

RDB技術者のための

NoSQL

ガイド

渡部 徹太郎 [監修・著]

河村 康爾／北沢 匠／佐伯 嘉康／佐藤 直生

原沢 滋／平山 毅／李 昌桓 [著]

エンタープライズRDBエンジニアに贈る、 NoSQL活用の最新バイブル登場！

- ◎バズワード「NoSQL」を分解して実態を解明！
- ◎RDBの課題をいかにNoSQLで解決できるか？
——HadoopやDWHとの使い分けも丁寧に説明！

• NoSQL 最新バージョン

MongoDB 3.2 / Cassandra 3.0 / Redis 3.0 /
HBase 1.1 / Neo4j 2.3 / Couchbase Server 4.1

• クラウド

Amazon DynamoDB / Microsoft Azure DocumentDB

想定される
ユースケースも
多数掲載！

RDB技術者のための NoSQL ガイド

渡部 徹太郎 [監修・著]
河村 康爾／北沢 匠／佐伯 嘉康／佐藤 直生
原沢 滋／平山 毅／李 昌桓 [著]

エンタープライズRDBエンジニアに贈る、 NoSQL活用の最新バイブル登場！

- ◎バズワード「NoSQL」を分解して実態を解明！
- ◎RDBの課題をいかにNoSQLで解決できるか？
——HadoopやDWHとの使い分けも丁寧に説明！

・NoSQL 最新バージョン
MongoDB 3.2 / Cassandra 3.0 / Redis 3.0 /
HBase 1.1 / Neo4j 2.3 / Couchbase Server 4.1
・クラウド
Amazon DynamoDB / Microsoft Azure DocumentDB

想定される
ユースケースも
多数掲載！

 秀和システム

RDB技術者のための

NoSQL

ガイド

渡部 徹太郎 [監修・著]

河村 康晴 / 北沢 匠 / 佐伯 嘉康 / 佐藤 直生

原沢 滋 / 平山 毅 / 李 昌桓 [著]

 秀和システム

謝 辞

本書を発刊するに当たって共著していただいた河村 康爾さん、北沢 匠さん、佐伯 嘉康さん、佐藤 直生さん、原沢 滋さん、平山 毅さん、李 昌桓さんに感謝いたします。

また、本書の執筆に際して業務内容を調整していただいた菊地原 拓さん、内容を添削していただいた佐々木 政昭さん、清水 隆介さん、添田 健輔さん、饗庭 秀一郎さんに感謝いたします。

最後に、私が執筆できる時間を作るために、休日に子供の面倒を見てくれた妻に深く感謝いたします。

渡部 徹太郎

注 意

- 1. 本書は著者が独自に調査した結果を出版したものです。
- 2. 本書は内容に万全を期して作成しましたが、万一ご不審な点や誤り、記載漏れなどお気づきの点がありましたら、出版元まで書面にてご連絡ください。
- 3. 本書の内容に関して運用した結果の影響については、上記にかかわらず責任を負いかねますのであらかじめご了承ください。
- 4. 本書およびソフトウェアの内容に関しては、将来予告なしに変更されることがあります。
- 5. 本書の例に登場する会社名、名前、データは特に明記しない限り、架空のものです。
- 6. 本書の一部または全部を出版元から文書による許諾を得ずに複製することは禁じられています。

商 標

本書に記載されている会社名、製品名は各社の商標または登録商標です。

はじめに

近年、NoSQLが当たり前になってきています。数年前は、NoSQLといえばオープンソースの尖った技術というイメージでしたが、近年は大手ベンダが商用製品やクラウドサービスを続々と登場させてきており、エンタープライズでの利用事例も増えています。

今までに出版されたNoSQLの書籍は「最先端技術の解説書」というイメージが強かったと思います。それらに対し本書では、一般のRDBエンジニアを対象として、エンタープライズ環境でNoSQLをどのように活用すべきかを説明します。

本書で伝えたい事は三つあります。

一つ目は、NoSQLというバズワードの実態を説明することです。NoSQLという言葉は、特定のプロダクトを意味する場合もあり、RDBに搭載されたSQLではないインターフェースを意味する場合もあり、定義があいまいです。そこで本書ではKVS、ドキュメントDB、グラフDB、そしてRDBのNoSQLインターフェースという明確な用語を用いて、NoSQLを分解し、説明していきます。

