

DBバックアップあーんどリカバリ



HN おいろん



1. はじめに

1. はじめに

- 自己紹介

- 名前: 守田 典男 (HN:おいろん) 29歳
- 職業: 某会社 技術社員

- 業界歴: 開発(汎用):2年→DB:4年→
開発・DB:2年
- DB歴: Oracle 6年
SQLServer2000、2005 1年半
HiRDB 半年

1. はじめに

- 本セッションについて
 - 対象：DBA初心者（Lv1クマー）

あくまで「**企業システムのデータベース**」を対象としておりますが、
バックアップをとることの重要性や
環境を復旧させることについては
どなたでも参考になるかと…。

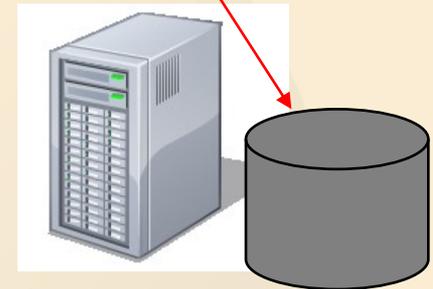
- バックアップとれば、安心！と思ってる方！！
復旧できなければ意味がありません！！

1. はじめに

- 対象範囲



アプリケーション



データベース

これ

システムの中でもデータを管理する部分が
障害で壊れてしまった場合、どのように復旧すれば
よいか？
復旧計画はどのように立てていけばよいか？

目次

1. はじめに
2. バックアップの考え方
3. リカバリの仕組み
4. おわりに

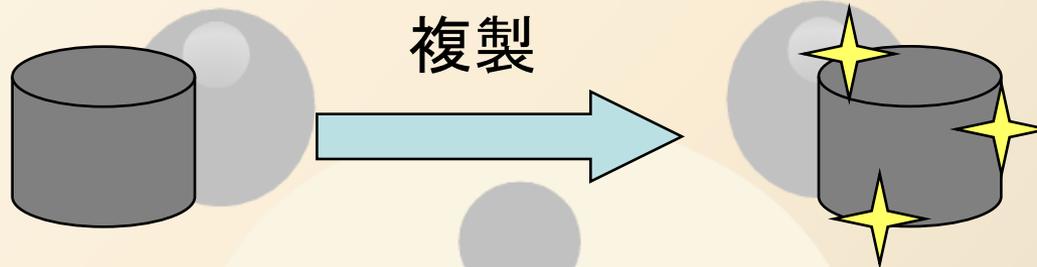
2. バックアップの考え方

2. バックアップの考え方

- 用語解説

- バックアップ

- ある時点のデータの複製

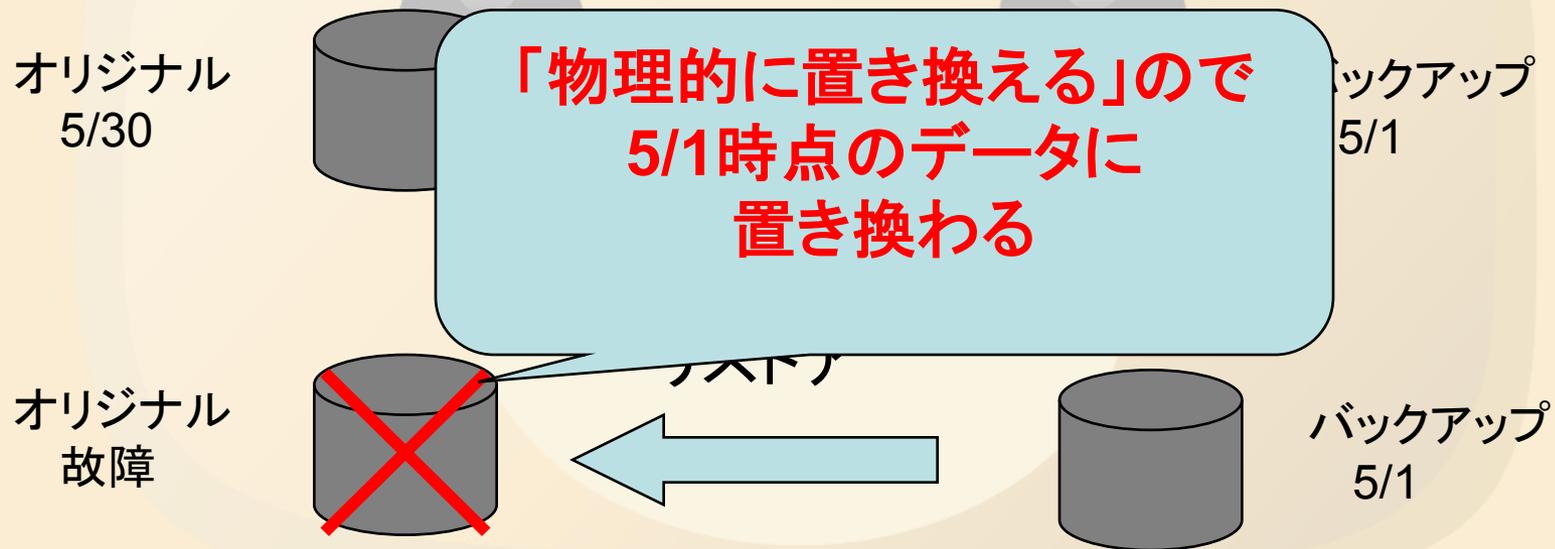


- ハードの故障や紛失など、「万が一」に対するデータの保護を目的とする。
- 使われなくなった過去のデータなど、保存目的で取得する場合もある

2. バックアップの考え方

- 用語解説
- リストア

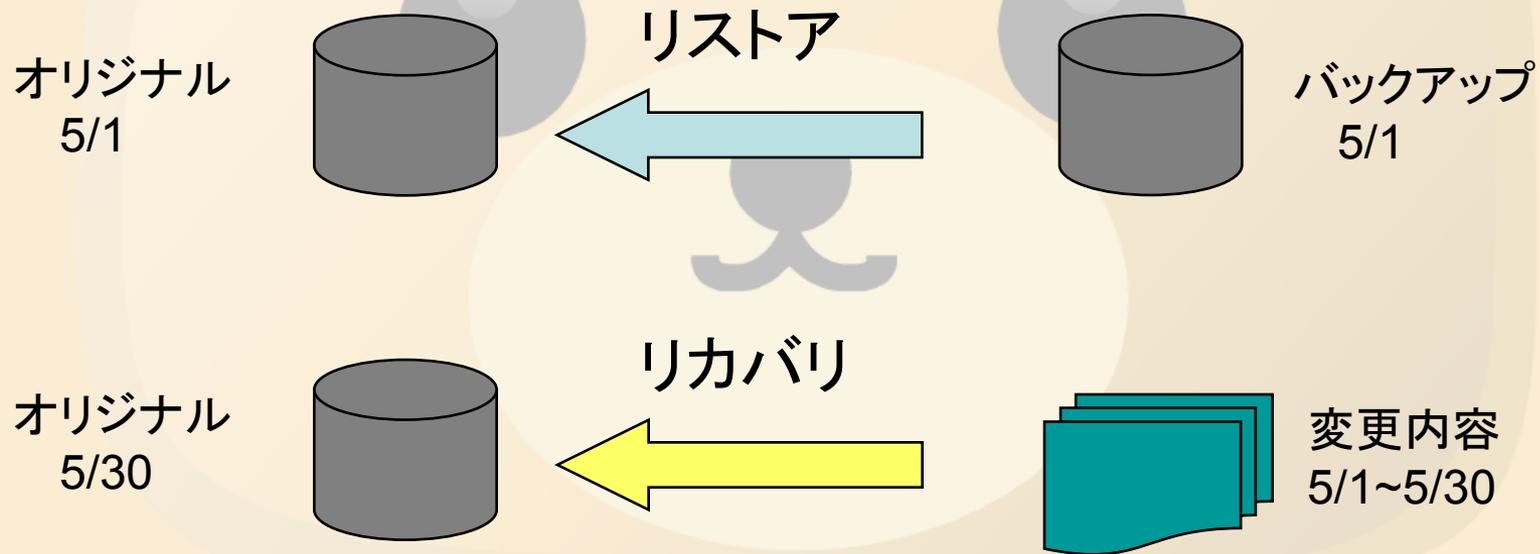
- 取得しているバックアップから、データを物理的に復元すること



2. バックアップの考え方

- 用語解説
- リカバリ

- リストアしたデータに対して、その後の変更内容を反映させ、障害発生直前まで復旧すること



2. バックアップの考え方

• バックアップ計画の立て方

– 検討すべきポイント

- 障害からの復旧にどこまで時間をかけられるか？
- どの時点のデータに復旧すればよいか？

