Zabbixサポート担当者が選ぶ、お客様が遭遇した謎事象と技術解説

NEC　野村　昌平

アジェンダ

Zabbixのサポート対応状況

問い合わせ事例のご紹介

　事例①：恐怖！ログ監視の結果がロスト！

　事例②：発生しないはずの「Low Memory Mode」が起きた！

まとめ

お問合せ先

Zabbixのサポート対応状況

■OSSミドルウェアサポートサービス

- システム導入後の運用支援として、OSSに対応したサポートサービスをご提供

サポート提供OSSラインナップ

カテゴリ	OSSミドルウェア名
Red Hat Application Services	Red Hat Integration(AMQ, Fuse, Camel, 3scaleが対象) Red Hat Runtimes(JBoss EAP, Data Grid, Keycloakが対象) Red Hat Application Foundations (Red Hat Integration, Red Hat Runtimesが対象)
Web/APサーバ	Apache HTTP Server, Apache Tomcat
運用管理	Zabbix, MIRACLE ZBX, Red Hat Ansible Automation Platform, Ansible
ファイルディレクトリ	Samba, OpenLDAP
メール関連	sendmail, Dovecot, Postfix
DNS	BIND
データベース	PostgreSQL
分散処理	Hadoop
その他	Red Hat build of OpenJDK, Spring Framework/Spring Boot

※一部Red Hat製品は、製品本体とセットでサポートを提供

サポート提供内容

問い合わせ対応
対象OSSに関するマニュアルレベルの技術的問い合わせにお答えします。

修正物件の問い合わせ
コミュニティから提供されている修正物件に関する問い合わせにお答えします。

発生現象の確認・調査
発生現象の確認と、同様の障害事例が過去に発生しているか調査を行います。

メッセージの調査
対象OSSが出力する各種ログを調査します。

環境再現による評価
必要に応じて、お客様環境にて発生した事象の再現確認を実施します。

コミュニティへのフィードバック
新規バグ判明時にコミュニティに対する障害報告と措置への働きかけを行います。

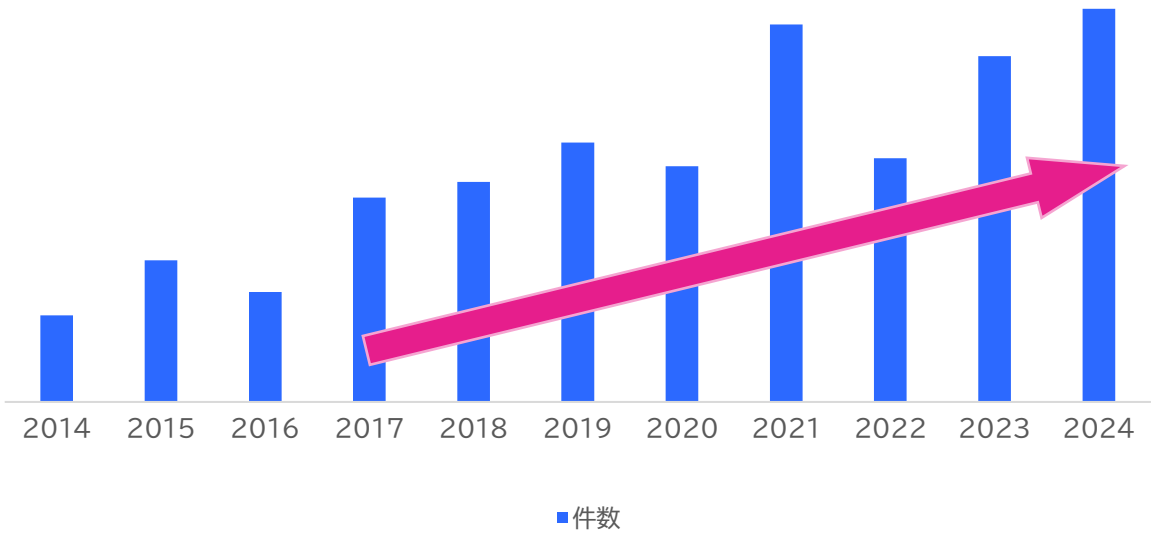
※ データの保証・復旧、パッチ作成、パフォーマンス分析・チューニングは対象外
※ プロジェクトの状況やOSSによって対応可能な範囲が異なる場合あり

Zabbixのサポート対応状況

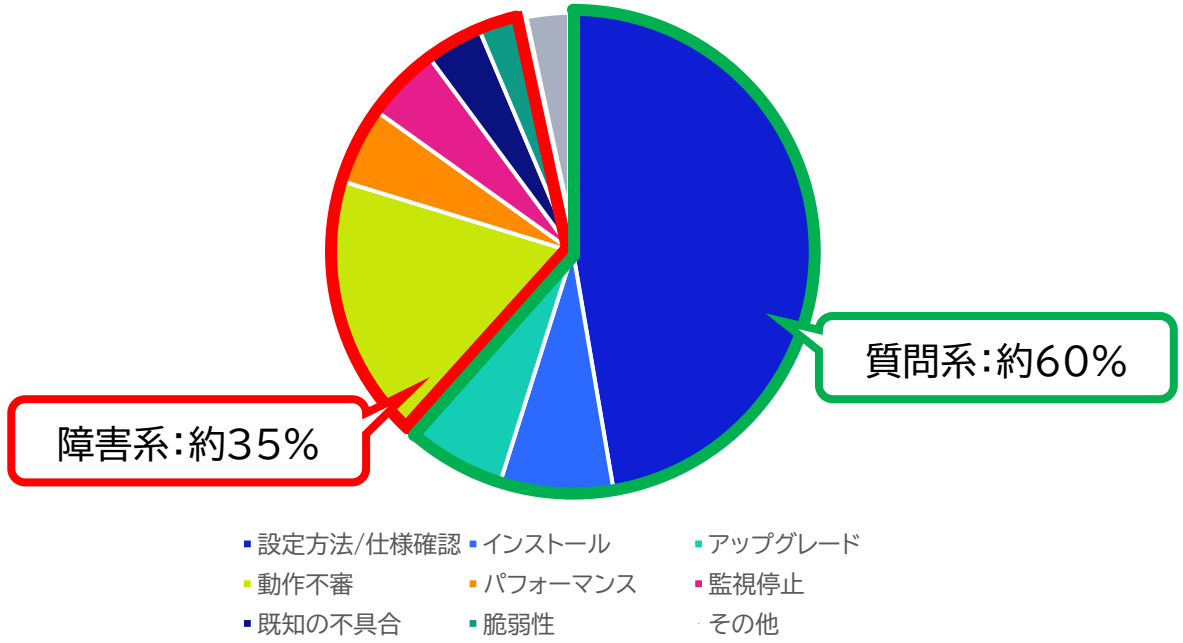
■対応状況

- 近年10年の問い合わせ件数は**右上がり成長**
- 2024年度の問い合わせ傾向として、**質問系が約60%・障害系が約35%**
  - **設定方法・仕様確認**の質問や、異常検知等の**動作不審**に関する障害解析依頼が多い

年間対応件数



分類毎の問い合わせ件数(2024年度)



問い合わせ事例のご紹介

■お客様からお問合せ頂いた事例の中から、以下のキャッシュに関する事例をご紹介

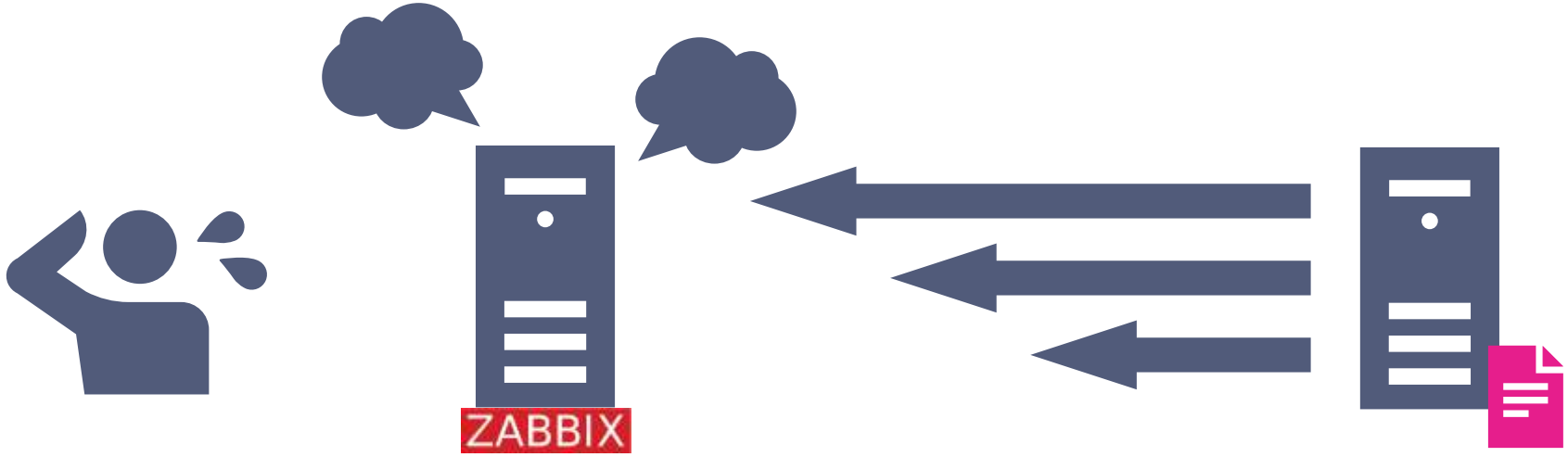
■事例①

恐怖！ログ監視の結果がロスト！

■事例②

発生しないはずの「Low Memory Mode」が起きた！

事例①：恐怖！ログ監視の結果がロスト！



ログ監視実行時に検知対象メッセージが大量発生したが、
一部メッセージがZabbixサーバのヒストリに記録されず、ロストした

事例①:恐怖！ログ監視の結果がロスト！

■何が起きたか？

- HistoryCacheが枯渇したため、監視結果をDBへ書き込みできなかった

■HistoryCacheとは？

- 監視結果をDBへ書き込む際、監視結果を一時的にメモリ上にバッファリングする領域
  - データベースアクセスの負荷低減や障害発生時のデータロスト対策などを図っている

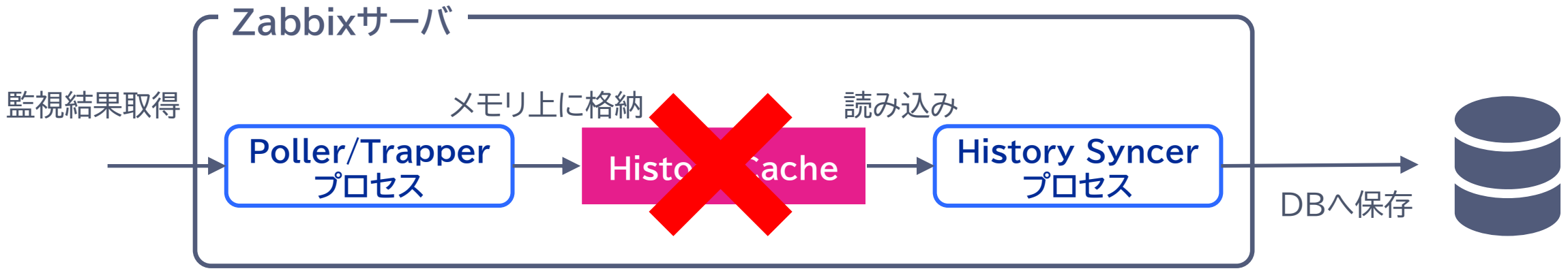
事例①:恐怖！ログ監視の結果がロスト！

■何が起きたか？

- **HistoryCache**が枯渇したため、監視結果をDBへ書き込みできなかった

■HistoryCacheとは？

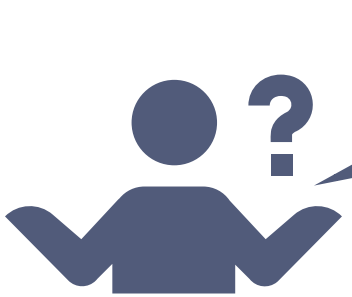
- 監視結果をDBへ書き込む際、監視結果を一時的にメモリ上にバッファリングする領域
  - データベースアクセスの負荷低減や障害発生時のデータロスト対策などを図っている
- 枯渇すると監視結果をメモリ上に格納できなくなり、History SyncerがDBへ書き込みできなくなる
  →**監視結果がロストする**