二つ目は、エンタープライズ視点でのNoSQL活用方法です。エンタープライズでデータベースを使う場合は、機能の評価だけでなく、運用、セキュリティ、サポート体制といった非機能に関する評価が重要ですので、その点をしっかり説明します。また、企業におけるRDBの課題を解決するという視点で、NoSQLだけではなく、HadoopやDWHといったデータ処理技術との使い分けを明確に説明します。

三つ目は、NoSQLの正確な最新情報を記載している点です。NoSQLは進歩が速いため、昔の書籍やインターネットのブログに載っている情報は、間違っている可能性があります。本書では、紹介するデータベースの最新情報について、それぞれのデータベースのスペシャリストに正確な情報を掲載してもらっています。

2016年1月
渡部 徹太郎

Contents 目次

第1章 前提

1-1	この本で伝えたいこと	2
1-1-1	NoSQLというバズワードの実態を理解	2
1-1-2	エンタープライズ視点のNoSQL活用方法	3
1-1-3	NoSQLの最新情報	6
1-2	想定する読者	6
1-3	本書の構成	7

第2章 イントロダクション

2-1	RDBだけだと辛くないですか？	10
2-1-1	RDBはデファクトスタンダード	10
2-1-2	Sierのエンジニアにとってのデータベースの経験	11
2-1-3	業務データを扱うだけでは不十分になってきている	12
2-1-4	RDBだけでは立ちいかない	14
2-1-5	RDBが適さない身近なエピソード	16
2-1-6	RDB以外を知ることが重要	24
2-1-7	これまでのまとめ	26
2-2	NoSQLとは	26
2-2-1	NoSQLはバズワード	27
2-2-2	KVS,ドキュメントDB,グラフDBの違い	27
2-2-3	NoSQLがバズワードになった背景	29
2-3	NoSQLにすると嬉しいこと・辛いこと	30
2-3-1	アプリケーション開発者にとって	31
2-3-2	データベース管理者にとって	37
2-3-3	マネージャや経営者にとって	40
2-4	よくあるNoSQLの勘違い	44
2-4-1	「バッチが高速になる」は勘違い	44
2-4-2	「トランザクションが高速になる」は勘違い	45

- 2-4-3 「ビッグデータを分析できる」は勘違い …… 46
- 2-4-4 「非構造データが効率的に扱える」は正確ではない …… 46
- 2-4-5 「RDBから置き換えると速くなる」は正確ではない …… 47
- 2-4-6 「オープンソースしかない」は昔の話 …… 48
- 2-4-7 「スキーマがない」は昔の話 …… 49
- 2-4-8 「SQLが使えない」は昔の話 …… 49

第3章 データベースの中のNoSQLの位置づけ

- 3-1 データベースを分類する2つの軸 …… 52
 - 3-1-1 重視する性能による分類軸 …… 52
 - 3-1-2 性能拡張モデルによる分類軸 …… 54
- 3-2 データベースの4つのエリア …… 56
 - 3-2-1 RDB (OLTP) …… 57
 - 3-2-2 RDB (DWH) …… 57
 - 3-2-3 Hadoop (HDFS+MapReduce) …… 60
 - 3-2-4 KVS …… 61
 - 3-2-5 ドキュメントDB …… 62
 - 3-2-6 グラフDB …… 63
- 3-3 RDB (OLTP) と KVS/DocDB の違い …… 65
 - 3-3-1 RDB (OLTP) は強い整合性 …… 66
 - 3-3-2 強い整合性を保ったまま性能をスケールするのは困難 …… 67
 - 3-3-3 KVS/DocDB では3つの工夫で性能をスケールさせている …… 69
 - 3-3-4 CAPの定理 …… 74
- 3-4 Hadoop と KVS/DocDB の違い …… 78
 - 3-4-1 Hadoop (HDFS+MapReduce) の動作 …… 78
 - 3-4-2 Hadoop と比較したときの KVS/DocDB の動作 …… 81
 - 3-4-3 4つのデータベースの比較表 …… 82
- 3-5 4つのエリアを超えて成長するデータベース達 …… 83
 - 3-5-1 応答が速くSQLを使えるHadoop …… 83
 - 3-5-2 集計できるKVS/DocDB …… 84
 - 3-5-3 SQLを使えるNoSQL …… 84
 - 3-5-4 JSONを格納するRDB (OLTP) …… 84
 - 3-5-5 スケールアウトするRDB (DWH) …… 85

