

sui-sei

消費者インサイトをリアルタイムに捉えるための
EC サイト向け高速検索エンジンサービス

目次

1. 検索エンジンの状況と課題
2. 分散 KVS を利用した検索エンジンの作成
3. 検索エンジン性能検証
4. 消費者インサイトを捉える検索
5. 実現方法
6. 有効性の実証
7. 今後の取り組み
8. まとめ

検索エンジンの状況と 従来の取り組み

sui-sei とは

サイト内検索エンジン ASP サービス

特徴

- 高速な検索処理
- 高負荷環境での安定した検索性能
- 検索ログ解析による検索最適化

検索エンジンの分類

■web サイト検索エンジン

■サイト内検索エンジン

web サイト検索エンジン



地图

[もっと見る](#) ▼

約 43,800,000 件 (0.25 秒)

他のキーワード: [なぜhoge](#)

[hogeとは - はてなキーワード - はてなダイアリー](#)

d.hatena.ne.jp/keyword/hoge ▼

「hoge」とは - 日本の情報処理技術者が替玉記号のように利用することが多い文字列。情報処理の世界ではメタ構文変数(Metasyntactic variable)とも呼ばれる。つまり「hoge」と書いてあったら「必要な文字に置き換...

[メタ構文変数 - Wikipedia](#)

[ja.wikipedia.org/wiki/メタ構文変数](#)

そのようなときに、「abcd」や「hoge」(ほげ)や「pakeratta」(パケラッタ)をはじめとするメタ構文変数を利用する。… 厳密には変数名ではない識別子(特定言語におけるクラス名やモジュール名など)について**hoge**等が使われた場合でも「メタ構文変数」と呼ぶ。

概要 - メタ構文変数の例 - メタ構文変数由来の固有名詞 - 脚注

→「ガニツガ」トノ所ニ「レ」ニ「エ」ヲ →「ガニツガ」ニ「エ」ヲ

google.com による検索例

サイト内検索エンジン

The screenshot displays the Amazon.co.jp homepage with a search for the term "category". The left sidebar shows a category menu with options like Kindle, MP3ストア, and Amazon Cloud Drive. The main content area shows search results for "category", including two books: "Category Theory (Oxford Logic Guides)" by Steve Awodey and "Categories" by Horst Schubert and Eva Gray. The search results include prices, availability, and delivery options.

amazon.co.jp マイストア Amazonポイント ギフト券 タイムセール 出品サービス ヘルプ

カテゴリからさがす すべて category 検索

Kindle
MP3ストア & Cloud Player
いつでもどこでも大好きな音楽を
Amazon Cloud Drive
5GBの無料ストレージ
Android アプリストア
日替り1日限定 無料アプリも
本・コミック・雑誌
DVD・ミュージック・ゲーム
家電・カメラ・AV機器
パソコン・オフィス用品
ホーム&キッチン・ペット
食品&飲料
ヘルス&ビューティー
ベビー・おもちゃ・ホビー
ファッション・バッグ・腕時計

category"

検索結果7,461件中1件から16件までを表示

なか見! 検索

Category Theory (Oxford Logic Guides) Steve Awodey (2010/8/13)
¥ 5,723 ペーパーバック **プライム**
18時間以内にご注文いただくと、2013/5/25 土曜日までにお届けします。
こちらからもご購入いただけます - ペーパーバック
¥ 4,927 新品 (12 出品)
¥ 7,480 中古品 (2 出品)