事例①：恐怖！ログ監視の結果がロスト！

Zabbixサーバがログを受信できない場合、Zabbixエージェントの再送処理があったはず。HistoryCacheが空くまで再送してくれてロストしないのでは？

事例①：恐怖！ログ監視の結果がロスト！

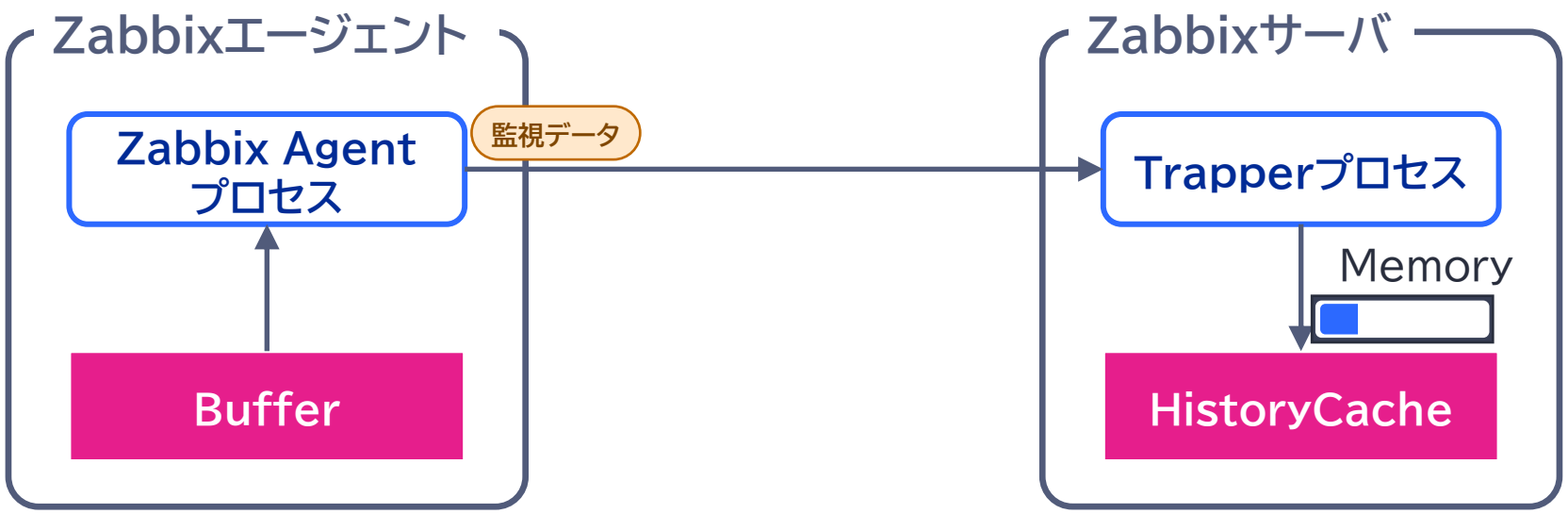


Zabbixサーバがログを受信できない場合、Zabbixエージェントの再送処理があったはず。HistoryCacheが空くまで再送してくれてロストしないのでは？

ログデータの送信自体は成功しているため、
Zabbixエージェントの再送処理は行われない

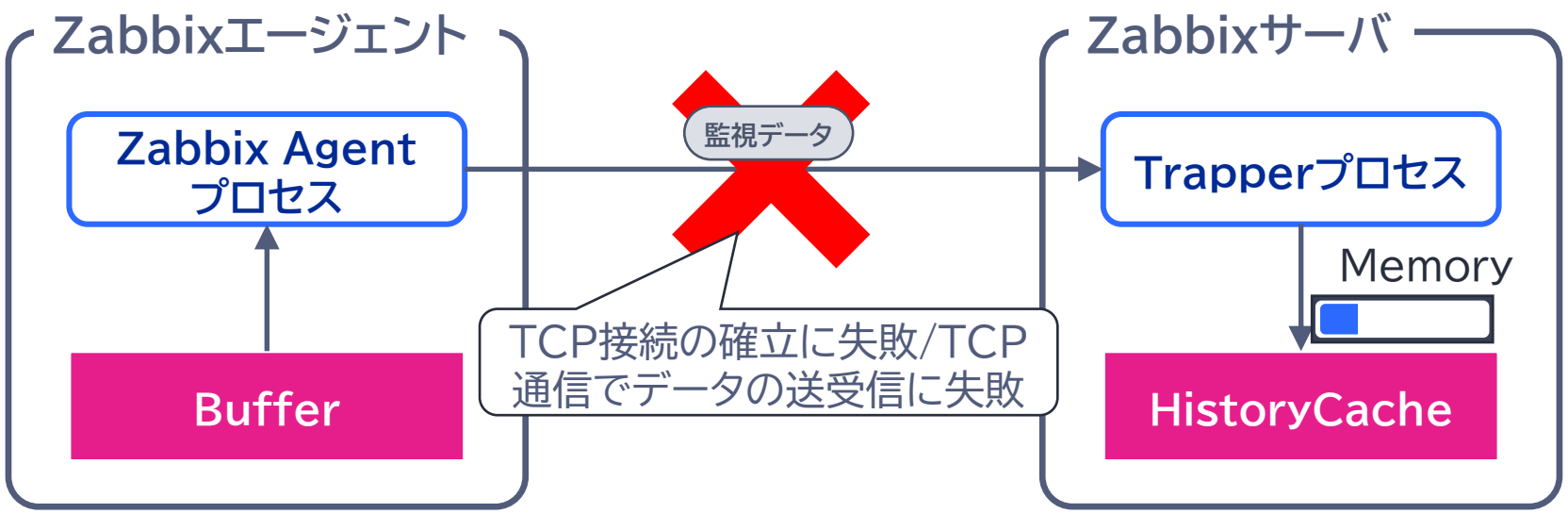
事例①:恐怖！ログ監視の結果がロスト！

Zabbixサーバが停止している等でデータ送信に失敗した場合、Zabbixエージェントは
BufferSize/BufferSendパラメータの設定値に基づいて、ログなど各種監視の結果を再送する



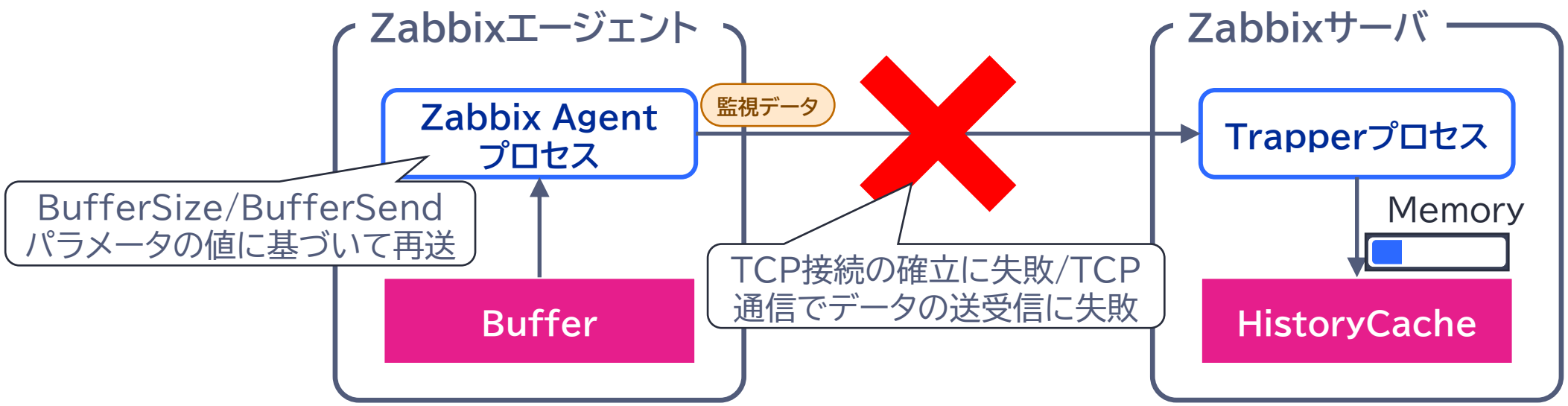
事例①:恐怖！ログ監視の結果がロスト！

Zabbixサーバが停止している等でデータ送信に失敗した場合、Zabbixエージェントは
BufferSize/BufferSendパラメータの設定値に基づいて、ログなど各種監視の結果を再送する



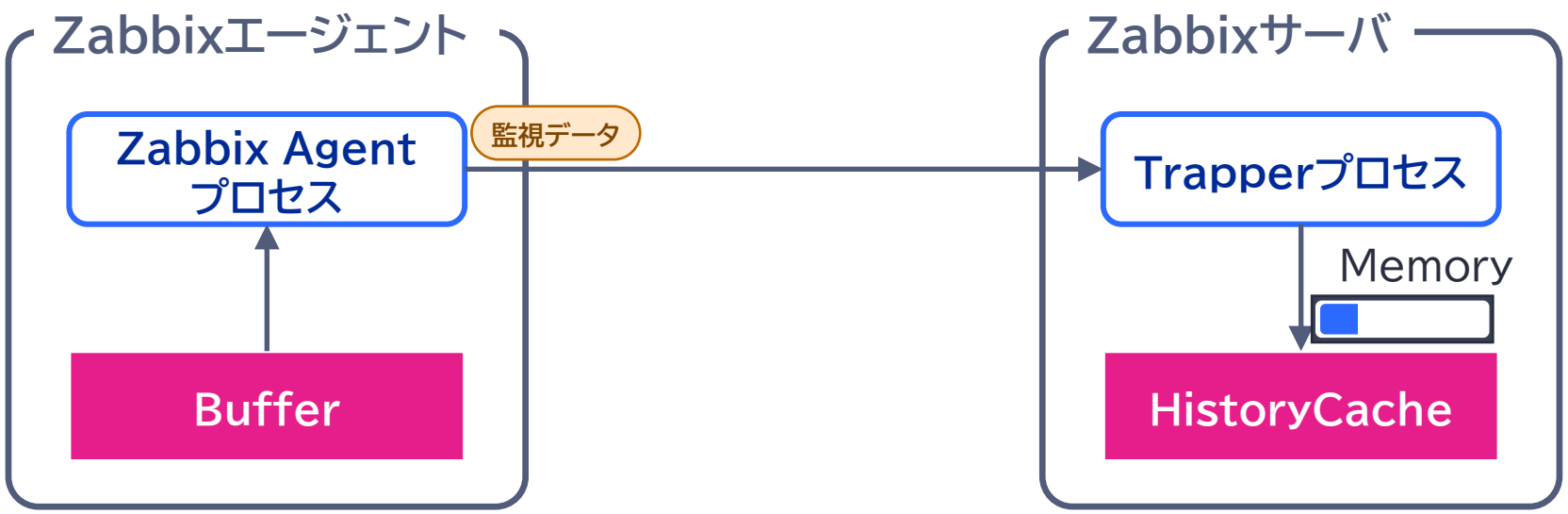
事例①:恐怖！ログ監視の結果がロスト！

Zabbixサーバが停止している等でデータ送信に失敗した場合、Zabbixエージェントは
BufferSize/BufferSendパラメータの設定値に基づいて、ログなど各種監視の結果を再送する