– バックアップ要件の確認

- バックアップ対象とサイズ
- バックアップ取得時間帯
- バックアップ取得方法
- バックアップ世代数と取得媒体

2. バックアップの考え方

- **バックアップ対象とサイズ**

- **全体バックアップ**

- データベース全体のバックアップを取得する。
 - バックアップ取得時間およびサイズが最大となる。

- **差分バックアップ**

- ある時点（全体バックアップ取得時点）から、現在まで更新された差分のバックアップを取得する。
 - 全体バックアップにくらべ取得サイズが小さくなり、応じて取得時間も少なくてすむ。

2. バックアップの考え方

- **バックアップ取得時間帯**

- **サービス停止期間**

- 深夜の時間帯など、システムが停止できる場合は停止時間帯を選択する。
停止できる時間によって、取得するバックアップも変わってくる。

- **システム使用率がもっとも小さい時間帯**

- バックアップを取得すると当然パフォーマンスが落ちるので、パフォーマンスが落ちててもシステムに影響がでにくい時間帯を選択する。
(深夜、休日など)

2. バックアップの考え方

- **バックアップ取得方法**

- **コールドバックアップ(オフラインバックアップ)**

- データベースを停止し、OSコマンド等によるバックアップ。
 - 静的な状態でのバックアップなので、リストアした後リカバリは不要。

- **ホットバックアップ(オンラインバックアップ)**

- 24時間365日システム稼動必須の場合はこちら。データベースを停止することなく、バックアップを取得する

2. バックアップの考え方

• バックアップ世代数と取得媒体

– 世代数

- 世代数が多ければ多いほど、過去の状態に戻ることができる。
- 当然のことながら、その分のバックアップファイルが必要なので、リソースもあわせて必要となる。

– 取得媒体

- 磁気テープ、バックアップサーバなど



2. バックアップの考え方

• バックアップ計画例



- 対象とサイズ
 - 平日は差分バックアップ、土日に全体バックアップ
- 取得時間帯
 - 平日、土日ともに深夜1:00～6:00の間
- バックアップ取得方法
 - 平日はホットバックアップ、土日はコールドバックアップ
- バックアップ世代数と取得媒体
 - 3世代まで管理、磁気テープに保存