★★★★★ (2)
通常配送無料
その他のフォーマット
ハードカバー
洋書: 全2,396商品を見

なか見! 検索

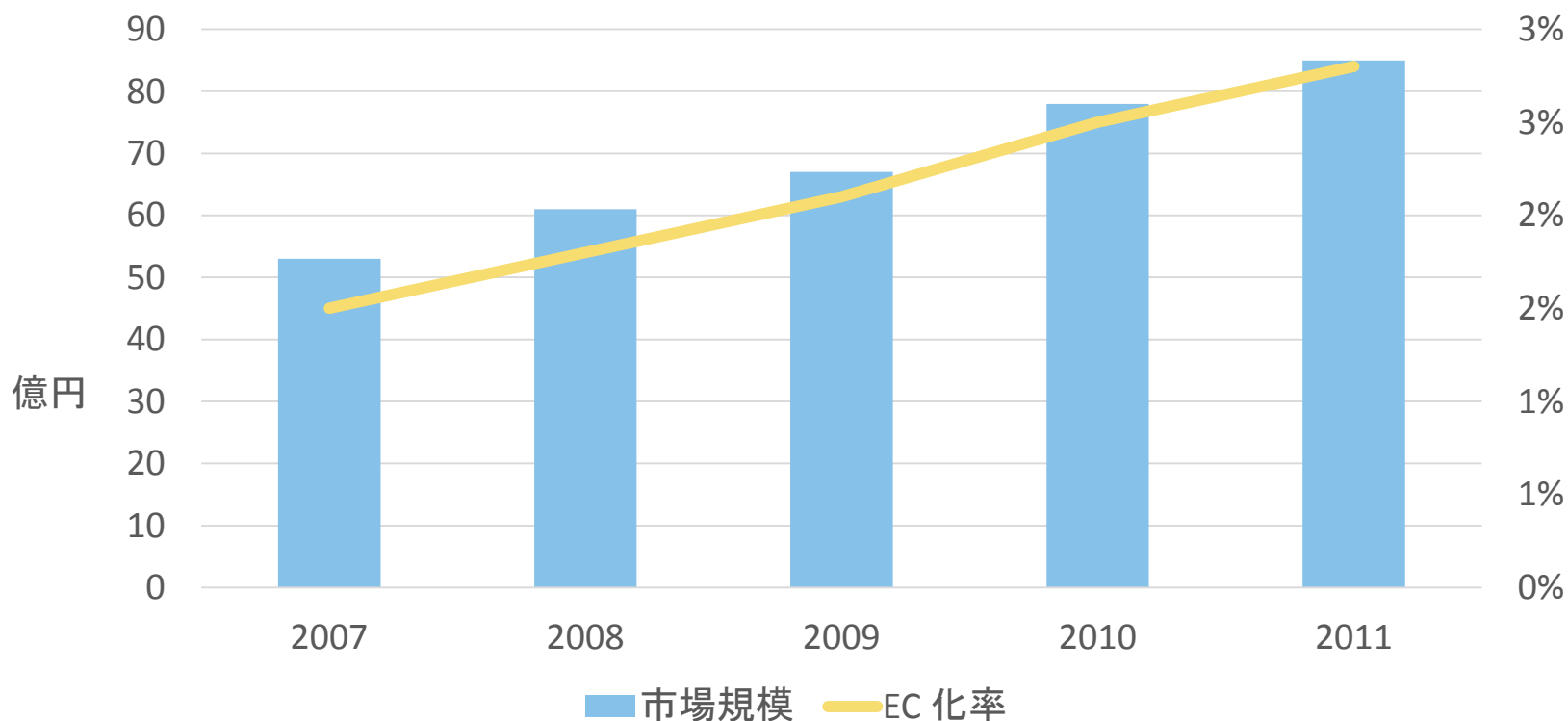
Categories Horst Schubert, Eva Gray (2011/11/11)
¥ 10,655 ペーパーバック **プライム**
13時間以内にご注文いただくと、2013/5/25 土曜日までにお届けします。