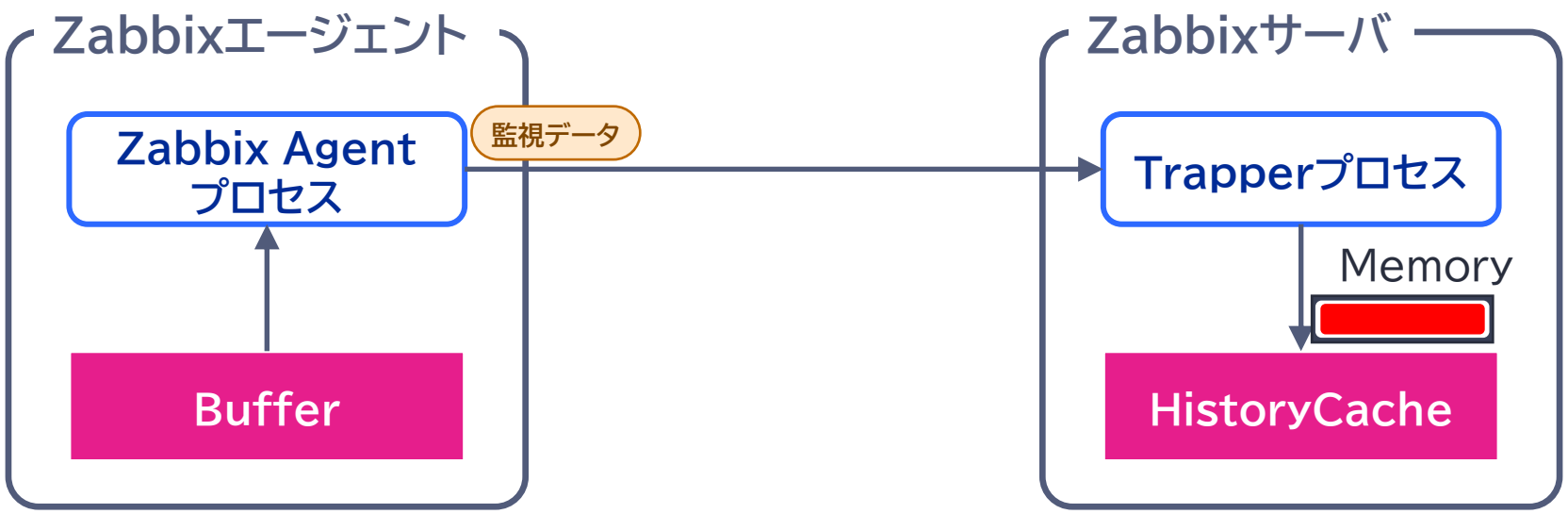
事例①:恐怖！ログ監視の結果がロスト！

HistoryCacheが枯渇してメモリ上に監視結果を展開できなかった場合、Zabbixサーバと
Zabbixエージェント間のデータ送信処理は既に成功しているため、**データの再送は行われない**



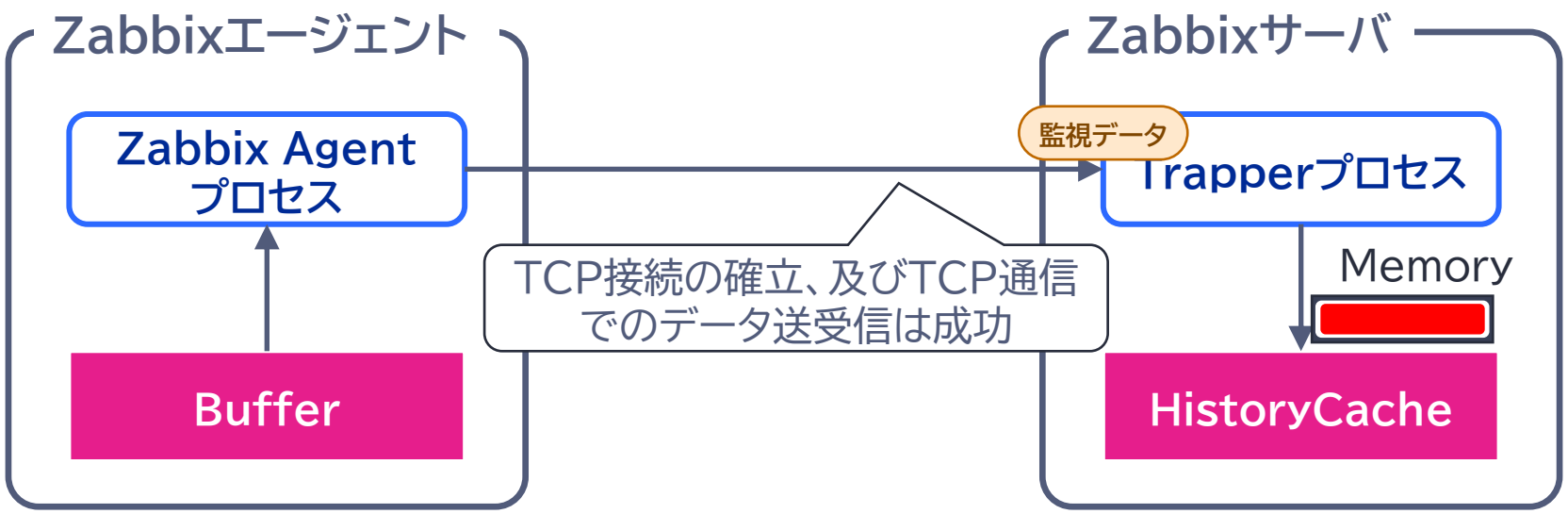
事例①:恐怖！ログ監視の結果がロスト！

HistoryCacheが枯渇してメモリ上に監視結果を展開できなかった場合、Zabbixサーバと
Zabbixエージェント間のデータ送信処理は既に成功しているため、**データの再送は行われない**



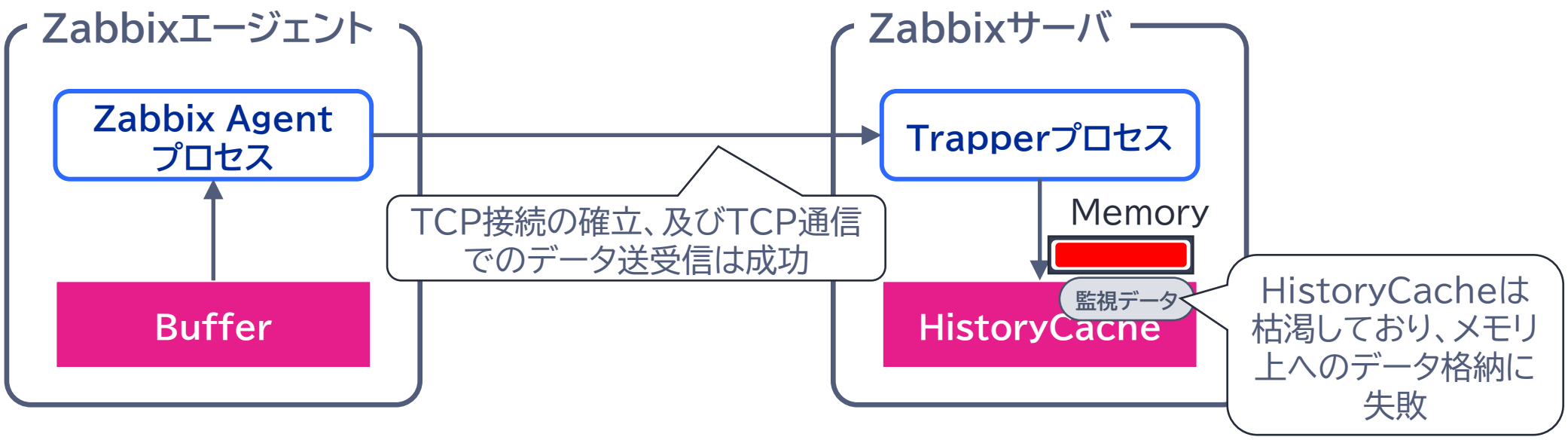
事例①:恐怖！ログ監視の結果がロスト！

HistoryCacheが枯渇してメモリ上に監視結果を展開できなかった場合、Zabbixサーバと
Zabbixエージェント間のデータ送信処理は既に成功しているため、**データの再送は行われない**



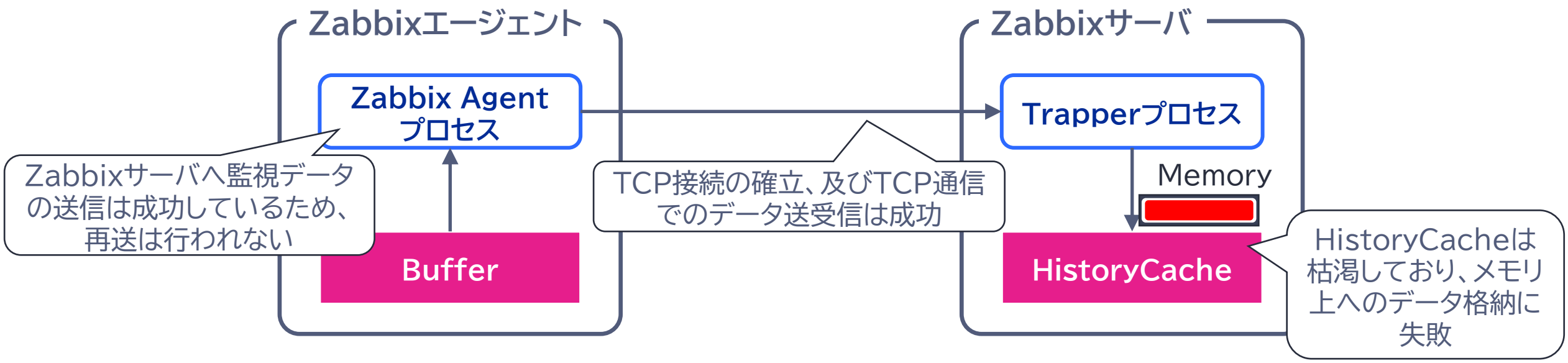
事例①:恐怖！ログ監視の結果がロスト！

HistoryCacheが枯渇してメモリ上に監視結果を展開できなかった場合、Zabbixサーバと
Zabbixエージェント間のデータ送信処理は既に成功しているため、**データの再送は行われない**



事例①:恐怖！ログ監視の結果がロスト！

HistoryCacheが枯渇してメモリ上に監視結果を展開できなかった場合、Zabbixサーバと
Zabbixエージェント間のデータ送信処理は既に成功しているため、**データの再送は行われない**



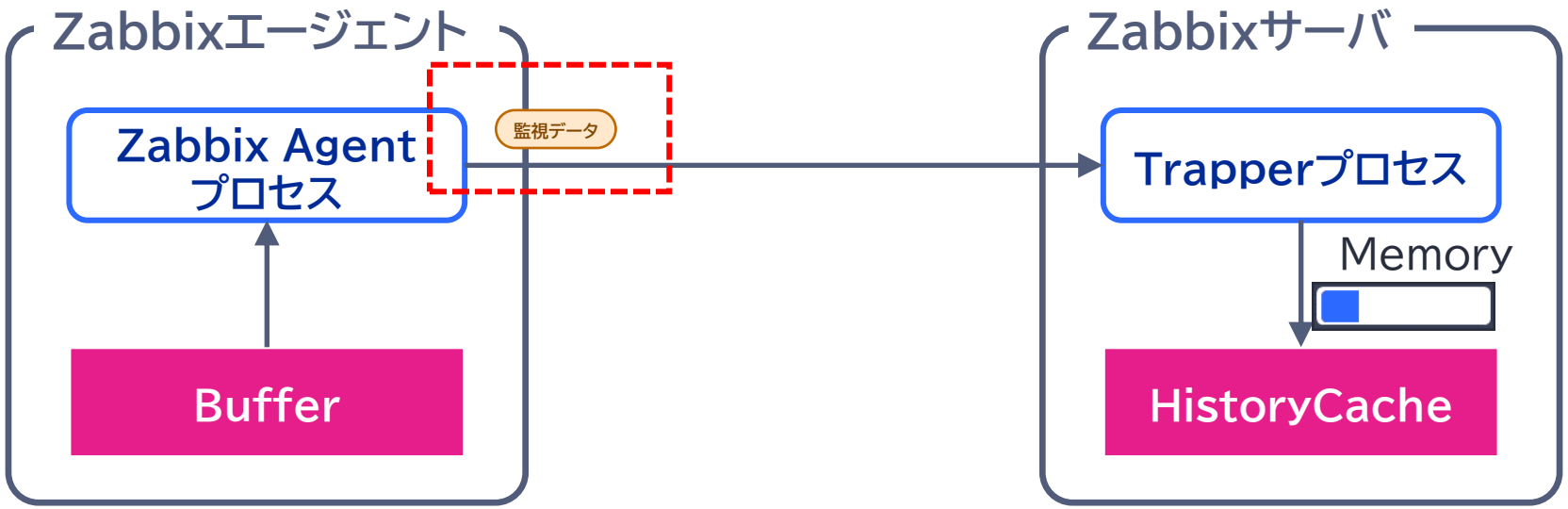
事例①:恐怖！ログ監視の結果がロスト！

■対処方法

方法①:ログ監視で取得するデータを絞り込む

ログ監視のキーlog/logrtには、取得対象のメッセージを絞り込むregexpや、秒間送信行数を設定するmaxlinesパラメータが存在する。これらを設定し取得データを絞り込み、負荷を減らす

```
logrt[file_regexp,<regexp>,<encoding>,<maxlines>,<mode>,<output>,<maxdelay>,<options>]
```