3. リカバリの仕組み

3. リカバリの仕組み

- 完全リカバリ

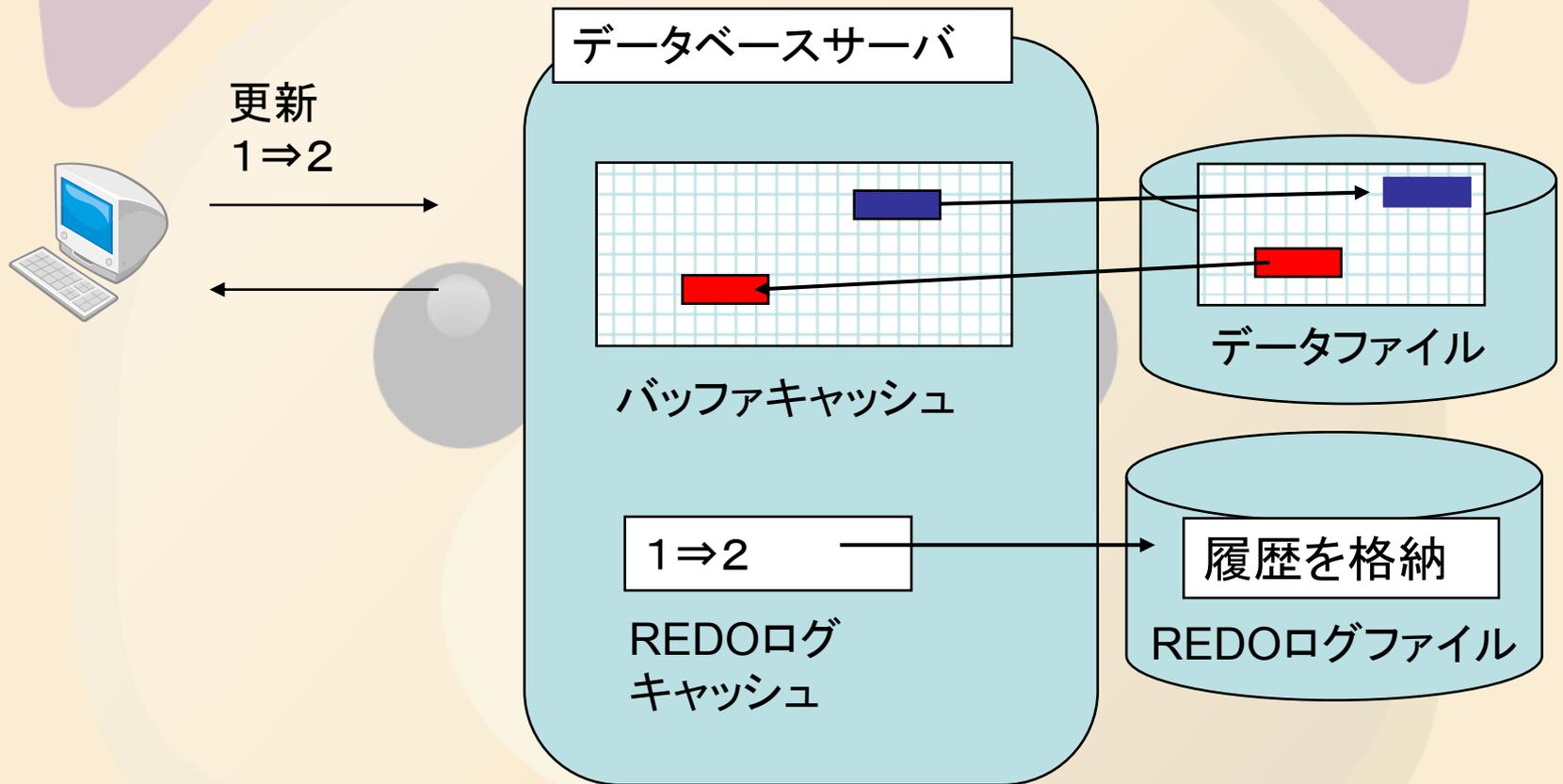
- 最新の状態に完全に復旧させること。
- **最新の状態に戻すための必要なファイルがすべてそろっている必要がある。**