通常配送無料
その他のフォーマット
ハードカバー

amazon.co.jp による検索例

市場の拡大

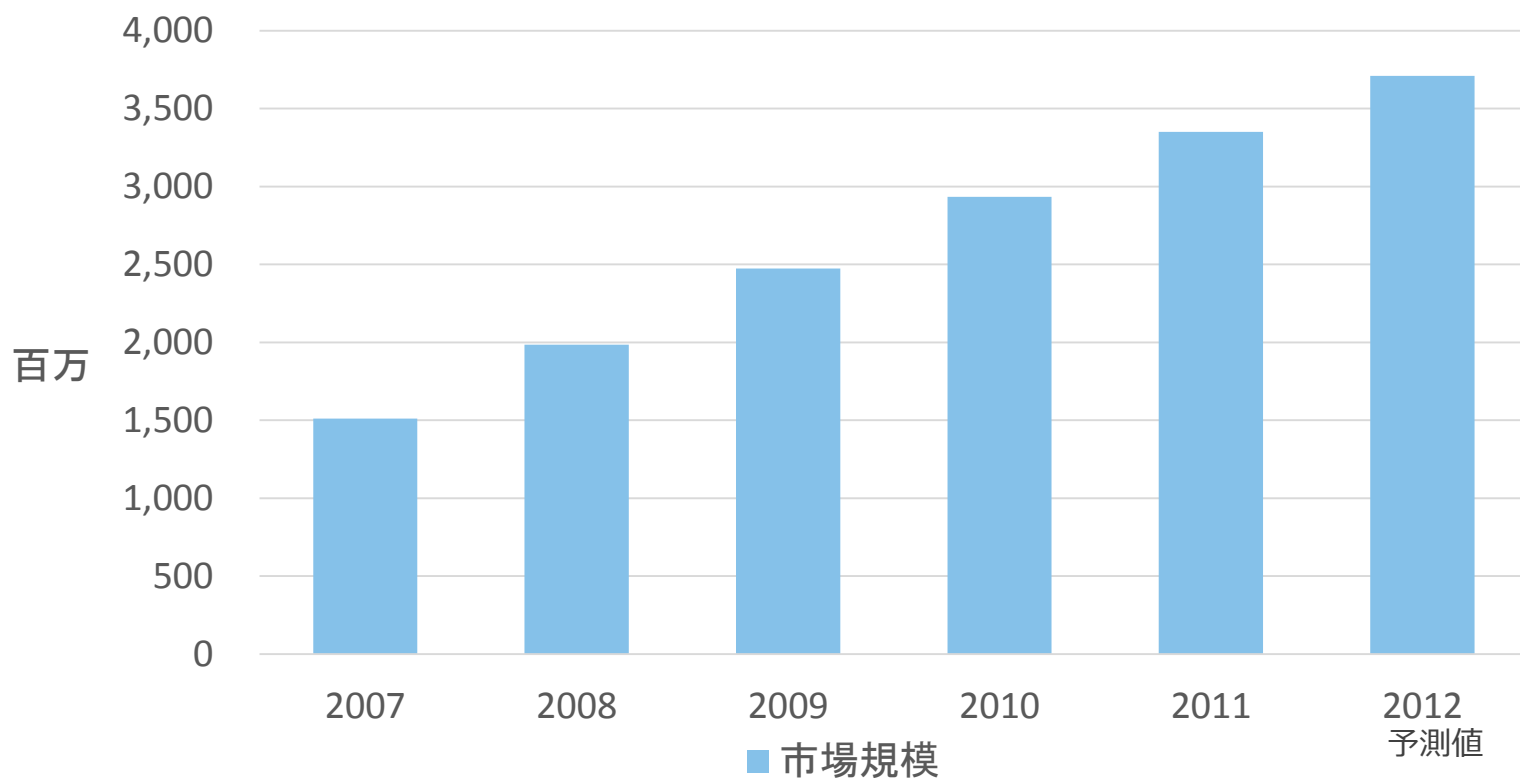
EC 市場規模と EC 化率の増大



経済産業省「平成 23 年度電子商取引に関する市場調査」より作成

市場の拡大

サイト内検索エンジン市場規模の推移



株式会社 矢野経済研究所

「エンタープライズサーチ・サイト内検索エンジン市場に関する調査結果 2011」より作成

市場の拡大

EC サイト市場規模の拡大



サイト内検索エンジンへの
需要拡大

検索エンジンへの取り組み

- ・ アルゴリズムの改良
- ・ ハードウェアの進化
- ・ 分散並列処理

→ 検索処理の高速化に集中

sui-sei 開発の動機

既存の取り組みの大部分は
web サイト検索の改善



サイト内検索に特化した
検索エンジンが必要

既存課題

1. 検索対象・候補の増加による
検索速度低下
2. サイト内検索に特化した機能の不足

既存課題への対応策 1

リレーショナルデータベース(RDB) を
利用した検索



分散キーバリューストア(KVS) を
利用した検索

既存課題への対応策 2

サイト内検索に特化する必要



消費者インサイトを
捉える機能を作成

分散 KVS を利用した 検索エンジンの作成

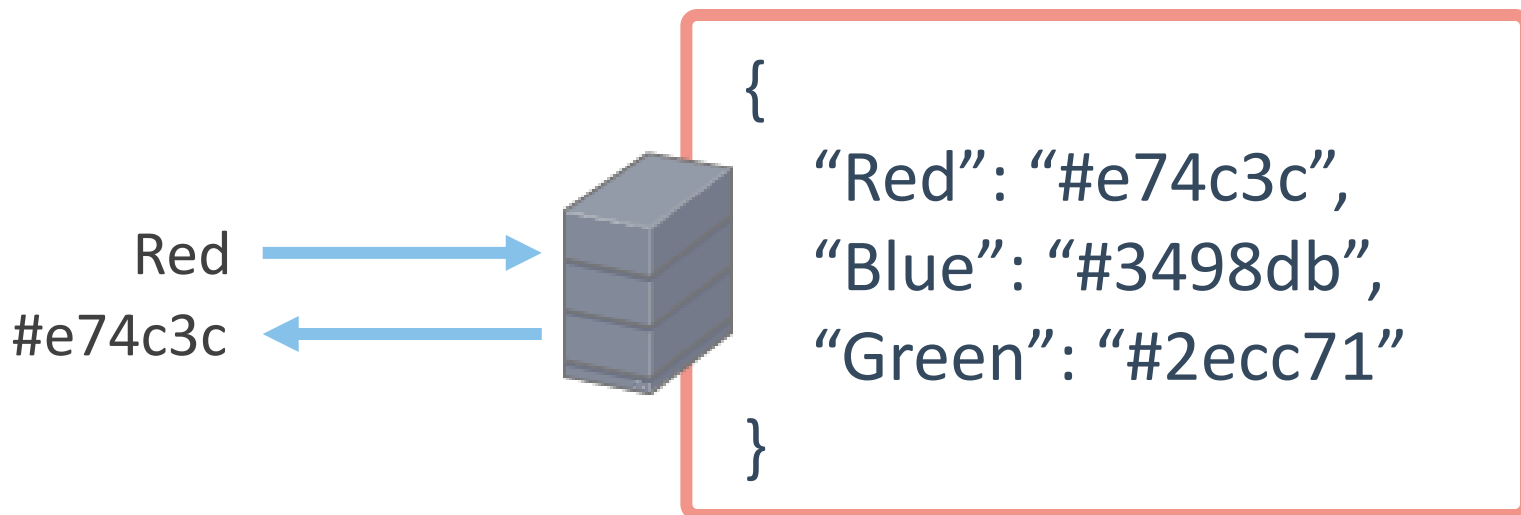
分散 KVS とは

- NOSQL データベース
- 単純な形式でデータを保存
- 複数台のマシン間でのデータ冗長化

分散 KVS とは

KVS (memcached など)