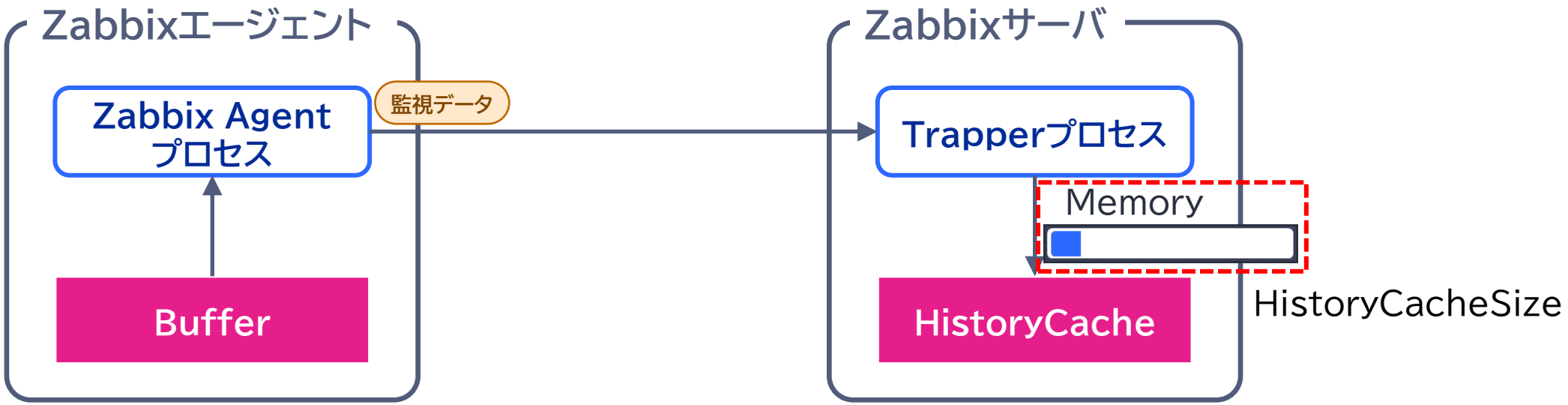


事例①:恐怖！ログ監視の結果がロスト！

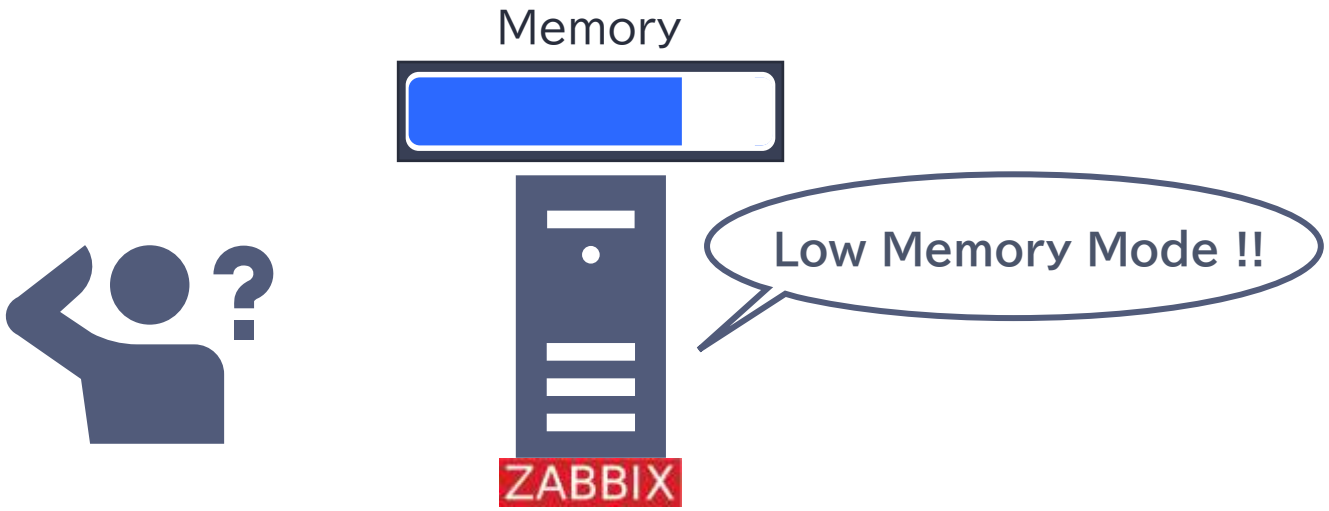
■対処方法

方法②:HistoryCacheの値を大きく設定する

Zabbix Server Healthテンプレートを適用したZabbixサーバホストを用意し、
キャッシュ使用状況を確認した上で、十分に余裕のある値を設定する



事例②:発生しないはずの「Low Memory Mode」が起きた！



Value Cacheの使用率が80%前後とまだメモリに空きがあるにも関わらず
「Zabbix value cache working in low memory mode」が検知された

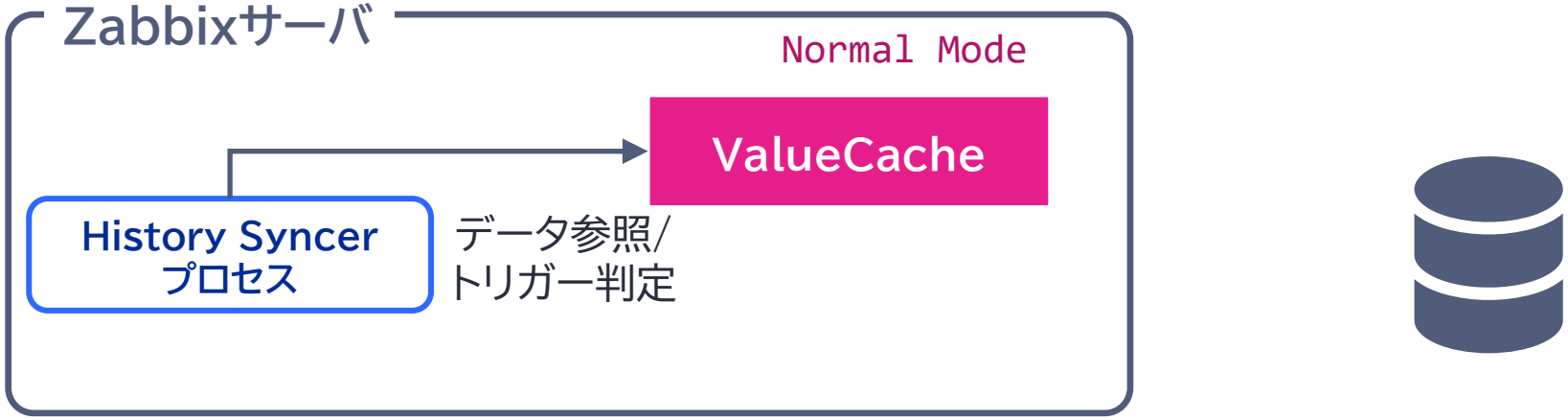
事例②:発生しないはずの「Low Memory Mode」が起きた！

■何が起きたか？

- **ValueCache**の**断片化**が進み、監視結果を格納することができるサイズの領域の確保に失敗した

■ValueCacheとは？

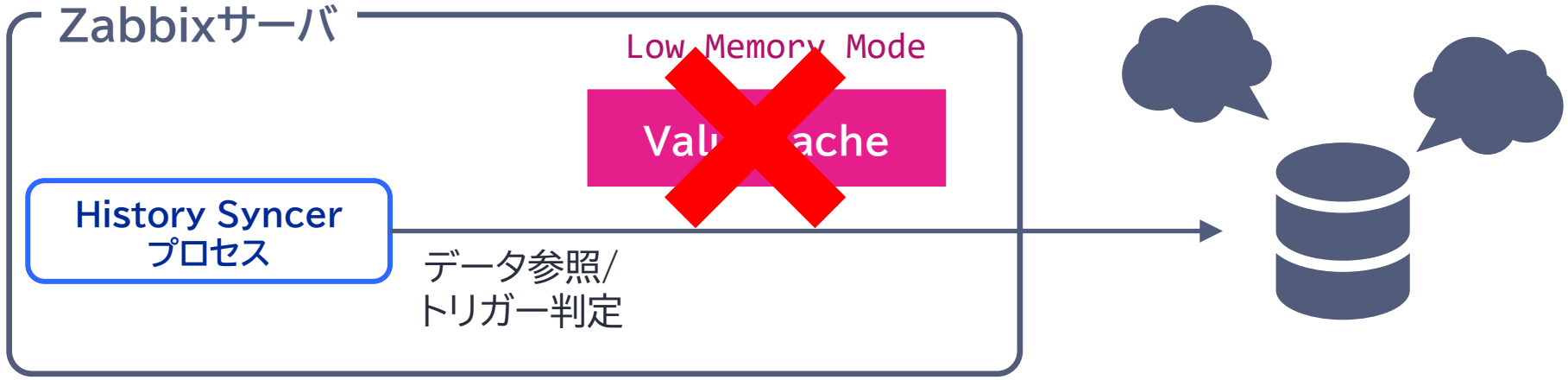
- 監視結果のうち直近のデータをメモリ上に保持する領域
- トリガー評価や計算アイテム等でメモリ上に格納されたデータを参照・処理することで、高速な判定・処理を実現



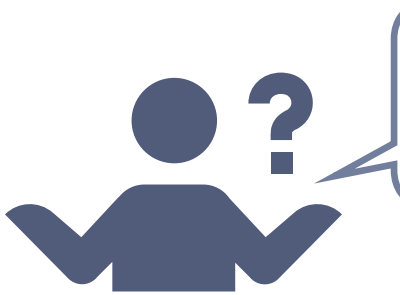
事例②:発生しないはずの「Low Memory Mode」が起きた！

■Low Memory Modeとは？

- ValueCacheが枯渇する等で監視データの格納ができなかった場合、**Low Memory Mode**と呼ばれるモードに遷移し、DBのデータを直接参照し各種処理が行われるようになる
- トリガー評価が行われる度にDBアクセスが発生するため、**Zabbixサーバ全体のパフォーマンス劣化やDBの高負荷が発生する**
- Low Memory Mode発生から1日経過で復帰を試みる。また、Zabbixサーバの再起動で復帰する

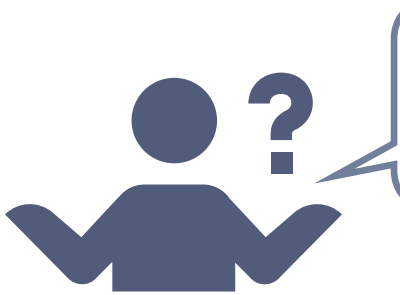


事例②:発生しないはずの「Low Memory Mode」が起きた！



あれ？ValueCacheが枯渇していないのであれば
Low Memory Modeに遷移しないのでは？

事例②：発生しないはずの「Low Memory Mode」が起きた！



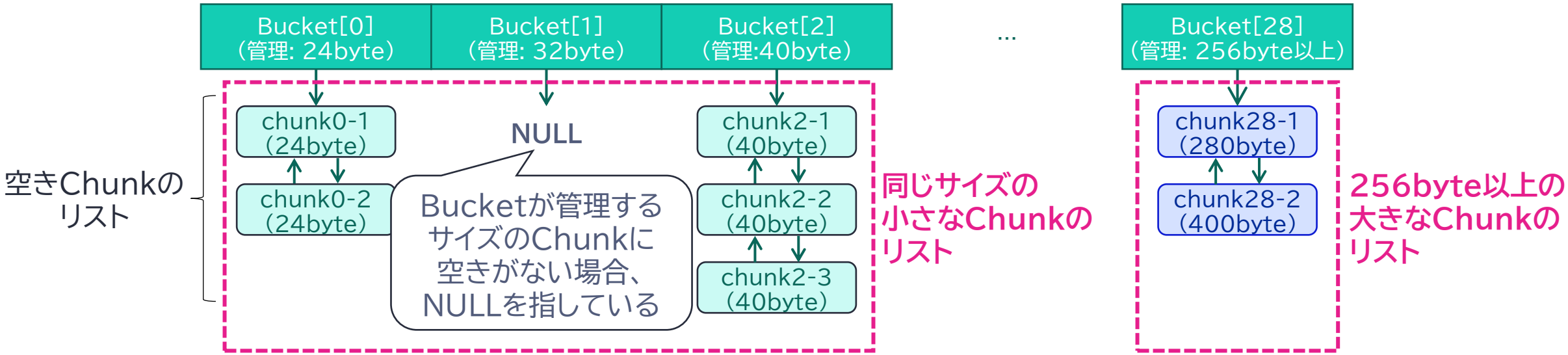
あれ？ValueCacheが枯渇していないのであれば
Low Memory Modeに遷移しないのでは？

