

データベースシステムの監視

監視の概要

技術を知る

- | | | | | |
|-----------------------------------|--|--------------------------------|---------------------------------|---------------------------------------|
| <input type="checkbox"/> 導入／環境設定 | <input type="checkbox"/> 移行 | <input type="checkbox"/> 性能 | <input type="checkbox"/> チューニング | <input type="checkbox"/> バックアップ／リカバリー |
| <input type="checkbox"/> 冗長化／負荷分散 | <input checked="" type="checkbox"/> 監視 | <input type="checkbox"/> データ連携 | <input type="checkbox"/> 災害対策 | <input type="checkbox"/> 豆知識 |

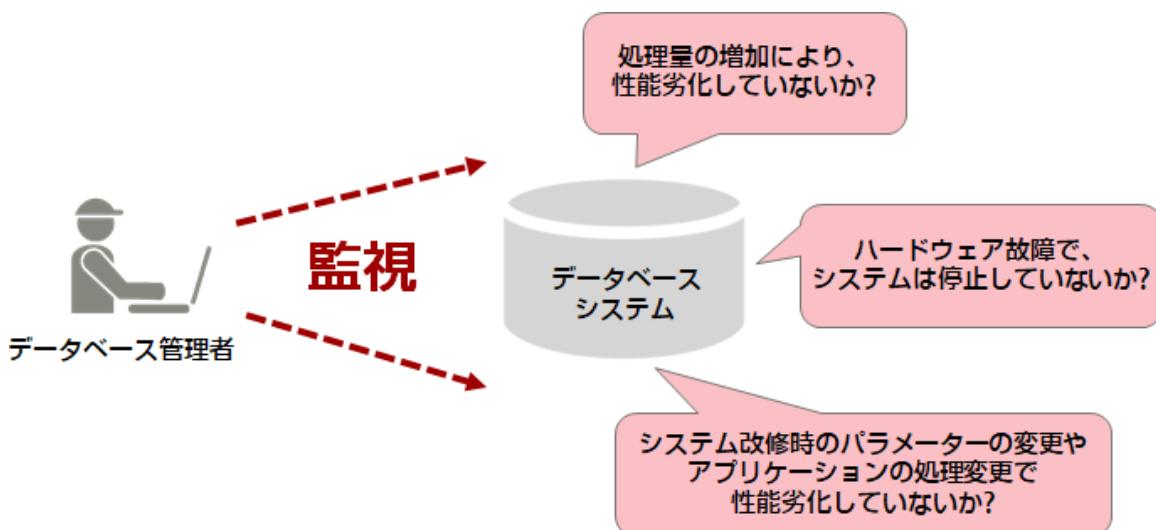
データベースシステムの異常や停止は業務の停止につながるため、あらかじめ、異常が発生したときの対策や、未然防止策を立てておく必要があります。これらの対策を行ううえで、「データベースシステムの監視」はとても重要です。データベースシステムが安定稼動している状態を知り、その状態が維持されていることを監視し、異常時の対策を十分に行っておくことで、異常が発生した場合には迅速に対処することができます。また、データベースシステムの情報を定期的に収集し、保存しておくことで、異常が発生したときの原因を特定したり、異常の兆候を検知することができます。