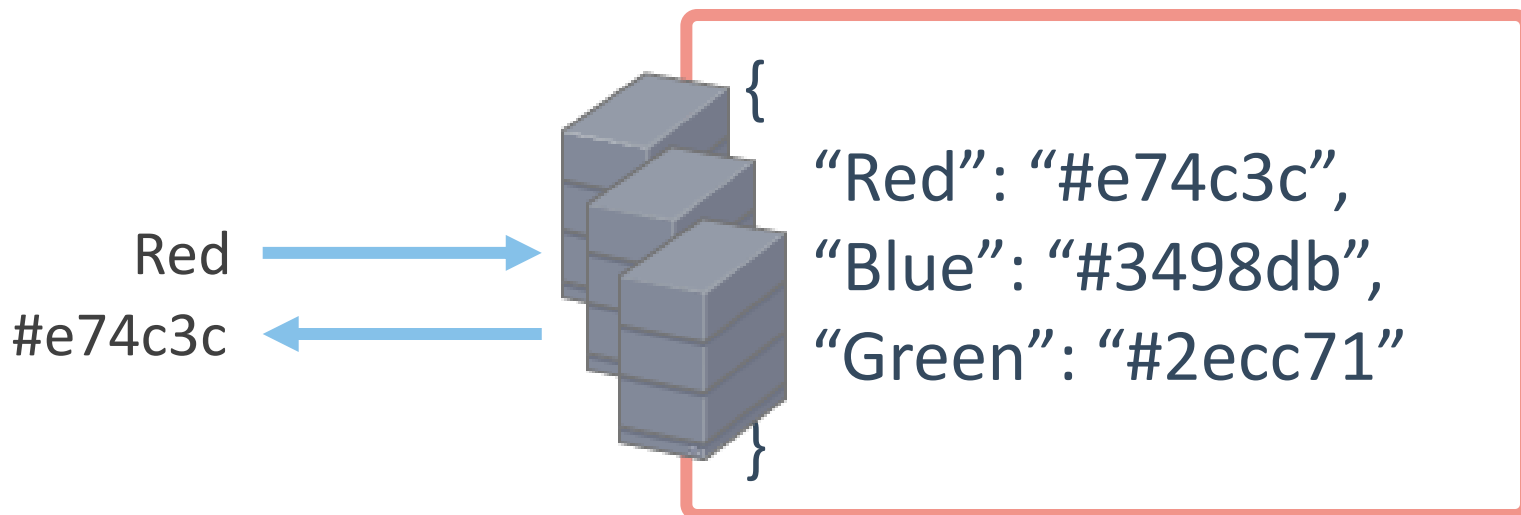
キーに紐付けてデータを保存



分散 KVS とは

分散 KVS (okuyama など)