正確には、ValueCacheが枯渇したことが契機ではなく、
監視データ格納時に必要なサイズの領域を確保できなかった場合に
Low Memory Modeへ遷移する

事例②:発生しないはずの「Low Memory Mode」が起きた！

■Zabbixのメモリ管理の仕組み

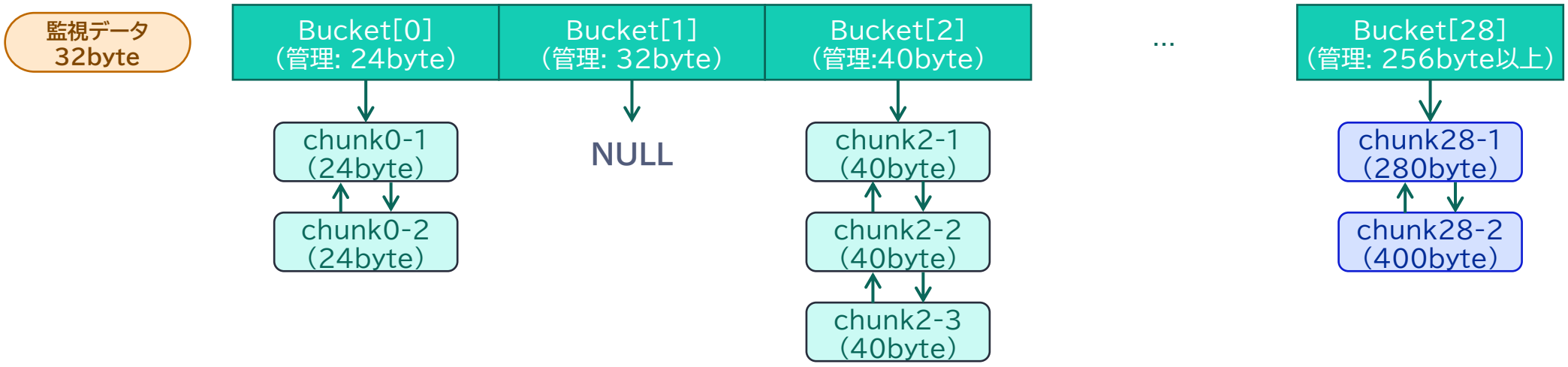
- ZabbixではChunkと呼ばれる単位でメモリを管理している
  - Chunkには、データが格納されている使用済Chunkと、データが格納されていない空きChunkが存在する
- 空きChunkはBucketと呼ばれるリスト構造で管理される
  - Bucket[0]～[27] ⇒ 24～248byte(8バイト刻み)の同じサイズが含まれる空きChunkリストを管理
  - Bucket[28] ⇒ 256byte以上の複数のサイズが含まれる空きChunkリストを管理



事例②:発生しないはずの「Low Memory Mode」が起きた！

■メモリに監視データを格納するときの動作

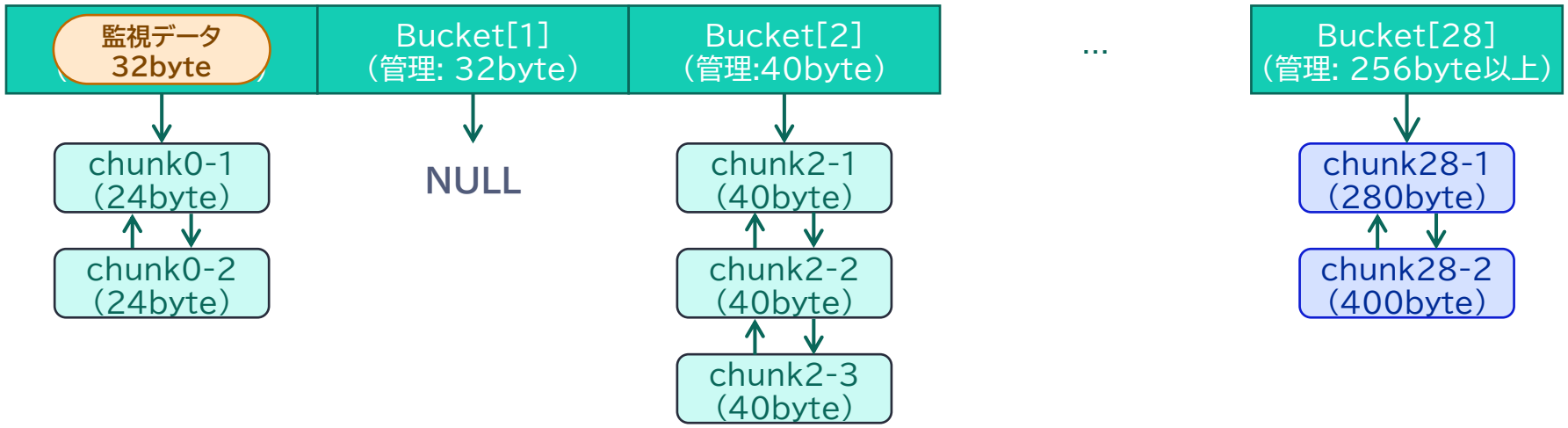
- Bucketのインデックス番号順に、格納したい監視データを格納できるサイズを持つ空きChunkを探索
- 空きChunkが見つかれば、見つかったChunkにデータを格納して空きChunkリストから削除



事例②:発生しないはずの「Low Memory Mode」が起きた！

■メモリに監視データを格納するときの動作

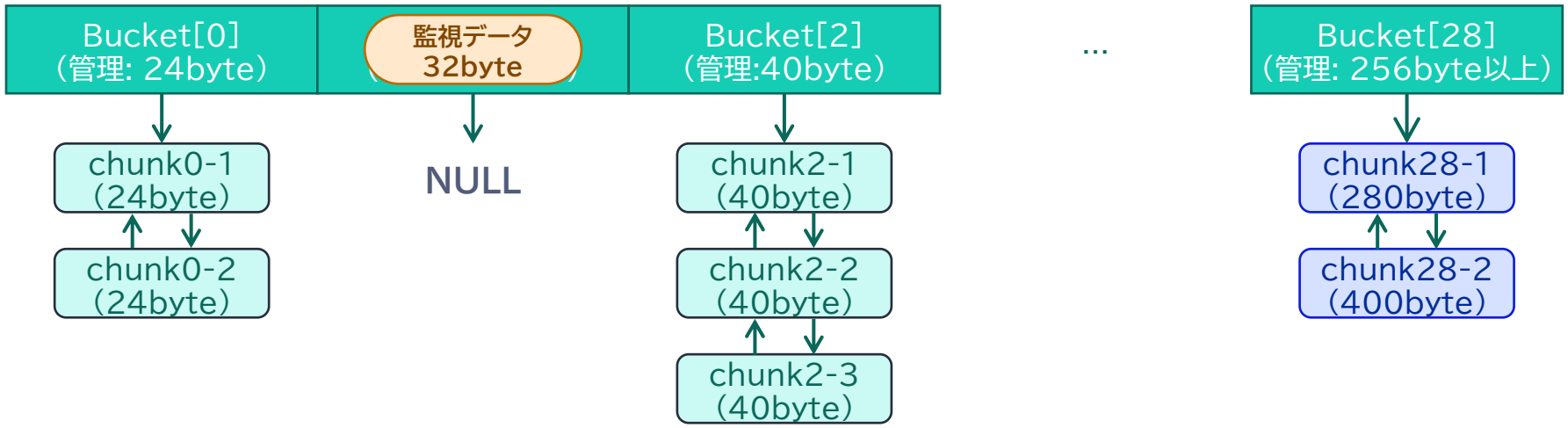
- Bucketのインデックス番号順に、格納したい監視データを格納できるサイズを持つ空きChunkを探索
- 空きChunkが見つかれば、見つかったChunkにデータを格納して空きChunkリストから削除



事例②:発生しないはずの「Low Memory Mode」が起きた！

■メモリに監視データを格納するときの動作

- Bucketのインデックス番号順に、格納したい監視データを格納できるサイズを持つ空きChunkを探索
- 空きChunkが見つかれば、見つかったChunkにデータを格納して空きChunkリストから削除



事例②:発生しないはずの「Low Memory Mode」が起きた！

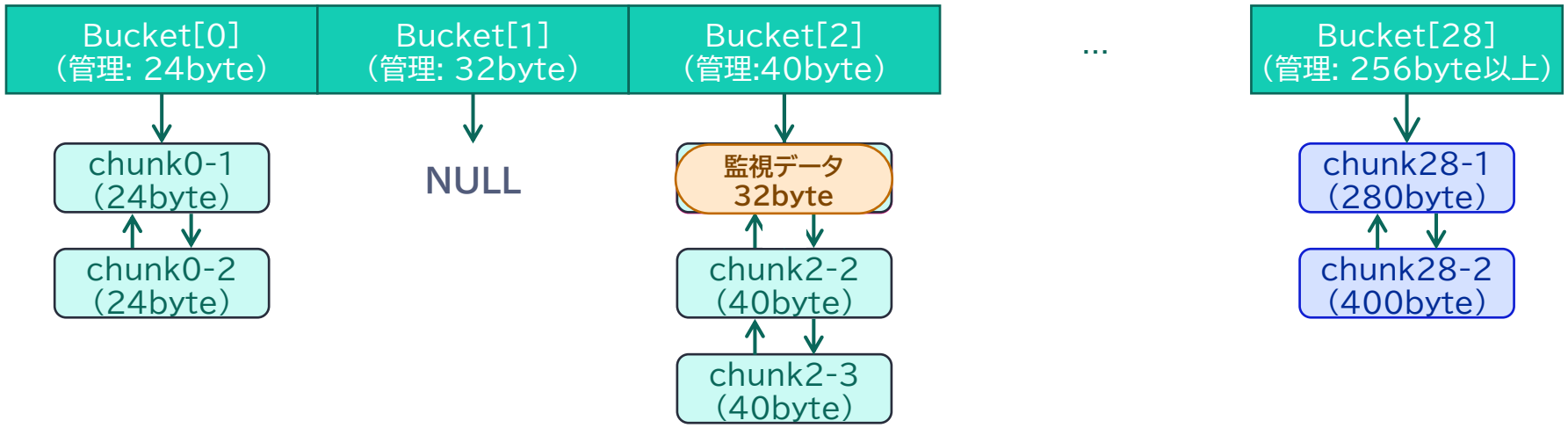
■メモリに監視データを格納するときの動作

- Bucketのインデックス番号順に、格納したい監視データを格納できるサイズを持つ空きChunkを探索
- 空きChunkが見つかれば、見つかったChunkにデータを格納して空きChunkリストから削除

事例②:発生しないはずの「Low Memory Mode」が起きた！

■メモリに監視データを格納するときの動作

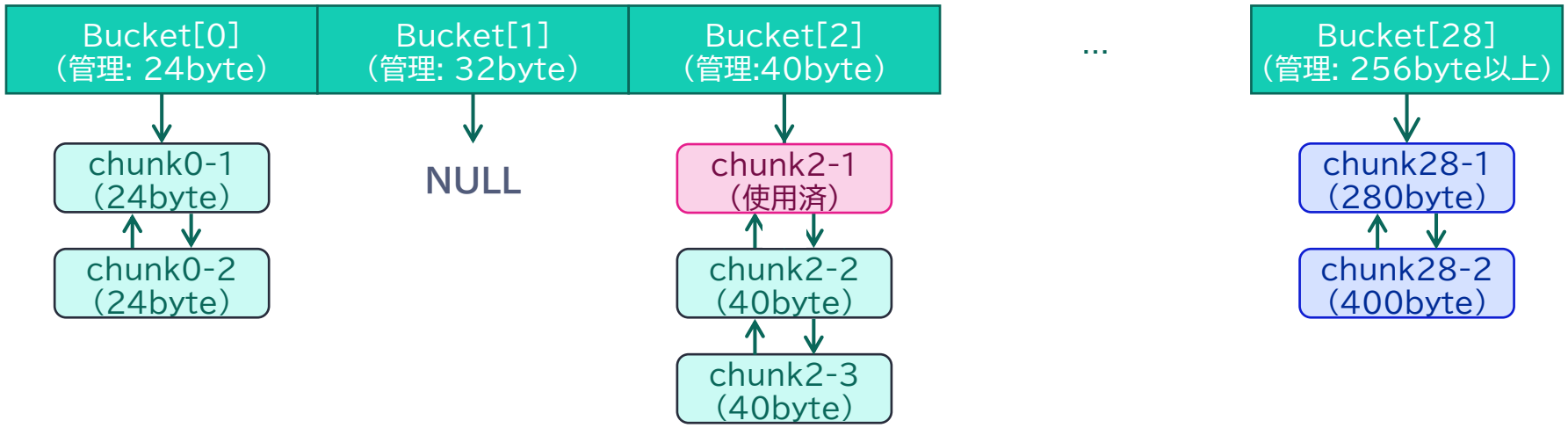
- Bucketのインデックス番号順に、格納したい監視データを格納できるサイズを持つ空きChunkを探索
- 空きChunkが見つかれば、見つかったChunkにデータを格納して空きChunkリストから削除