- 不完全リカバリ

- 最新の更新情報がなく、ある一時点まで復旧させること。
- 利用者が誤って表を削除した場合など、意図的にある地点に戻す場合もある。

3. リカバリの仕組み

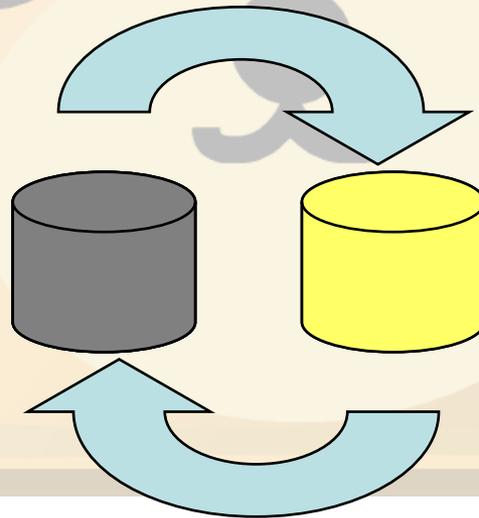
- データ更新時の動き



3. リカバリの仕組み

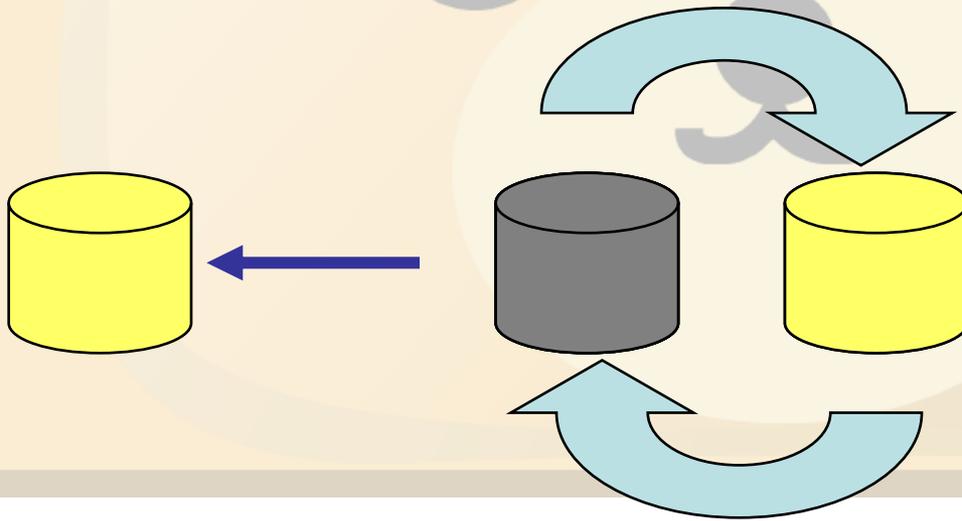
- REDOログファイル

- 更新の内容を記録していく
- 複数のファイルで構成され循環して使用される。
- ファイルがいっぱいになると次のファイルに切り替わり、すべてがいっぱいになると上書きされる。