複数台のマシンに分散して
キーに紐付けたデータを保存



分散 KVS のメリット

- 高いデータ処理能力
- データベースの水平拡張能力
- 汎用ハードウェアの利用
- データ分散による高可用性

RDB を利用した検索の課題 1

検索対象となるデータの増大



RDB のデータ構造は
検索に特化したものではない



検索に要する時間が増大

分散 KVS を利用した課題解決 1

検索対象となるデータの増大



単純なデータ構造のため
検索データ取得が容易で高速



検索に要する時間を
抑えることが可能

RDB を利用した検索の課題 2

高負荷状態



- ・ 1 台のハードウェアで動作する
- ・ 複数台での処理並列化が難しい



高価なハードウェアへの
置き換えが必要

分散 KVS を利用した課題解決 2

高負荷状態



- ・ 複数台のハードウェアで動作する
- ・ 処理並列化と分散化が容易



安価なハードウェアを
追加することでスケールアップ可能

サイト内検索特有の課題解決策

サイト内検索対象はほぼ一定

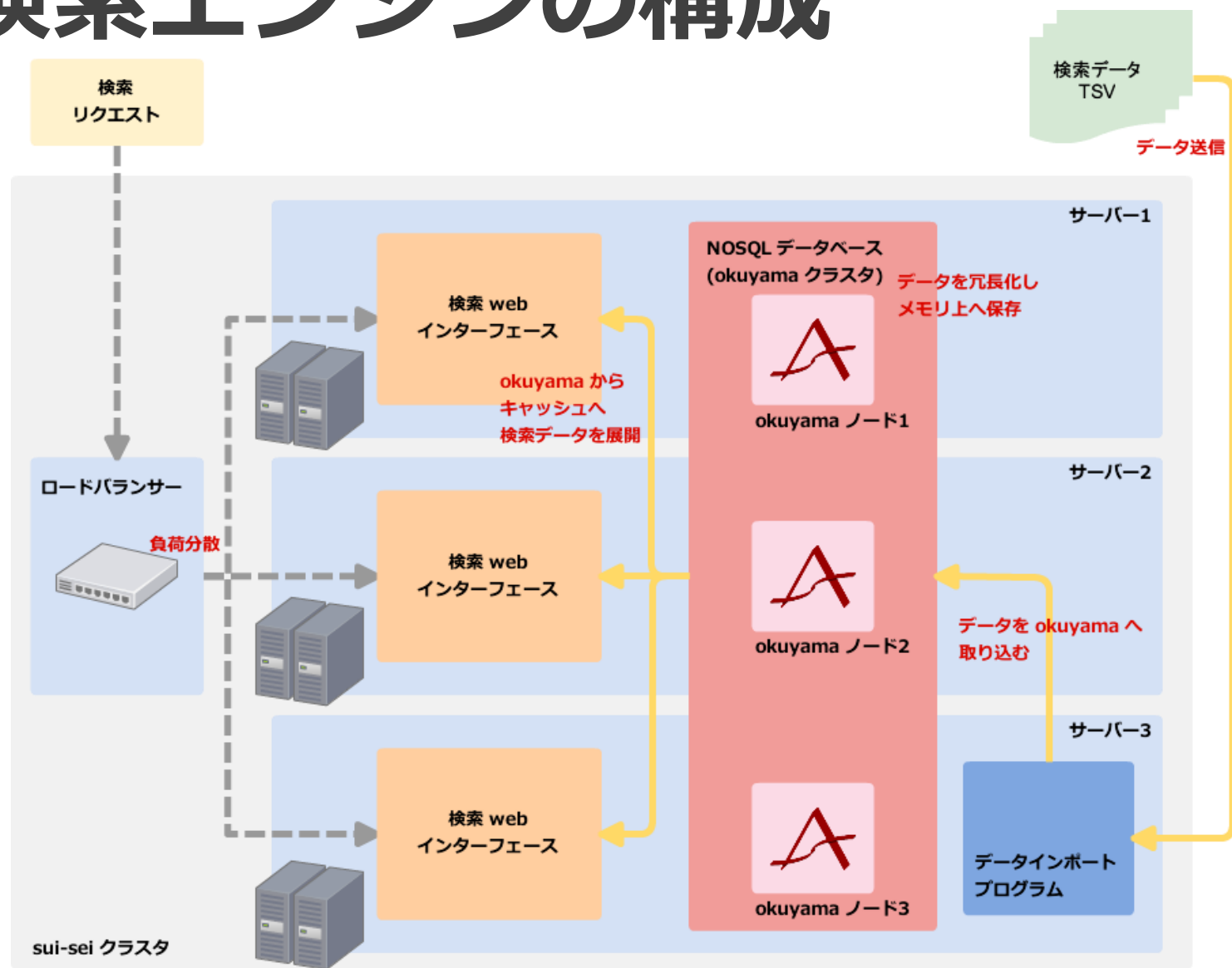


検索結果のパーツを用意

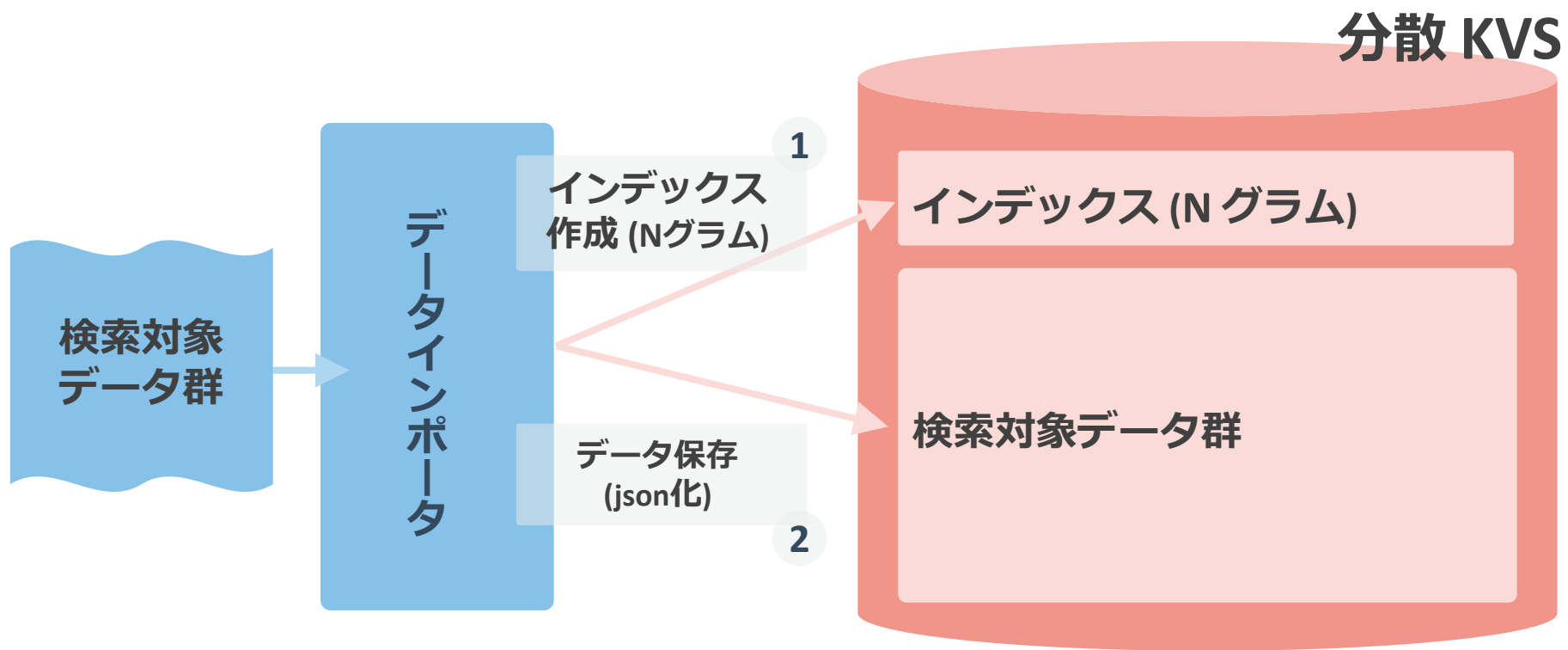