事例②:発生しないはずの「Low Memory Mode」が起きた！

■メモリに監視データを格納するときの動作

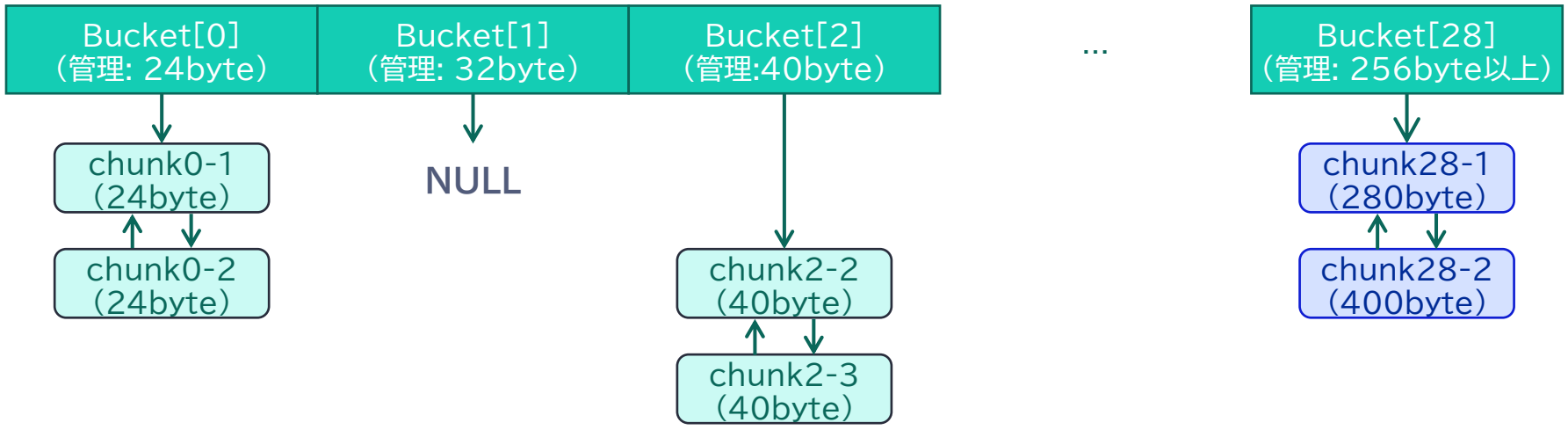
- Bucketのインデックス番号順に、格納したい監視データを格納できるサイズを持つ空きChunkを探索
- 空きChunkが見つかれば、見つかったChunkにデータを格納して空きChunkリストから削除



事例②:発生しないはずの「Low Memory Mode」が起きた！

■メモリに監視データを格納するときの動作

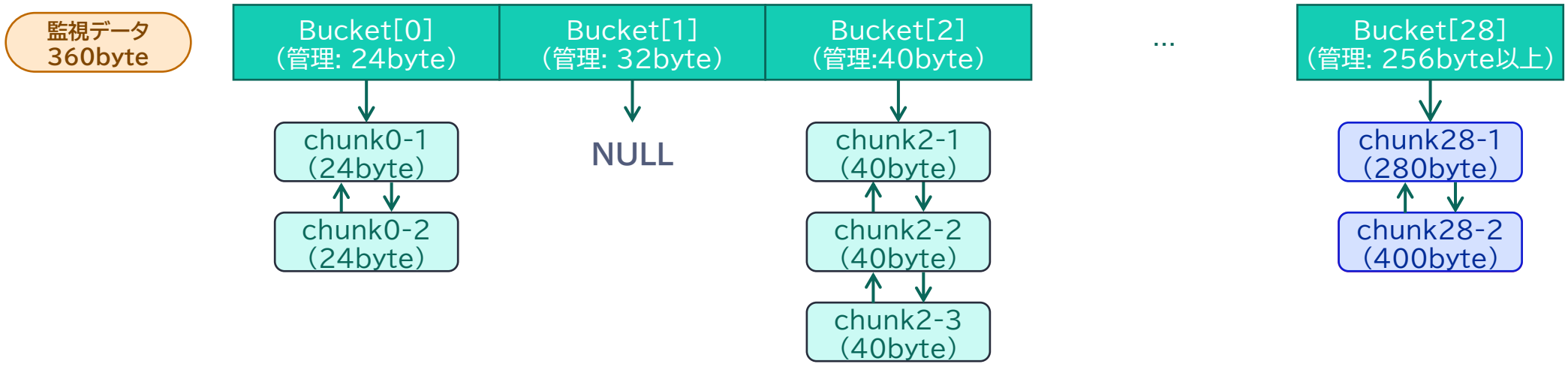
- Bucketのインデックス番号順に、格納したい監視データを格納できるサイズを持つ空きChunkを探索
- 空きChunkが見つかれば、見つかったChunkにデータを格納して空きChunkリストから削除



事例②:発生しないはずの「Low Memory Mode」が起きた！

■メモリに監視データを格納するときの動作

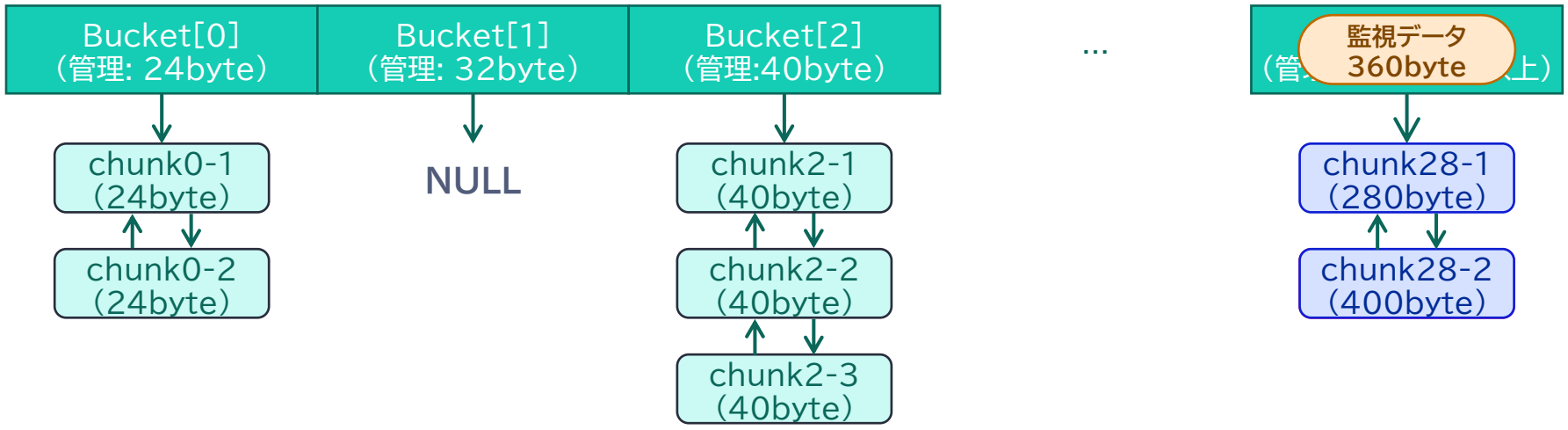
- 256byte以上の監視データを格納する場合、Bucket[28]が管理する空きChunkリストより格納できるChunkを探索
- Chunkリストの先頭から探索し、空きChunkが見つかれば、データを格納して空きChunkリストから削除
  - データを格納した余りが一定サイズあれば、それを分割して空きチャンクとして利用



事例②:発生しないはずの「Low Memory Mode」が起きた！

■メモリに監視データを格納するときの動作

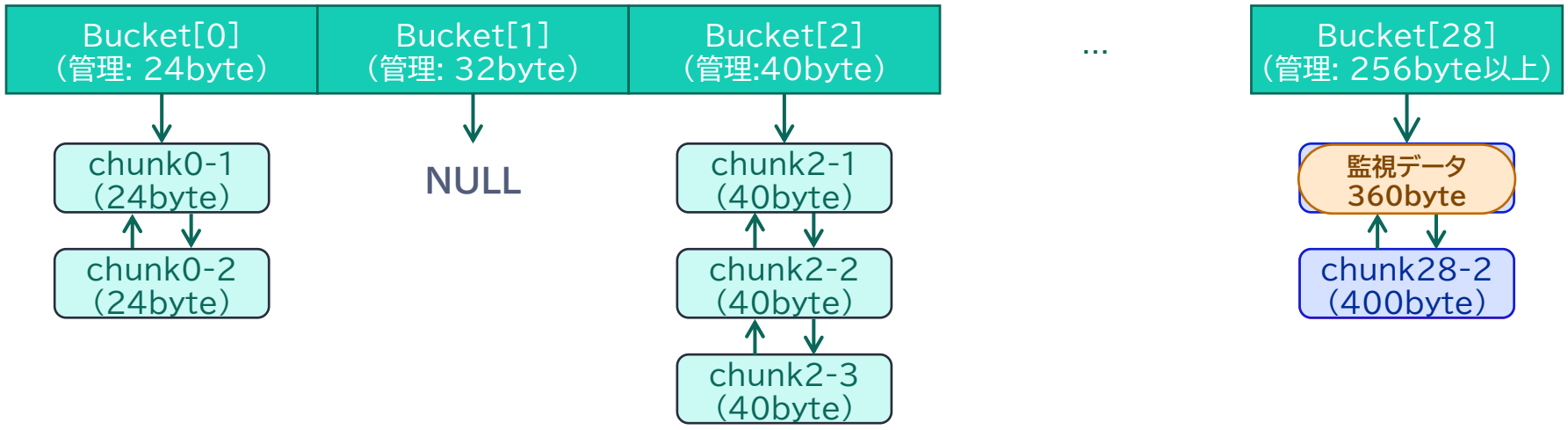
- 256byte以上の監視データを格納する場合、Bucket[28]が管理する空きChunkリストより格納できるChunkを探索
- Chunkリストの先頭から探索し、空きChunkが見つかれば、データを格納して空きChunkリストから削除
  - データを格納した余りが一定サイズあれば、それを分割して空きチャンクとして利用



事例②:発生しないはずの「Low Memory Mode」が起きた！

■メモリに監視データを格納するときの動作

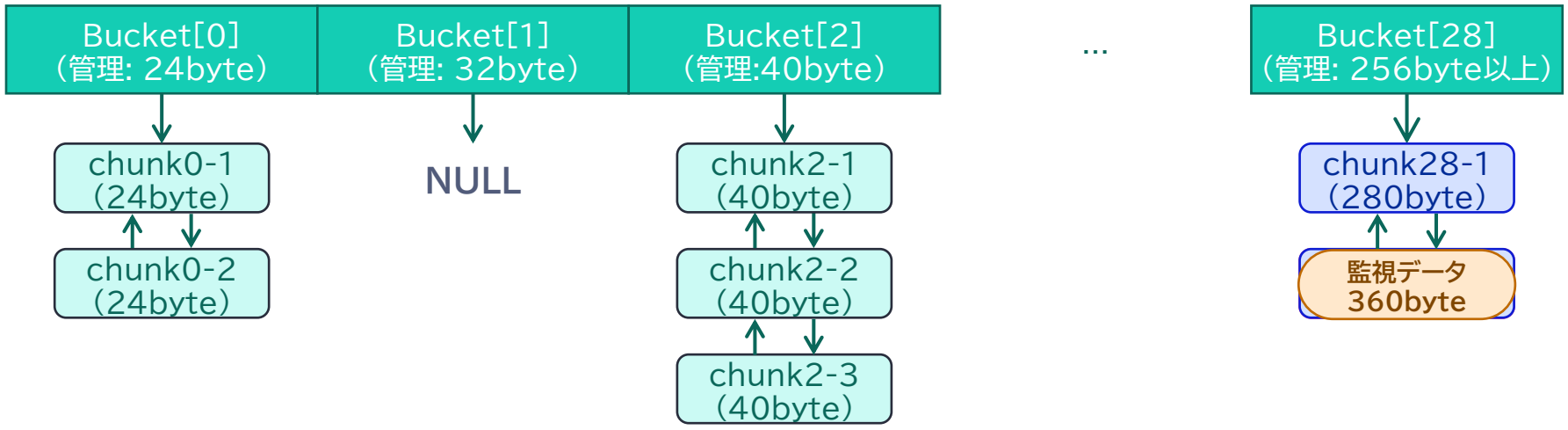
- 256byte以上の監視データを格納する場合、Bucket[28]が管理する空きChunkリストより格納できるChunkを探索
- Chunkリストの先頭から探索し、空きChunkが見つかれば、データを格納して空きChunkリストから削除
  - データを格納した余りが一定サイズあれば、それを分割して空きチャンクとして利用