ここでは、PostgreSQL のデータベースシステムの監視対象と監視方法の概要について説明します。本資料は、Linux を前提としています。

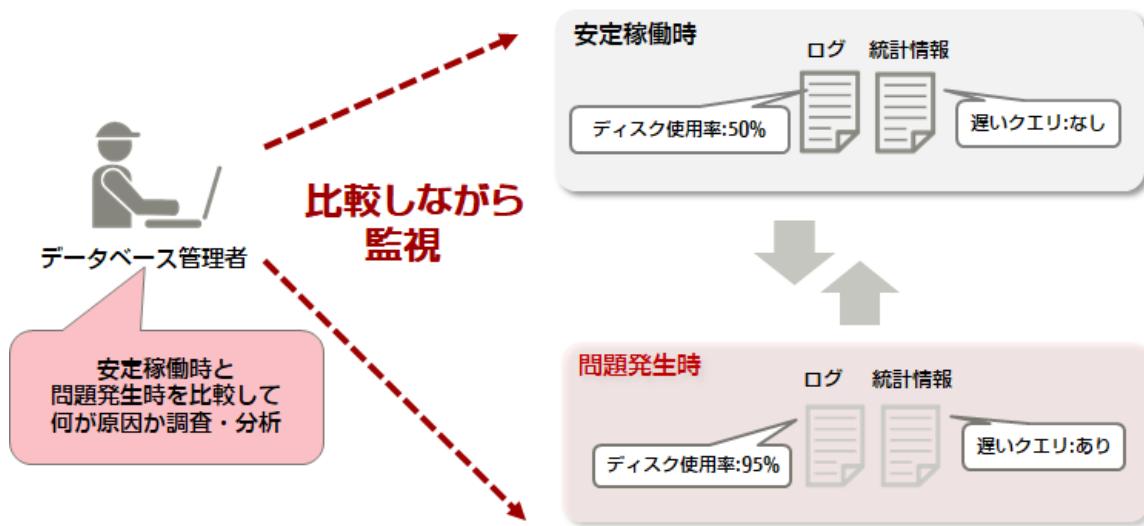
1. データベースシステムの監視とは

データベースシステムの監視は、システムを安定稼働させ続けるために必要であり、何に影響を受けるかを把握し、異常を検出した際に適切に対処する必要があります。

データベースシステムに影響を与える要因には、想定外の処理量の増加やデータ量の増加、システム改修時のパラメーターの変更やアプリケーションの処理変更、ディスク異常などのハードウェア故障などがあり、これらによりデータベースの性能が劣化したり、データベースシステムが停止したりする場合があります。



データベースの異常を検出しても、その原因を特定することが難しい場合があります。そのような場合を想定して、各種統計情報やログ (syslog、サーバーログ) を定期的に収集・保存し、正常時の基準値を把握しておくことが重要です。そうすることで、異常が発生したときの原因特定に利用できます。また、異常発生の予兆判断にも利用できます。



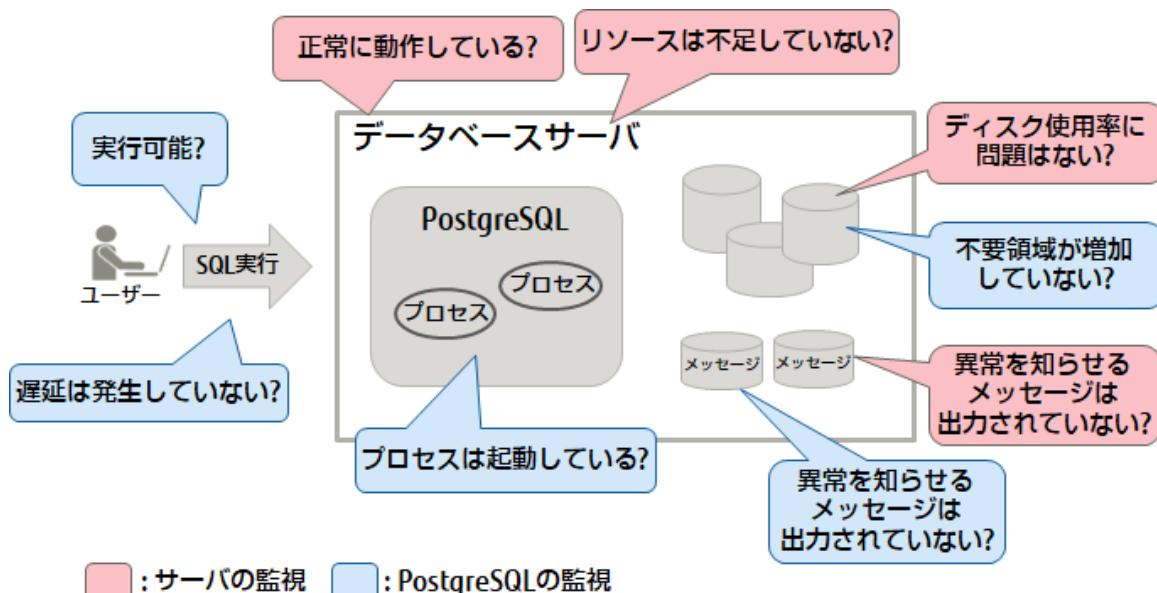
2. データベースシステムの監視対象

「1. データベースシステムの監視とは」で記載した要因は、それぞれ事象として表れます。ここでは、どのような情報を監視すれば良いかを説明します。

データベースシステムの監視を大きく分類すると、「サーバーの監視」と「PostgreSQL の監視」があります。

サーバーの監視は、PostgreSQL を稼働させるサーバー自体が正常に稼働しているか、OS のリソースが不足していないか、syslog に異常を知らせるメッセージが出力されていないか、ディスク使用率に問題は無いかなどを確認します。