パーツを組み合わせて
検索結果を高速に作成

検索エンジンの構成

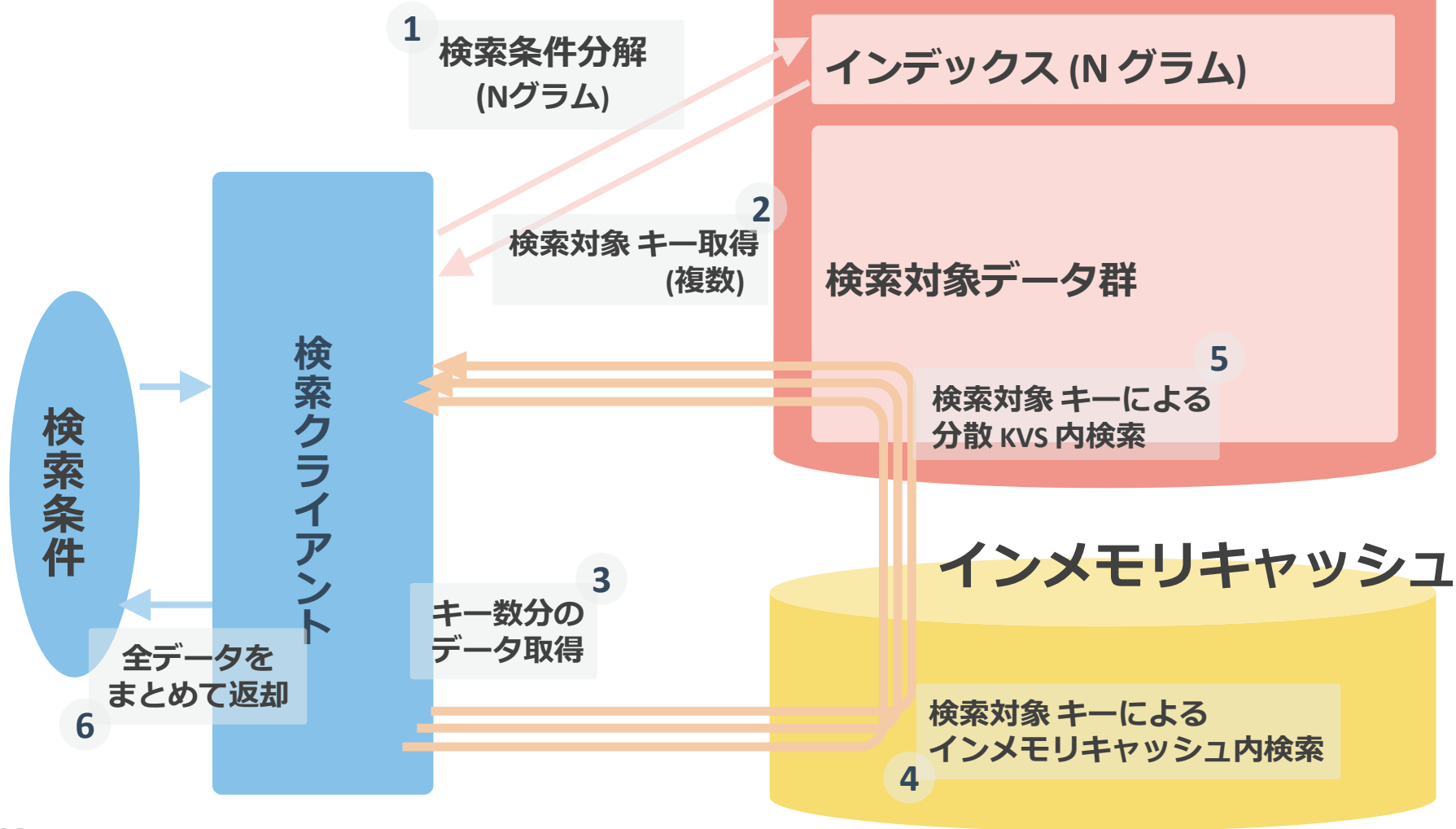


検索対象データの取り込み



検索処理(データ取得部分)

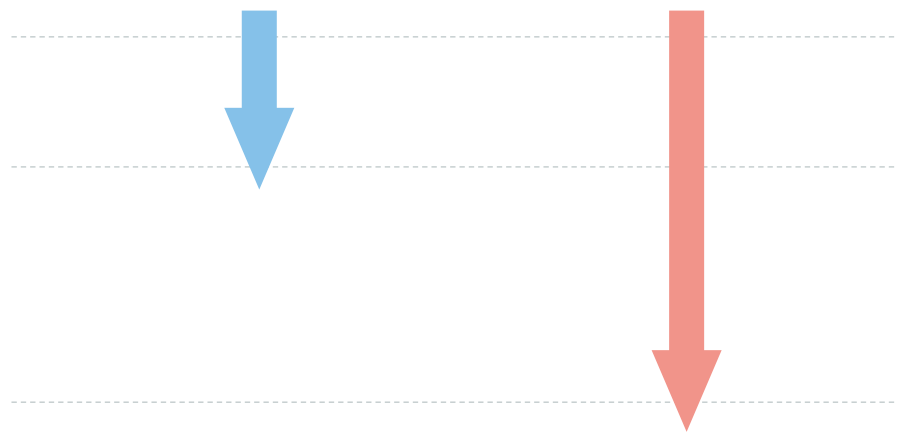
分散 KVS



検索処理

検索開始
検索キャッシュ

分散 KVS データ



検索時間

短

長

検索エンジン性能検証

要求性能

1. 1 秒間に最大 70 回の検索可能
2. 6 時間連続の検索要求に対し
性能が一定

→ それぞれ 1 回あたり
1 秒以内に検索結果を返却可能

検証条件

対象データ

- 全文検索
- 検索対象部分 40 字 ～ 100 字程度
- 5 万件

検証環境

- クラウドコンピューティング事業者 3 社
- CPU 2 GHz * 4 コア / メモリ 16 GB

検証条件

検索パターン 1

- Apache JMeter を利用したシナリオ実施
検索結果 100 件 ~ 5000 件の12パターンを
毎回ランダムに使用
- 検索頻度
 - 秒間 6 回 → 108 回へと 12 回刻みで増加
 - 2 分間検索継続

