一時ファイルを使用するSQL実行ログを出力

確認項目

PostgreSQL サーバーログ内容

”temporary file：[一時ファイル名], size [一時ファイルのサイズ]”の出力箇所

PostgreSQLサーバーログ出力結果イメージ

```
2013-01-15 18:30:01 JST 16424 LOG:  temporary file: path "base/pgsql_tmp/pgsql_tmp16424.3", size 623386624
2013-01-15 18:30:01 JST 16424 STATEMENT:  SELECT COUNT(*) FROM (select a.c1 from test as A left join test AS B on A.c1 = B.c1) AS X;

2013-01-15 18:30:01 JST 16424 LOG:  temporary file: path "base/pgsql_tmp/pgsql_tmp16424.2", size 623386624
2013-01-15 18:30:01 JST 16424 STATEMENT:  SELECT COUNT(*) FROM (select a.c1 from test as A left join test AS B on A.c1 = B.c1) AS X;
```

一時ファイルのファイル名と使用サイズ

一時ファイルを使用したSQL

分析観点

- **一時ファイルの使用による無駄な I/O は、当該SQLのみならず、PostgreSQL全体の性能に影響を及ぼす可能性があるため、リストアップの対象とする**

オブジェクト診断(テーブルスペース・ディスクサイズの増加傾向)

KS Solutions

概要	テーブルスペースのサイズの増加傾向がディスクI/Oの高騰に繋がっていないかどうかを確認する。
診断方法	pg_reporter ＞ Disk Usage ＞ Disk Usage per Tablespace

確認項目

項目名	説明
used (MB)	テーブルスペース格納ディレクトリのデバイスの使用済みサイズ (MB)
avail (MB)	テーブルスペース格納ディレクトリのデバイスの利用可能サイズ (MB)
remain%	テーブルスペース格納ディレクトリのデバイスで利用可能な領域の割合

pg_reporterレポート出力結果イメージ

Disk Usage per Tablespace

ID	tablespace	location	device	used (MB)	avail (MB)	remain%
1	<pg_xlog>	/pg_wal	8:17	194	1820	90.327
2	pg_default	/usr/local/pgsql/data	8:2	6651	29533	81.618
3	pg_global	/usr/local/pgsql/data	8:2	6651	29533	81.618

分析観点

- ベースラインとの間に大きな差異があるかどうか
- テーブルスペースのディスク使用状況と空き状況から、将来の状態予測を行い、ディスクサイズの増加傾向がI/O の高騰に繋がっていないかどうかを考える
- ディスク空き領域がゼロになると、ソフトウェア動作障害が生じるリスクが高まる(remain%=10以上の確保が望ましい)
- ディスク空き領域の割合が50%(remain%=50)を切った時点から、ディスク性能が徐々に劣化する可能性がある

オブジェクト診断(テーブル・ディスクサイズの増加傾向)

概要	テーブルのサイズの増加傾向がディスクI/Oの高騰に繋がっていないかどうかを確認する。
診断方法	pg_reporter ＞ Disk Usage ＞ Disk Usage per Table

確認項目

項目名	説明
Size(MB)	テーブルサイズ (MB)

pg_reporterレポート出力結果イメージ

Disk Usage per Table

ID	Database	Schema	Table	Size (MB)	table_reads	index_reads	toast_reads
1	postgres	public	test	2561	3390369	383870	0
2	postgres	statsrepo	column_20130203	7	3925	186	0
3	postgres	statsrepo	function	3	507	22	0
4	postgres	statsrepo	column_20130131	6	98	246	0
5	postgres	statsrepo	table_20130203	0	316	22	0
6	postgres	statsrepo	column_20130130	5	103	226	0
7	postgres	statsrepo	column_20130201	6	96	215	0
8	postgres	statsrepo	column_20130115	4	62	186	0
9	postgres	statsrepo	column_20130114	3	48	198	0
10	postgres	statsrepo	column_20130113	3	48	152	0

分析観点

- ベースラインとの間に大きな差異があるかどうか
- テーブルのディスク消費状況から、将来の状態予測を行い、
  ディスクサイズの増加傾向がI/Oの高騰に繋がっていないかどうかを考える
- ディスク消費量の大きいテーブルについて、
  自動VACUUMが適切に行われているかどうか確認する
  （自動VACUUM診断 参照）