PostgreSQL の監視は、PostgreSQL のプロセスが起動しているか、SQL の実行が可能か、サーバーログに異常を知らせるメッセージが出力されていないか、実行した SQL の性能に遅延が発生していないか、不要領域が増加していないなどを確認します。



「サーバーの監視」および「PostgreSQL の監視」の、主な監視対象を以下に示します。

分類	監視対象	
サーバーの監視	サーバーの死活	サーバーOS の応答
	OS のリソース	CPU 使用率
		メモリー使用率
		I/O ビジー率
		ネットワーク帯域使用状況
	メッセージ	syslog (システム内で行われた処理、ハードウェアからのメッセージ (注 1))
	ディスクの容量	使用率、使用容量、空き容量
PostgreSQL の監視	PostgreSQL の死活	PostgreSQL プロセスの有無
		SQL 実行可否 (検索や更新が正常に実行できるか確認)
	メッセージ	サーバーログ
	ディスクの容量	データベースやインデックスで使用されるディスク容量
	性能	遅い SQL の有無
		長時間放置されているトランザクションの有無
		ロック待ち時間の長い SQL の有無
		レプリケーションの遅延状況 (レプリケーション構成の場合)
	性能 (断片化)	不要領域 (データの挿入や削除によって発生した空き領域)
		インデックスのばらつき (インデックスの並び順とテーブルの並び順が揃っているか)

- 注 1 SNMP (Simple Network Management Protocol) によるメッセージを syslog に出力した場合

3. データベースシステムの監視方法

データベースの監視は、「2. データベースシステムの監視対象」で述べた監視対象を、「どう」監視するのかが、とても重要です。

ここでは、監視方法や監視する頻度などについて説明します。

3.1 監視方法

監視方法は、「死活監視、状態監視」、「メッセージ監視」、「容量監視」、「性能監視」の4つに分類されます。それぞれの監視方法について説明します。

死活監視、状態監視

サーバーからの応答があるか、PostgreSQL のプロセスが正常に稼働しているか、SQL の実行が可能かを監視します。稼働状態を監視し、早期に障害検出と対処を行うことで、システムの停止を最小限にすることができます。サーバーの死活監視は、OS の ping コマンドを利用します。PostgreSQL の死活監視は、OS の ps コマンドや PostgreSQL の pg_isready コマンドを利用します。SQL の実行可否は、「SELECT 1」などのシンプルなクエリを利用します。

また、CPU やメモリーなど、OS のリソースの状態についても監視します。リソースの状態を監視することで、障害発生時の早期解決や、異常の兆候をより早く検出できます。OS のリソースの状態監視は、OS の sar コマンドや vmstat コマンドなどを利用します。

メッセージ監視

PostgreSQL は、インフォメーションやエラーメッセージをサーバーログに出力しています。システム内で行われた処理やハードウェアからのメッセージは syslog に出力しています。これらのログを監視することで、エラーを検出することができます。また、エラーが発生した場合は、エラーメッセージからエラー原因の分析を行い、対処することができます。

メッセージには、メッセージレベル (DEBUG[1~5]、INFO、NOTICE、WARNING、ERROR、LOG、FATAL、PANIC) が付加されています。PostgreSQL でエラーが発生した場合は、「ERROR」「FATAL」「PANIC」のいずれかが割り当てられるため、この3つを監視することで、エラーの発生を早期に検出し対処することができます。

なお、PostgreSQL のインフォメーションやエラーメッセージを syslog に出力すると、「ERROR」は「WARNING」に、「FATAL」は「ERR」に、「PANIC」は「CRIT」に変換されます。そのため、「WARNING」「ERR」「CRIT」を監視します。

容量監視

データベースクラスタ領域 (PGDATA)、TABLESPACE 領域、アーカイブ WAL (Write-Ahead Log) 領域のディスク容量が足りなくなると、データベースが更新できなくなったり、PostgreSQL が強制終了したりと運用に大きな支障をきたします。ディスクの使用率や空き容量を監視することで、領域不足の予兆を検知することができます。これにより、容量不足を起こす前にディスクを増設するといった対処をすることができます。