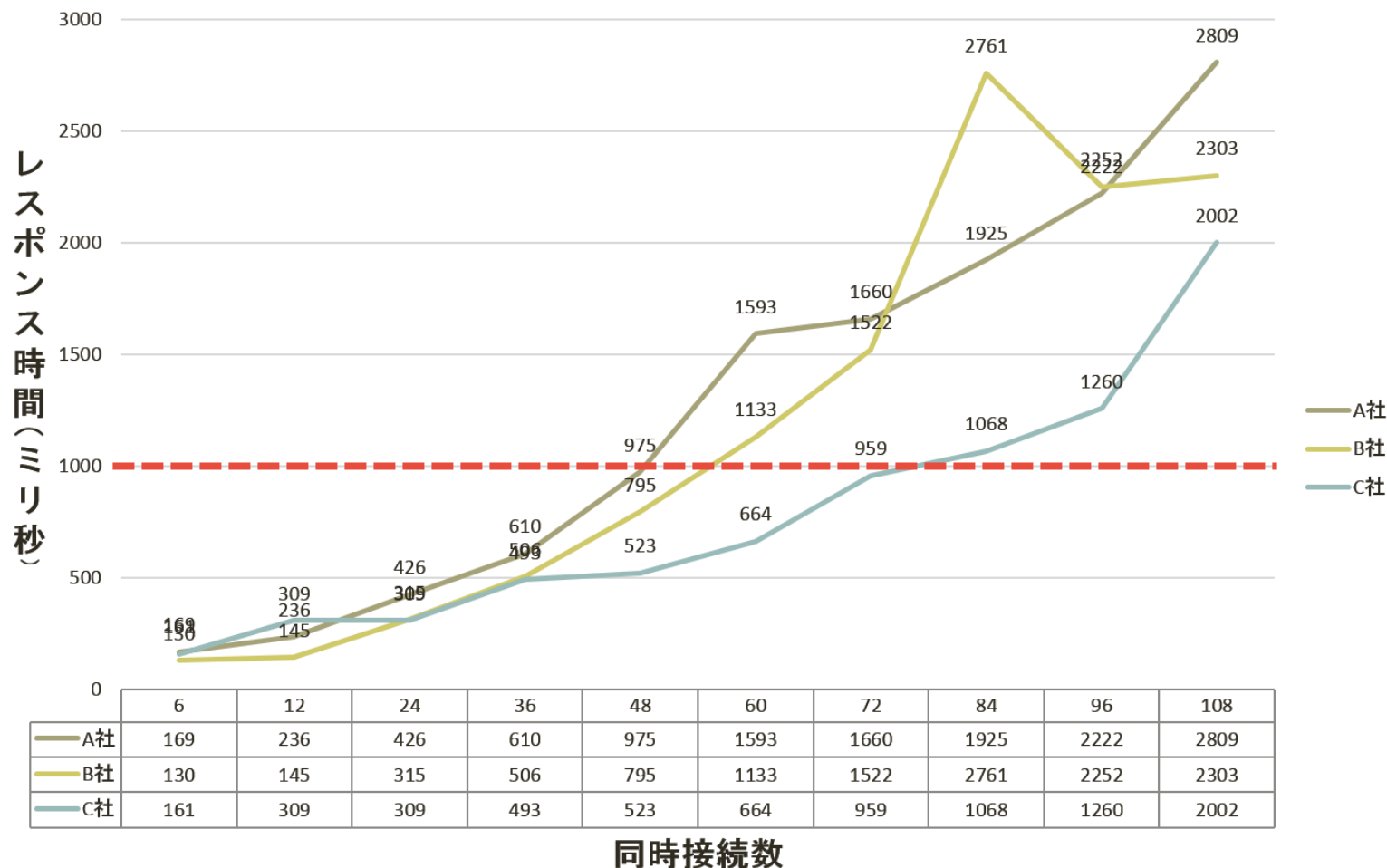
検証条件

検索パターン 2

- Apache JMeter を利用したシナリオ実施
検索結果 100 件 ~ 5000 件の12パターンを
毎回ランダムに使用
- 検索頻度
 - 秒間 48 回検索に固定
 - 6 時間連続