オブジェクト診断（オブジェクトサイズの肥大化）

概要	ブロック内の有効な行データの密度が低いテーブルを確認する。
診断方法	pg_reporter ＞ Low Density Tables

確認項目

項目名	説明
logical_pages	テーブル内の有効行が占めるサイズをページ単位(ブロック単位：1ページ 8KB)で表した値
physical_pages	テーブル内に占めている実際の物理ページ数(ブロック数)
tratio	行データ密度。有効な行データが占める割合(logical_pages／physical_pages)

pg_reporterレポート出力結果イメージ

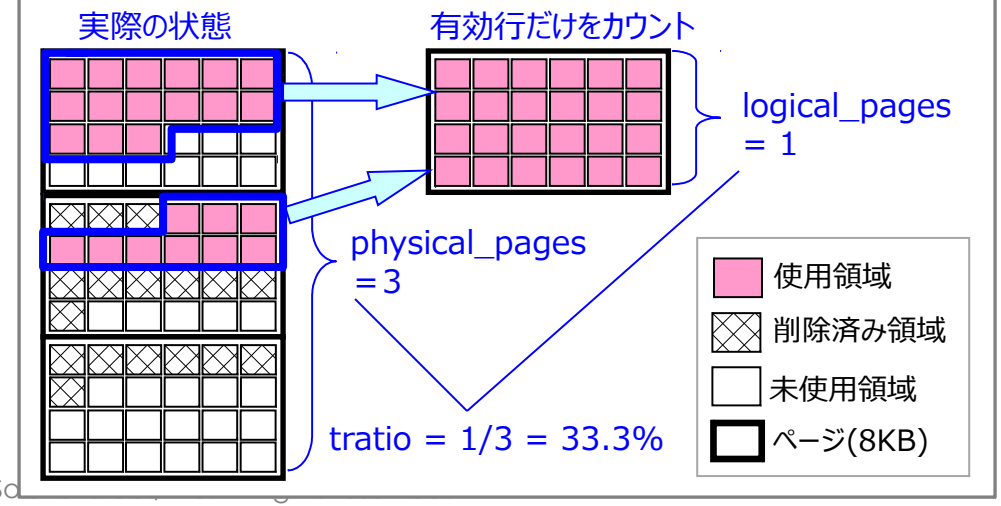
Low Density Tables

ID	Database	Schema	Table	tuples	logical_pages	physical_pages	tratio
1	postgres	public	test	8874537	145485	327869	44

分析観点

- **ベースラインとの間に大きな差異があるかどうか**
- **更新量が少ないにも関わらず、tratioが70(%) を下回るテーブルは、自動VACUUMの実行が不十分な可能性がある(自動VACUUM診断 参照)**

(補足) logical_pages と physical_pages の違い



オブジェクト診断（未使用インデックス）

概要	更新性能劣化の要因となる未使用インデックスが存在しないかどうかを確認する。
診断方法	pg_reporter ＞ Schema Information ＞ Indexes

確認項目

項目名	説明
scans	インデックス・スキャンの実行回数

pg_reporterレポート出力結果イメージ

Indexes

ID	Database	Schema	Index	Table	size (MB)	incremental size (MB)	scans	rows/scan	blks_read	blks_hit	keys
1	postgres	public	dept_pkey	dept	1	0	0	0.000	0	0	deptno
2	postgres	public	test_pkey	test	428	0	11	0.182	383870	30	c1

scansの値が0。
レポート期間中はこのインデックスを使用していない。

分析観点

- scans＝0（未使用）があるかどうか
- 特定の処理でしか使わないインデックスも存在するため、1週間程度の期間では未使用かどうかは判断できない（1か月以上に渡って未使用状態が継続していれば、そのインデックスは未使用である可能性が高い）
  ⇒ scans＝0 がどれぐらい連続すれば未使用とみなすかについては、アプリケーションに依存する部分が大きいため、アプリケーション要件の確認を併せて行う