事例②:発生しないはずの「Low Memory Mode」が起きた！

■メモリに監視データを格納するときの動作

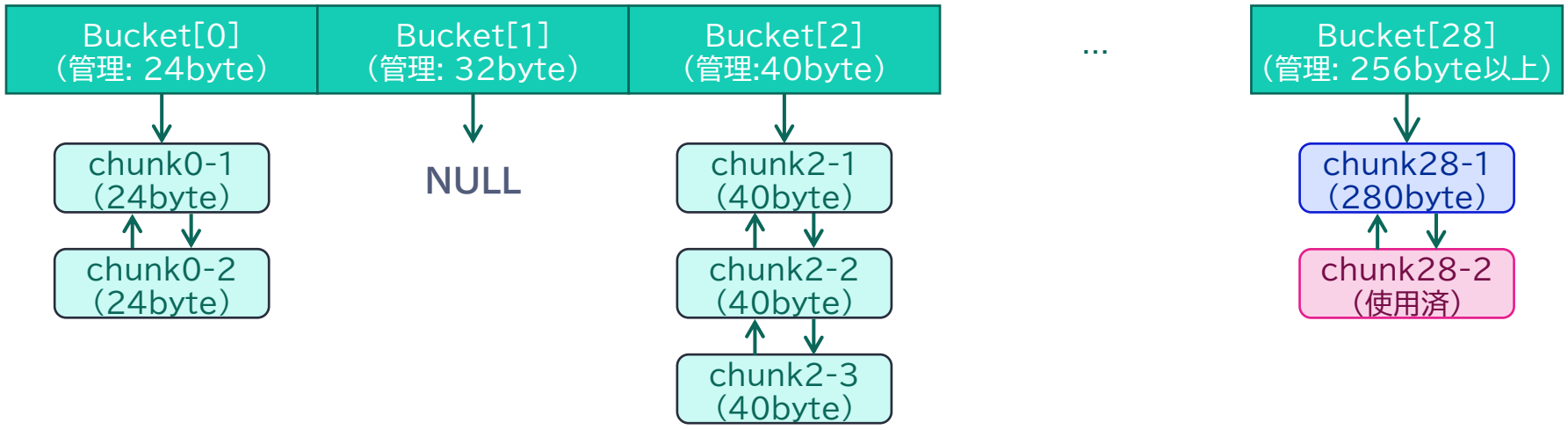
- 256byte以上の監視データを格納する場合、Bucket[28]が管理する空きChunkリストより格納できるChunkを探索
- Chunkリストの先頭から探索し、空きChunkが見つかれば、データを格納して空きChunkリストから削除
  - データを格納した余りが一定サイズあれば、それを分割して空きチャンクとして利用



事例②:発生しないはずの「Low Memory Mode」が起きた！

■メモリに監視データを格納するときの動作

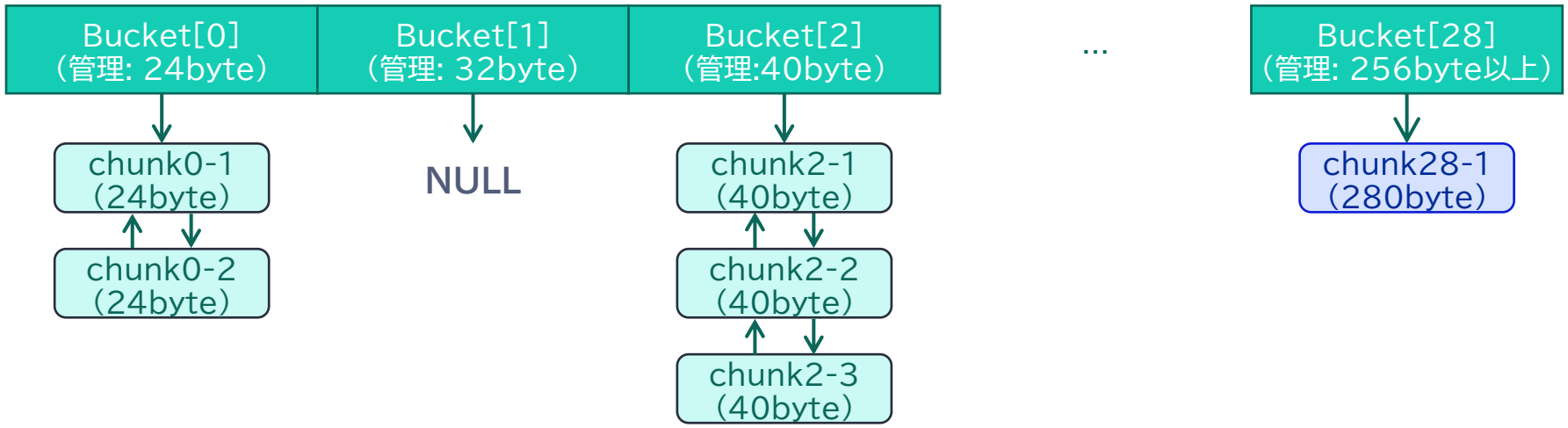
- 256byte以上の監視データを格納する場合、Bucket[28]が管理する空きChunkリストより格納できるChunkを探索
- Chunkリストの先頭から探索し、空きChunkが見つかれば、データを格納して空きChunkリストから削除
  - データを格納した余りが一定サイズあれば、それを分割して空きチャンクとして利用



事例②:発生しないはずの「Low Memory Mode」が起きた！

■メモリに監視データを格納するときの動作

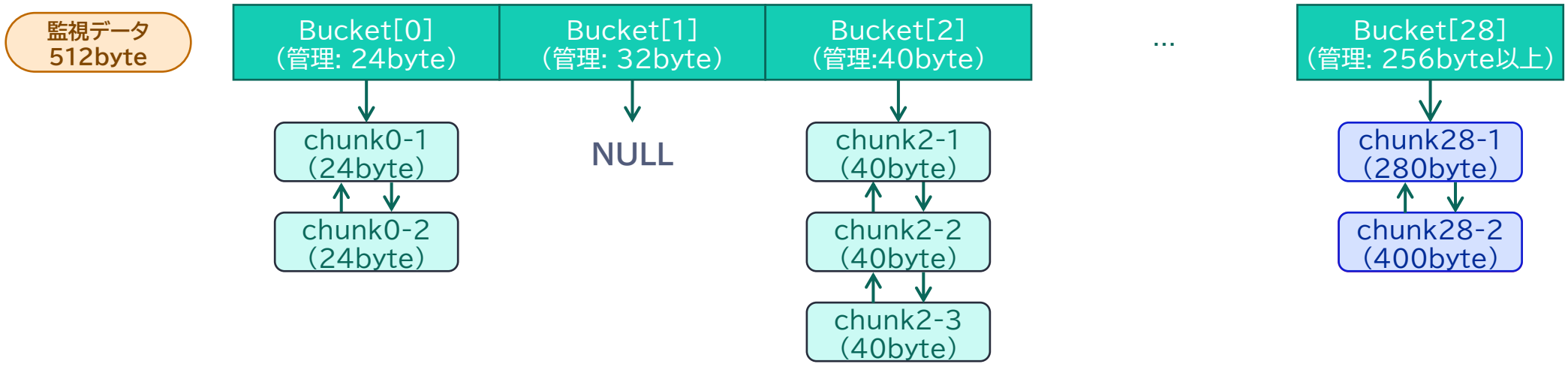
- 256byte以上の監視データを格納する場合、Bucket[28]が管理する空きChunkリストより格納できるChunkを探索
- Chunkリストの先頭から探索し、空きChunkが見つかれば、データを格納して空きChunkリストから削除
  - データを格納した余りが一定サイズあれば、それを分割して空きチャンクとして利用



事例②:発生しないはずの「Low Memory Mode」が起きた！

■監視データの格納に失敗した時の動作

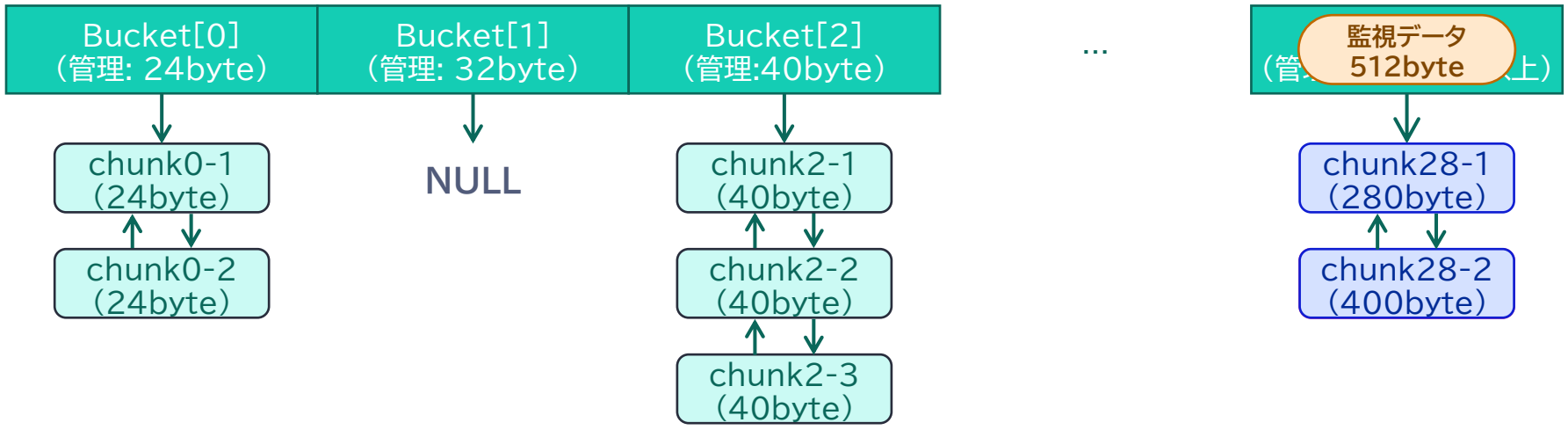
- **空きChunkが見つからなければデータ格納失敗と判定**され、Low Memory Modeとなる
  - 全てのChunkが使われており、空きChunkが存在しない（キャッシュが100%使用中）
  - 空きChunkはあるが、監視データを格納できるサイズのChunkが見つからない
    →**メモリには空きがあるのに格納するのに適したサイズのChunkが無いため、Low Memory Modeとなってしまう**



事例②:発生しないはずの「Low Memory Mode」が起きた！

■監視データの格納に失敗した時の動作

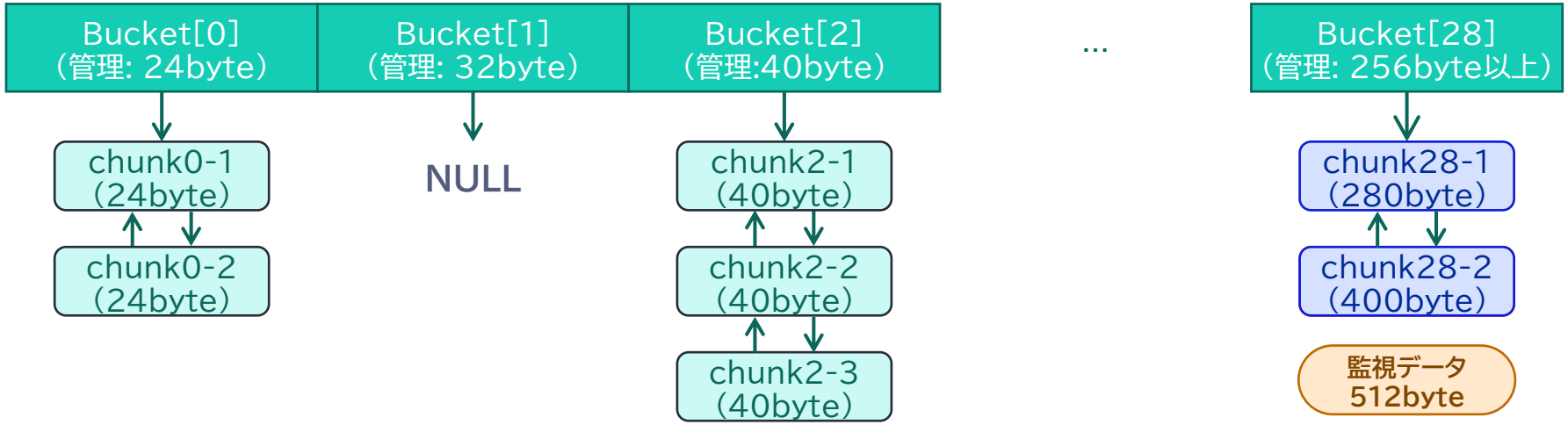
- **空きChunkが見つからなければデータ格納失敗と判定**され、Low Memory Modeとなる
  - 全てのChunkが使われており、空きChunkが存在しない（キャッシュが100%使用中）
  - 空きChunkはあるが、監視データを格納できるサイズのChunkが見つからない
    →**メモリには空きがあるのに格納するのに適したサイズのChunkが無いため、Low Memory Modeとなってしまう**