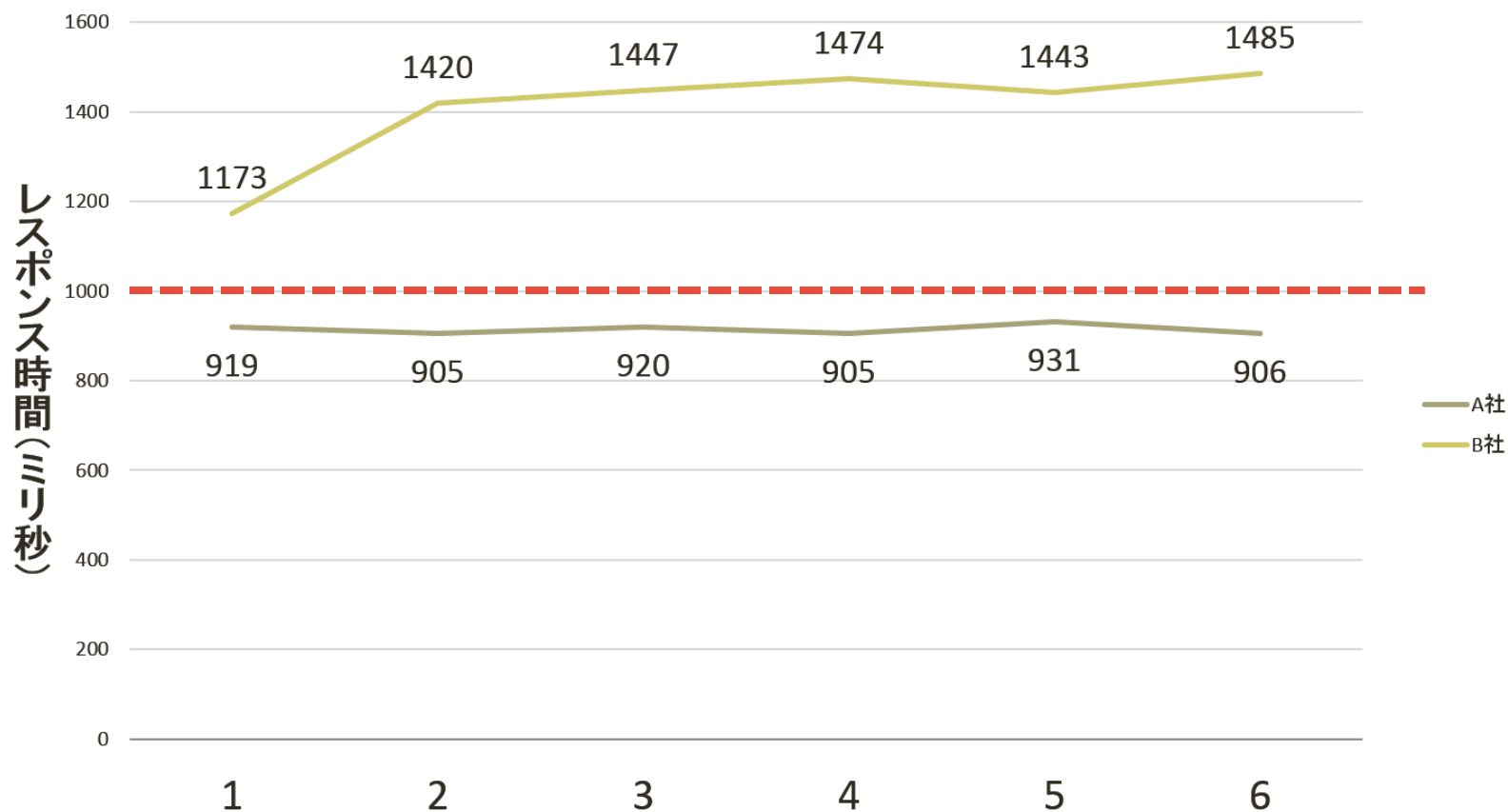
検索パターン 1 結果

1リクエストあたりのレスポンス時間の平均値(ミリ秒)



検索パターン 2 結果

長時間の負荷継続時のレスポンス速度の変化



負荷継続時間(単位:時間) ※同時接続数は48で固定

考察

検証パターン 1

- 要求性能を満たすことが可能
ただし、サーバーの性能や
サービス事業者の影響を受ける
- 今回の検証で要求性能以下の環境も
より上位性能のサービスを利用することで
要求性能を満たす

考察

検証パターン 2

- 要求性能を満たすことが可能
ただし、サーバーの性能や
サービス事業者の影響を受ける
- 負荷を継続しても
検索処理に要する時間は安定している

課題

■キャッシュ効率化

■省メモリ化

■サイト内検索に特化した機能の追加

消費者インサイトを 捉える検索

検索対象による特性の差異

web サイト検索

検索対象 → 不特定多数

検索結果 → 検索条件に対して正確

検索対象による特性の差異

サイト内検索

検索対象 → 特定・有限の検索対象

検索結果 → 検索対象(名前や要素)に対して
正確

“バック”の検索例

web サイト検索

優先順位

1. バック
2. バックの含まれる複合語
3. パック/バッグ

“バック”の検索例

サイト内検索 - 検索失敗例

「“バック”に一致する商品はありませんでした。」

→ 販売機会の損失

“バック”の検索例

web サイト検索 – 理想

優先順位

1. バッグ
2. 鞆 / カバン / かばん
3. パック / バック

web サイト検索の特性

検索語が正



莫大な検索対象の中から
一致する検索結果を提示

サイト内検索の特性

検索対象が正



検索対象が最適となるように
適宜検索語を修正する

サイト内検索の特性

検索条件より判断される
検索対象や消費者ニーズを優先

消費者インサイトを考慮した
検索結果を作り出す必要

消費者インサイト

消費者にとって無自覚なニーズ

サイト内検索によって
潜在的な購入意欲を刺激

実現方法

検索手法の例

- 利用者全体の傾向を反映した結果作成
- 検索履歴による最適結果作成
- 検索結果 0 件の場合に
検索条件から推測される検索結果の作成

利用者全体の傾向を反映

人気のモノを反映して
検索結果を作成する

全利用者のログを集計
・ 利用傾向を反映

履歴による検索結果の最適化

過去に購入したモノを考慮して
検索結果を作成する

ユーザーの検索・購買履歴を蓄積

- ・ 類似ユーザーの傾向を反映
- ・ 購買履歴を反映

「検索結果 0 件」の回避

検索結果が 0 件となる場合に
検索条件から判断して
必要とされる検索結果を作成する

まとめ

まとめ

- サイト内検索エンジンの基盤技術として分散 KVS を利用することは実用に足りる
- web サイト検索と比較して
サイト内検索では消費者インサイトを捉えるための機能・施策を必要とする