3. リカバリの仕組み

- アーカイブログファイル
 - REDOログファイルを上書きする直前にアーカイブしたもの。
 - 「アーカイブログモード」に設定する必要がある

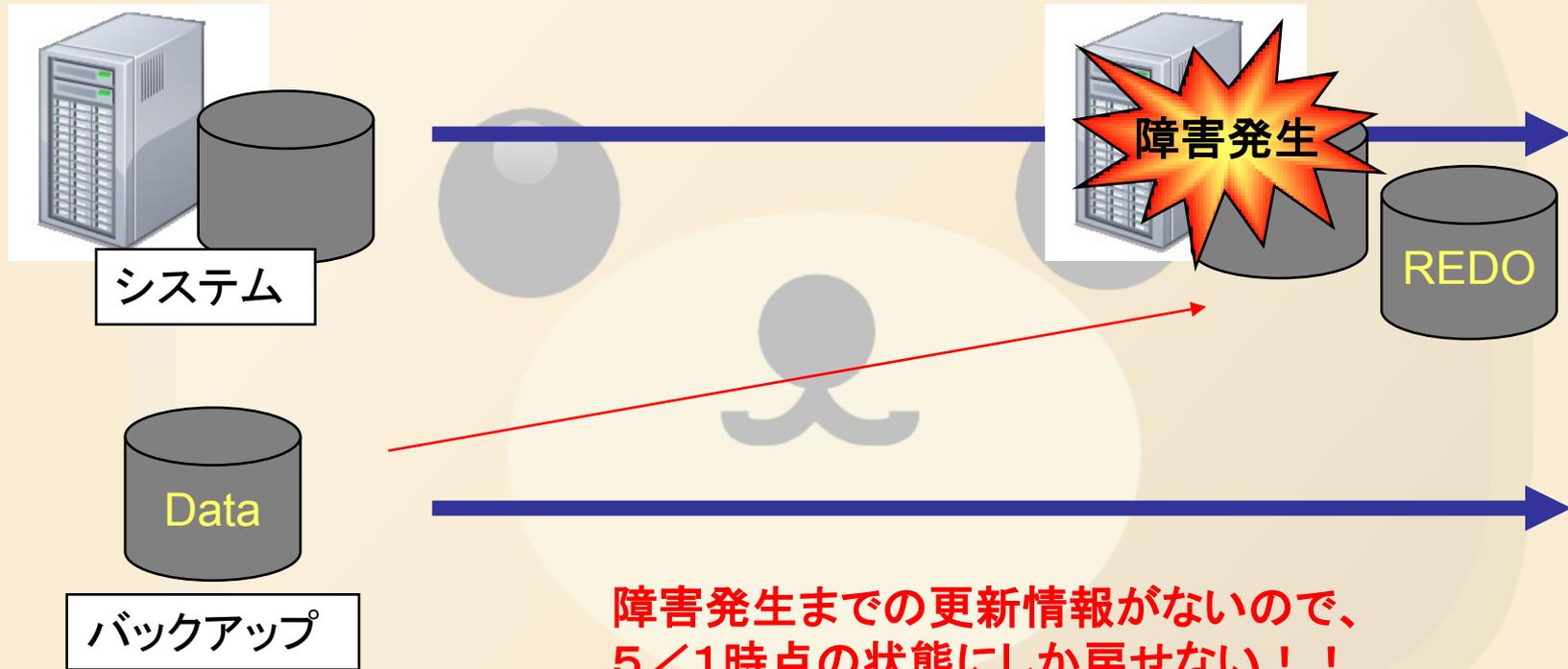


3. リカバリの仕組み

- ノーアーカイブログモードの場合

5/1

5/30

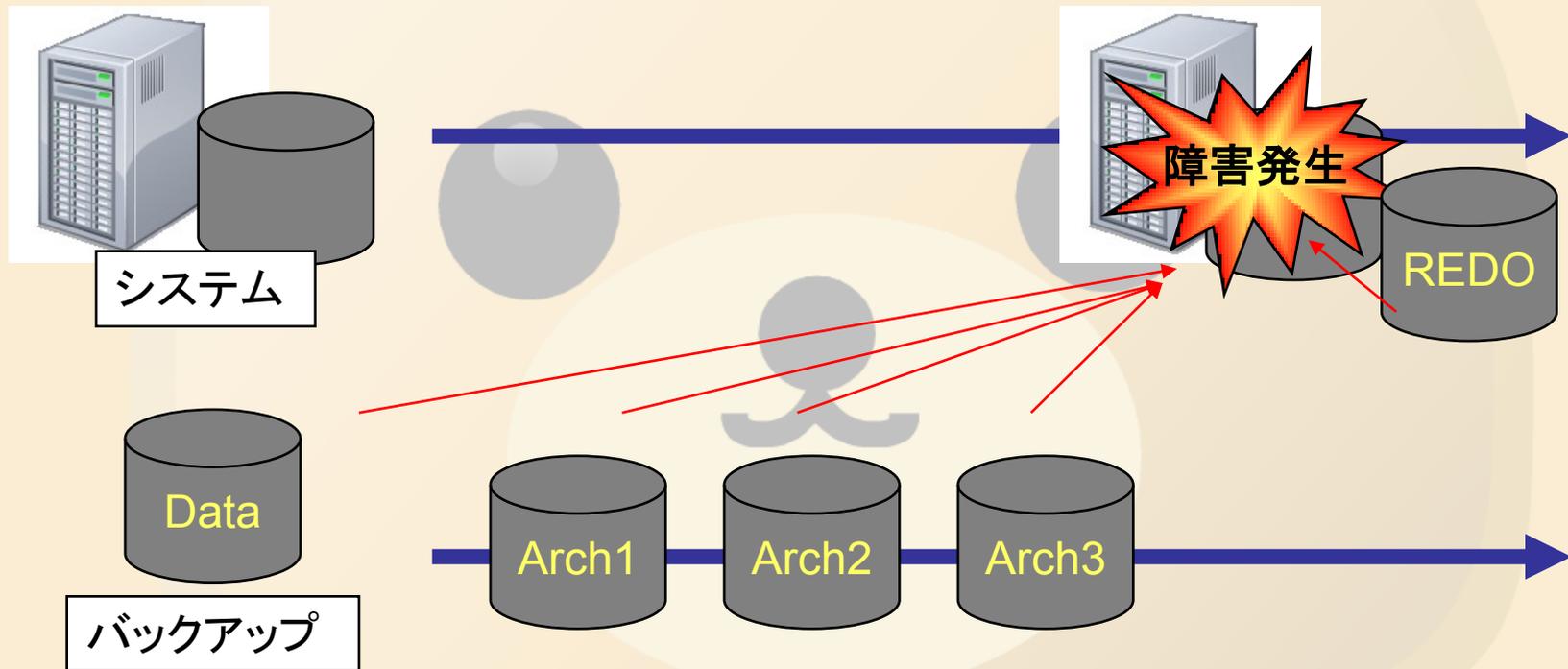


3. リカバリの仕組み

- アーカイブログモードの場合