事例②:発生しないはずの「Low Memory Mode」が起きた！

■監視データの格納に失敗した時の動作

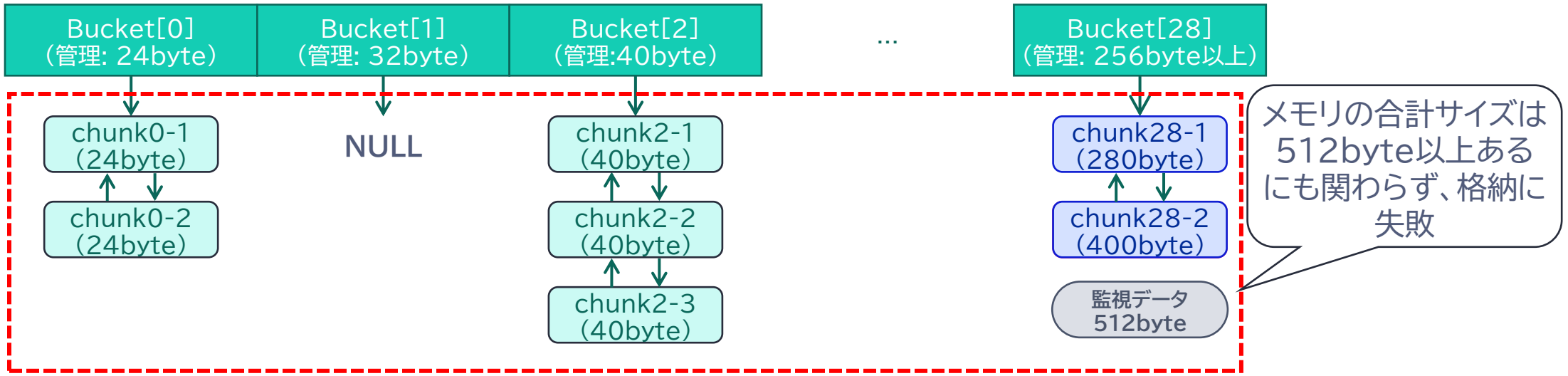
- **空きChunkが見つからなければデータ格納失敗と判定**され、Low Memory Modeとなる
  - 全てのChunkが使われており、空きChunkが存在しない（キャッシュが100%使用中）
  - 空きChunkはあるが、監視データを格納できるサイズのChunkが見つからない
    →**メモリには空きがあるのに格納するのに適したサイズのChunkが無いため、Low Memory Modeとなってしまう**



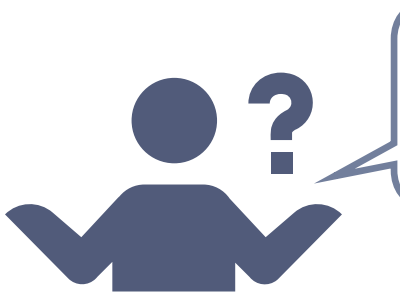
事例②:発生しないはずの「Low Memory Mode」が起きた！

■監視データの格納に失敗した時の動作

- **空きChunkが見つからなければデータ格納失敗と判定**され、Low Memory Modeとなる
  - 全てのChunkが使われており、空きChunkが存在しない（キャッシュが100%使用中）
  - 空きChunkはあるが、監視データを格納できるサイズのChunkが見つからない
    →**メモリには空きがあるのに格納するのに適したサイズのChunkが無いため、Low Memory Modeとなってしまう**



事例②：発生しないはずの「Low Memory Mode」が起きた！



今の話だと、時間が経つにつれてChunkの断片化が進み、
最終的には全て小さいサイズのChunkしか存在しなくなるのでは？

事例②:発生しないはずの「Low Memory Mode」が起きた！

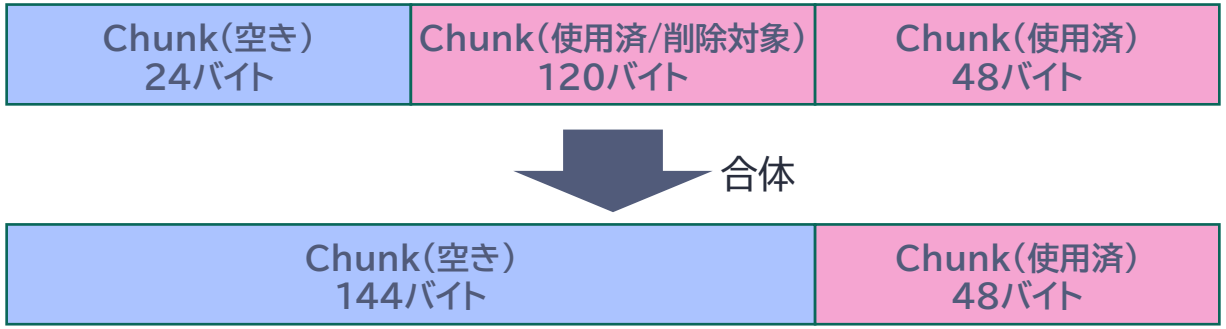
今の話だと、時間が経つにつれてChunkの断片化が進み、
最終的には全て小さいサイズのChunkしか存在しなくなるのでは？

小さいChunkを大きいChunkに再編するような処理は存在するが、
過度な期待は禁物

事例②:発生しないはずの「Low Memory Mode」が起きた！

■メモリからデータを削除する動作はあるものの・・・

- メモリにデータを格納する際、**24時間以上未参照のデータをキャッシュから削除した上でデータを格納する**
- 削除対象のChunkの「前後のChunk」の内容を参照して、使用済か空きかを確認する
  - 「前後のChunk」とはリスト構造での前後関係ではなく、メモリ空間の物理的なChunkの位置関係を指す
- 前後のChunkが空いていれば、それらのChunkと合体させて空きChunkのリストに追加する
  - 物理的な位置関係とChunkのサイズは無関係
- **データを格納できるサイズのChunkが用意されるという保証はない**



事例②:発生しないはずの「Low Memory Mode」が起きた！

■対処方法

方法:ValueCacheの値を大きく設定する

Zabbix Server Healthテンプレートを適用したZabbixサーバホストを用意し、
キャッシュ使用状況を確認した上で、十分に余裕のある値を設定する

```
### Option: ValueCacheSize
#       Size of history value cache, in bytes.
#       Shared memory size for caching item history data requests.
#       Setting to 0 disables value cache.

ValueCacheSize=8M
```

断片化自体は防げないが、空きメモリに余裕を持たせることで断片化されていないChunkを
見つけることができる可能性を高めることができ、事象発生を低減できる

まとめ

■本日はキャッシュに関連した事例2件をご紹介

- キャッシュが枯渇したり必要なメモリを確保できない場合、監視結果のロストやパフォーマンス低下が発生
- 主な対策としては、キャッシュに十分なサイズを設定すること
  - 設定サイズに対して**使用率がおおよそ40～50%で推移し、十分な空きがある状態となる**ことを推奨
- **Zabbix Server Health**テンプレートを適用してメモリの使用状況を確認することをおススメ！
  - Zabbix Server Healthは、Zabbixサーバ自身の稼働状況を監視するためのテンプレート
  - **サポート対応時に、本テンプレートを設定しておらず、原因究明に時間を要してしまうケースは結構あり！**

お問合せ先

■OSSサポート製品一覧

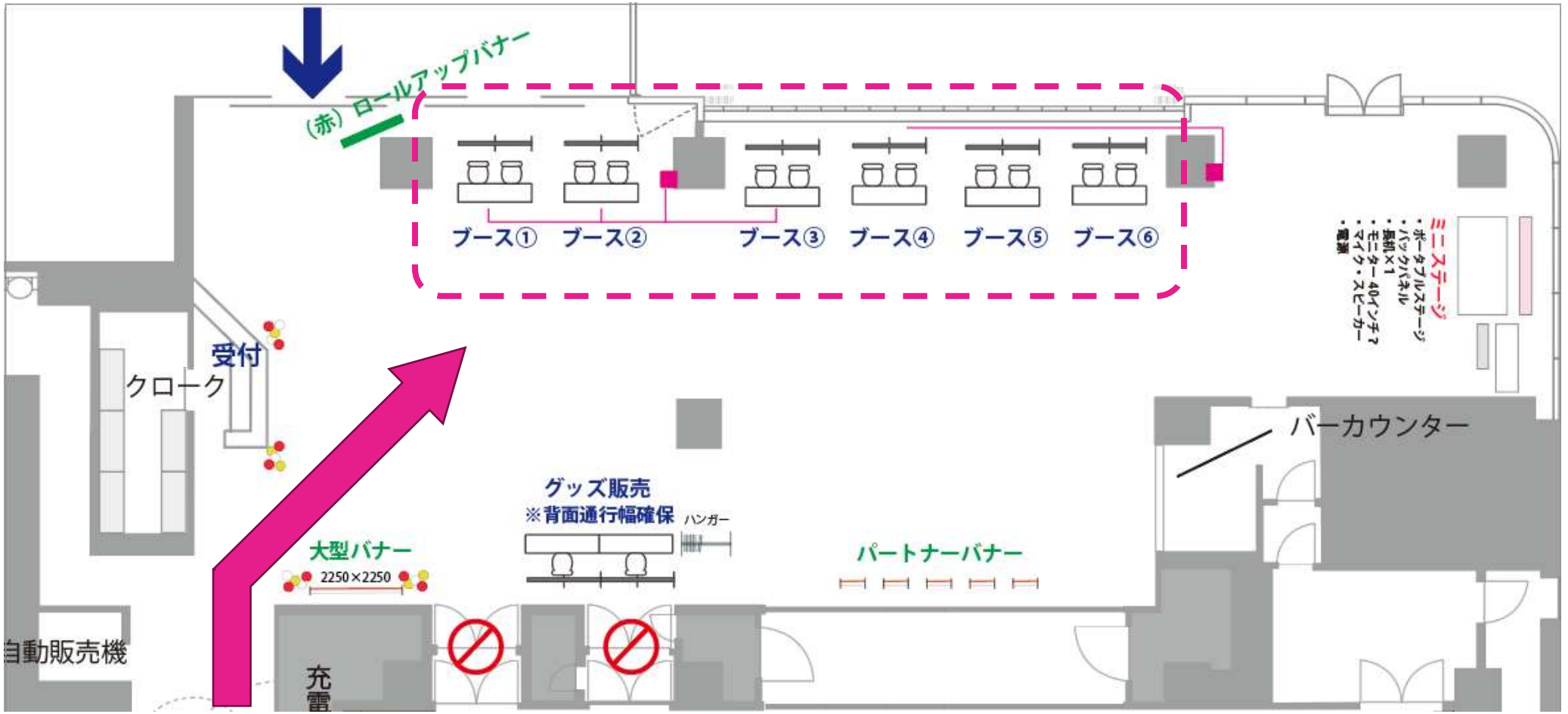
- https://jpn.nec.com/oss/solution/index.html

■Zabbixサポートサービス

- https://jpn.nec.com/oss/middle support/community zabbix/support.html

Zabbixの保守サポート、構築、その他お問い合わせ先
NEC　OSS推進センター
e-mail：oss@osspf.jp.nec.com

NECブースのご案内



\Orchestrating a brighter world