自動VACUUM診断（自動VACUUM概況）

概要	自動VACUUMが適切に行われているかどうかを確認する。
診断方法	pg_reporter ＞ Autovacuum Activity

確認項目

項目名	説明
avg removed rows	削除された行数の平均値
avg duration (sec)	処理時間の平均値（秒）

pg_reporterレポート出力結果イメージ

Autovacuum Activity

ID	Database	Schema	Table	count	avg index scans	avg removed rows	avg remain rows	avg duration (sec)	max duration (sec)
1	postgres	public	test	2	1.000	3076284.000	9716284.000	1703.950	2262.860

自動VACUUMの実行により、約300万件程度の行を削除している

自動VACUUMの実行により、約1700秒程度を要している

分析観点

- ベースラインとの間に大きな差異があるかどうか
- オブジェクト診断の「テーブル・ディスクサイズの増加傾向」および「オブジェクトサイズの肥大化」でリストアップされたテーブルが本セクションに表示されているかどうかを確認する（因果関係の確認）
- avg removed rows の値が小さい場合、長いトランザクションによって自動VACUUMの実行が阻害された可能性がある

自動VACUUM診断(自動VACUUM/自動ANALYZEの実行状況)

概要	自動VACUUMおよび自動ANALYZEが、いつ実行されたのかを確認する。
診断方法	=# SELECT relname, last_autovacuum, last_autoanalyze, autovacuum_count, autoanalyze_count -# FROM pg_stat_user_tables -# WHERE schemaname <> 'statsrepo';

確認項目

項目名	説明
last_autovacuum	最後に自動VACUUMが実行された日時
last_autoanalyze	最後に自動ANALYZEが実行された日時

実行結果イメージ

```
relname|       last_autovacuum         |      last_autoanalyze         |autovacuum_count|autoanalyze_count
-------+-------------------------------+-------------------------------+----------------+----------------
dept   |2013-01-15 18:03:42.688566+09|2013-01-15 16:47:40.305076+09|               6|               8
test   |2013-02-03 23:43:35.370758+09|2013-02-03 23:05:00.406744+09|               4|               4
```

分析観点

- last_autoanalyze の日時がレポート出力時点に対して古過ぎる日時の場合、適切に自動ANALYZEが実行されていない可能性がある(SQL実行計画が適切に立案されずにパフォーマンス低下に繋がっている可能性がある)
- last_autovacuum の日時がレポート出力時点に対して古過ぎる日時の場合、適切に自動VACUUMが実行されていない可能性がある(テーブルサイズの肥大化の要因になっている可能性がある)

ディスク書込診断（チェックポイント処理）

チェックポイント処理

概要	ディスクI/O の更新量の多さに起因するチェックポイントが多発していないかどうかを確認する。
診断方法	pg_reporter ＞ Checkpoint Activity

確認項目

項目名	説明
checkpoints by time	checkpoint_timeout(デフォルト 5分)契機で実行されたチェックポイントの回数
checkpoints by xlog	checkpoint_segments(デフォルト 3)契機で実行されたチェックポイントの回数

pg_reporterレポート
出力結果イメージ

Checkpoint Activity

total checkpoints	68
checkpoints by time	24
checkpoints by xlog	44
avg written buffers	397.015
max written buffers	2408.000
avg duration (sec)	13.482
max duration (sec)	151.199