容量の監視は、OS の ls コマンドや du コマンド、PostgreSQL の pg_database_size 関数を利用します。また、OSS の pg_statsinfo などを利用することもできます。

性能監視

遅い SQL はないか、ロック待ち時間の長い SQL はないか、などを監視します。キャッシュヒット率やインデックスの使用率、OS のリソースの状態と併せて監視することで、チューニングポイントを判断することができ、問題に素早く対応できるようになります。チューニングポイントは、システムチューニング (postgresql.conf といった設定ファイル、ハードウェアリソースなど) と SQL チューニング (テーブル設計、SQL 設計など) に分類できます。また、不要領域やインデックスのばらつきを監視し、対応することで、性能の向上が期待できます。

性能の監視は、PostgreSQL の統計情報ビューやサーバーログを利用します。また、OSS の pg_statsinfo や pgBadger などを利用することもできます。

3.2 監視間隔、異常とみなす値（閾値）

「2. データベースシステムの監視対象」と「3.1 監視方法」で監視対象と監視方法を決めたあとは、その監視対象を「どのくらいの頻度」で監視するのか決めます。また、監視した結果「どのレベルから異常とみなすのか」を決めます。ここでは、「監視間隔」と「異常とみなす値（閾値）」について説明します。

監視間隔

監視間隔は、SLA（Service Level Agreement）などのシステム要件や、監視による負荷を考慮して決定します。例えば、監視を頻繁に行えば異常を早期に検知することはできますが、取得する情報量が多くなるためディスク容量が圧迫することになります。また、通信が必要な監視の場合は、ネットワーク帯域に影響を与えます。そのため、これらを考慮し、監視する内容・目的に応じた監視間隔を設定してください。

監視方法	監視間隔
死活監視、状態監視	障害発生時に早期にスタンバイサーバーに切り替え、システムの停止時間を最小限にするため、「秒間隔」での監視を行います。
メッセージ監視	問題発生時の情報だけでなく、ある程度の期間の情報が必要になるため、「分間隔」での監視を行います。問題が発生した原因の追求や、問題の切り分けができるように、多くの情報を収集・保存するため、秒間隔の監視ではディスク容量を圧迫する可能性があります。
容量監視	
性能監視	

異常とみなす値（閾値）

「異常とみなす値（閾値）」の判断はシステムによって異なるため、安定稼働時の状態、今後のシステムの変動、異常時の対応時間を考えたうえで、問題と判断すべき値（閾値）を決めてください。例えば、ディスクの空き容量が1%になった時点で異常を検知しても、ディスクを準備している間にシステムが停止してしまうかもしれません。「ディスクの空き容量が20%未満になったらアラームを挙げる」のような余裕を持った設定にしておくことで、運用に影響を与えることなく対応することができます。このように、監視する内容・目的に応じた閾値を設定してください。

3.3 監視ツール

PostgreSQL本体にはデータベースの監視に必要な情報を収集する機能はありますが、「監視」自体をする機能はありません。監視するためには監視用のシェルスクリプトやSQLシェルスクリプトを作成するか、以下に示すような監視ツールを使用します。異常が検出された場合には、管理者に異常を知らせる仕組みも併せて構築してください。

OSSの監視ツール	概要	通知機能
pg_statsinfo	統計情報やosのリソース情報などを時系列で取得・保存します。統計情報の取得機能、簡易レポート出力機能、アラート機能などを備えています。	有り
pgBadger	ログファイル解析 / 統計レポートを出力します。	無し
Zabbix	サーバー、ネットワーク、アプリケーションを集中監視します。統合監視に必要な、監視、障害検知、通知機能を備えています。	有り
pg_monz (テンプレート)	複数のコンピューターを単一のコンピューターのイメージで運用管理します。状態監視、ジョブ管理、性能管理、環境構築、収集蓄積を行う機能を備えています。	有り
Hinemos		

PostgreSQL のデータベースシステムの監視対象と監視方法の概要について説明しました。データベースシステムを監視することで、異常やその兆候をより早く検出し、対応することができます。これにより、業務システムの停止を防ぎ、可用性の高いシステムを構築することができます。

2019 年 9 月 27 日