5/1

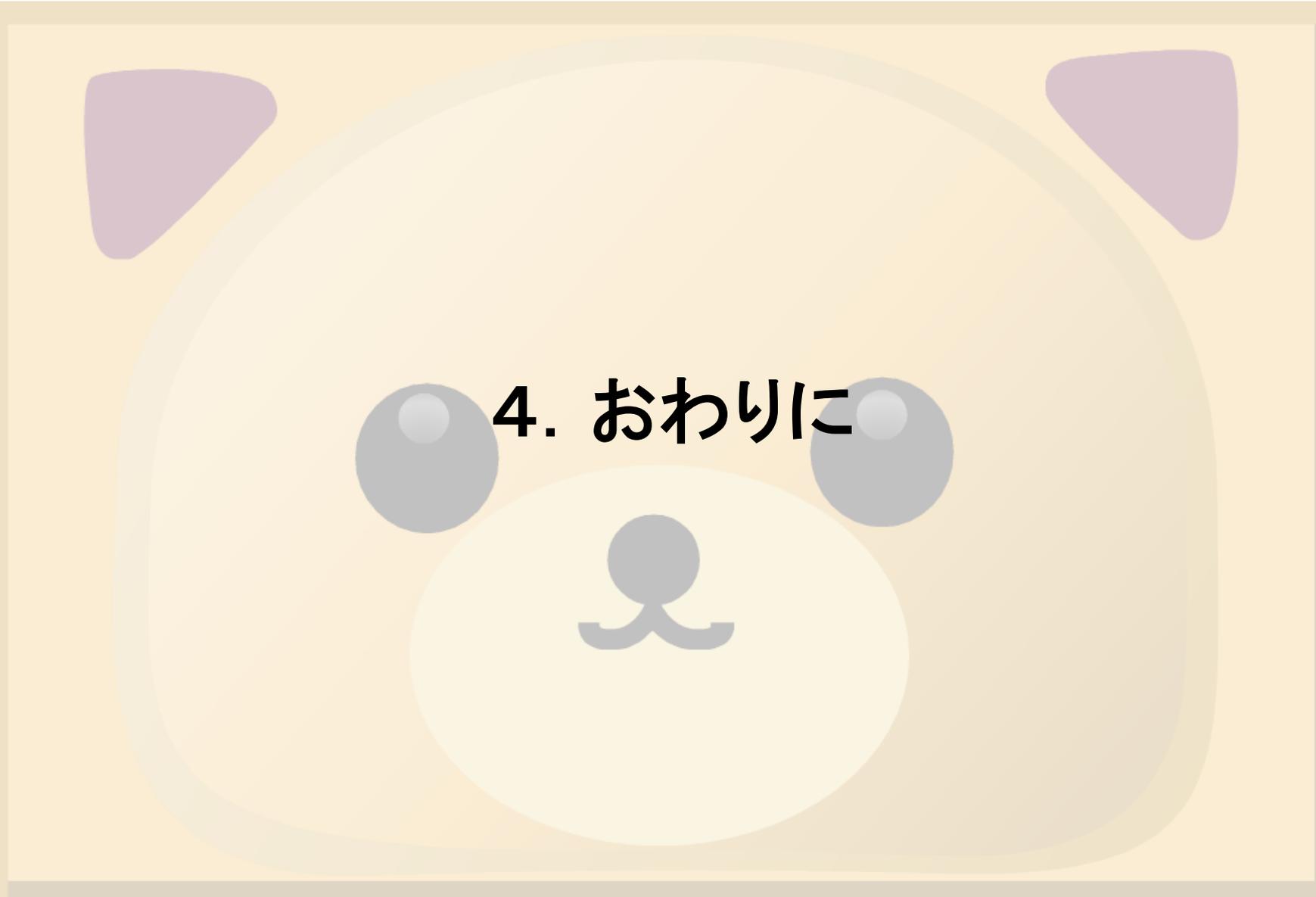
5/30



障害発生直前まで復旧が可能

3. リカバリの仕組み

- データベース運用モード
 - アーカイブログモード
 - データベースの停止が許されない場合
 - データを失うことが許されない場合
 - 過去のある地点への復旧が必要な場合
 - ノーアーカイブログモード
 - データを失っても構わない場合
 - アーカイブ適用以外で復旧が可能な場合



4. おわりに

4. おわりに

- 参考サイト

- @IT Database Expert

- <http://www.atmarkit.co.jp/fdb/>



4. おわりに

- いかがでしたか？
 - 少しでも、DBを意識したり、興味を感じる事ができたでしょうか？
 - 「バックアップ取得は当たり前」といえど、要件を熟知して、要件にあった計画をたてる事が大切となります。
 - 最悪の状況を回避できる、顧客に喜ばれるシステムを開発しましょう！！

ご清聴ありがとうございました！！