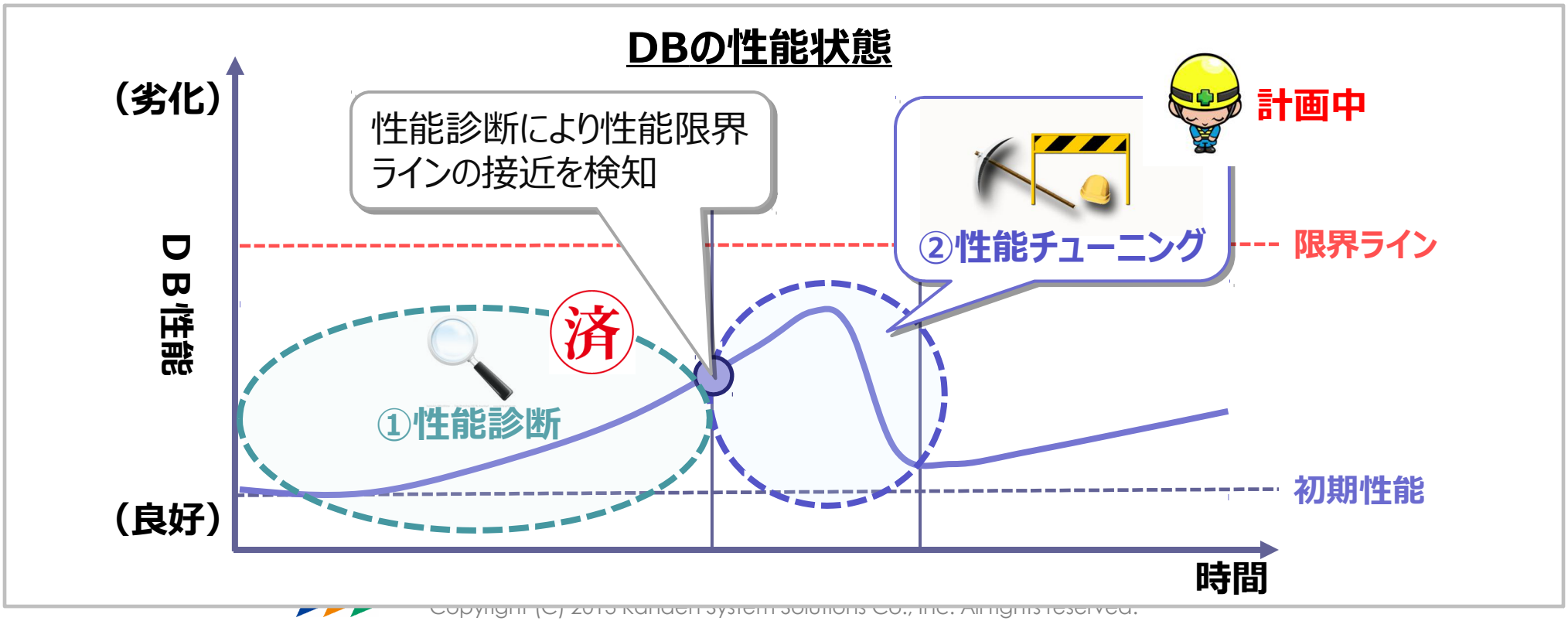
分析観点

- **ベースラインとの間に大きな差異があるかどうか**
- **checkpoints by xlog の値が checkpoints by time の3倍以上（目安）の場合、更新量の多いチェックポイントが多発している可能性がある**
- **チェックポイント実行間隔が30秒（checkpoint_warningのデフォルト値）未満の場合、サーバログに以下の警告メッセージが出力される**
  このようなログ出力が多発している場合、ディスクI/Oが高騰している可能性がある

```
LOG: checkpoints are occurring too frequently
HINT: Consider increasing the configuration parameter "checkpoint_segments"
```

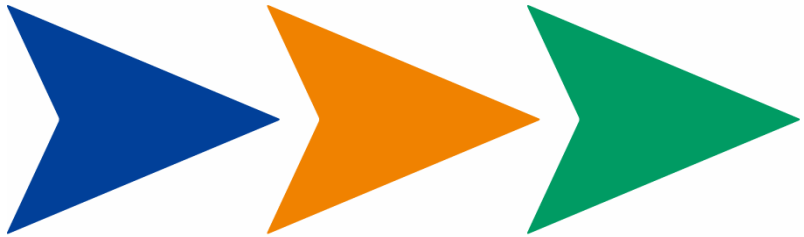
まとめ

- 性能診断は性能問題の発覚時に実施する作業(性能チューニング)ではなく、
  性能問題を未然に防止するための作業
- 当社では、PostgreSQL性能診断業務への浸透を図るため、
  ドキュメントを作成し、社内にて説明会を実施
  (現場からの声をドキュメントにフィードバックし、随時改善を図っている)
- 今後は発覚した性能問題を解決するための性能チューニングに関してドキュメント作成予定



ご清聴、ありがとうございました。

あしたをつくる革新を、ともに。



Innovate with you