- 3-5-6 オペレーションも分析もできるRDB(DWH) 86
- 3-5-7 まとめ 86

第4章 データモデルごとの NoSQLプロダクト紹介

- 4-1 データモデルの種類 92
 - 4-1-1 データモデルの説明 92
 - 4-1-2 複雑度比較 98
 - 4-1-3 データ間の関連度とスケーラビリティ比較 99
- 4-2 データモデル毎のプロダクトの紹介 101
 - 4-2-1 キーバリュモデルを採用するプロダクト 103
 - 4-2-2 ワイドカラムモデルを採用するプロダクト 105
 - 4-2-3 ドキュメントモデルを採用するプロダクト 107
 - 4-2-4 グラフモデルを採用するプロダクト 111

第5章 NoSQLの代表プロダクト紹介を 読む前に

- 5-1 紹介するプロダクトの選定基準 114
 - 5-1-1 データモデルの中で広く使われていること 114
 - 5-1-2 国内のサポート体制が整っているもの 115
- 5-2 プロダクト紹介の観点 115

第6章 Redis

- 6-1 概要 120
- 6-2 データモデル 121
 - 6-2-1 データ型 122
 - 6-2-2 永続化 127
- 6-3 API 131
 - 6-3-1 クエリの実行例 131
 - 6-3-2 利用できるクエリ 133
 - 6-3-3 アプリケーションからの通信手段 135

6-3-4	部分的トランザクション	137
6-4	性能拡張	139
6-4-1	Redis Clusterのシャーディング	139
6-4-2	クエリの分散	140
6-4-3	リシャーディング	141
6-4-4	Redis Clusterのレプリケーションによる読み取り負荷分散	142
6-4-5	ハッシュタグを用いたRedis Cluster上での複数キー操作	142
6-5	高可用	143
6-5-1	レプリケーションによる可用性向上	143
6-5-2	フェイルオーバー	144
6-5-3	非同期レプリケーションによるデータのロス	144
6-5-4	永続化していないマスターのリカバリ時の注意点	145
6-6	運用	146
6-6-1	バックアップ	146
6-6-2	監視	147
6-6-3	稼働統計	147
6-6-4	バージョンアップ	148
6-7	セキュリティ	148
6-7-1	パスワード認証	148
6-7-2	コマンドのリネーム・無効化	149
6-7-3	暗号化	150
6-8	出来ないこと	150
6-8-1	条件検索や集計などの処理が存在しない	150
6-8-2	ロールバック機能が存在しない	150
6-8-3	厳密な一貫性の担保	151
6-8-4	セキュリティ機能に乏しい	151
6-9	主なバージョンと特徴	151
6-10	国内のサポート体制	152
6-11	ライセンス体系	153
6-12	効果的な学習方法	153
6-13	その他	154
6-13-1	Redis Clusterの詳細	154
6-13-2	シャーディングとレプリケーションを組み合わせ	156

第7章 Cassandra

7-1 概要	160
7-1-1 Cassandraの特徴	163
7-1-2 Cassandraのユースケース	164
7-1-3 OSS版と商用版	165
7-2 データモデル	165
7-2-1 Cassandraオブジェクト	167
7-3 API	168
7-3-1 Cassandra Query Language (CQL)	168
7-3-2 Cassandraのドライバ/コネクタ	170
7-3-3 軽量トランザクション	171
7-3-4 バッチ分析	171
7-3-5 外部Hadoopのサポート	172
7-3-6 データの検索	172
7-3-7 分析と検索に対応したワークロードの管理	173
7-3-8 高速な書き込みと読み込み	174
7-4 性能拡張	176
7-4-1 クラスターアーキテクチャの概要	176
7-4-2 Cassandraのクラスター、データセンタ、ノード	178
7-4-3 データの分散	179
7-4-4 クエリの分散	180
7-5 高可用	180
7-5-1 レプリケーションの基礎	181
7-5-2 マルチデータセンタとクラウドという選択	182
7-5-3 レプリケーション係数とクエリの整合性レベル	184
7-6 運用	186
7-6-1 クエリツール、管理ツール	186
7-6-2 バックアップとリカバリ	187
7-6-3 パフォーマンス管理	189
7-6-4 データの移行	195
7-7 セキュリティ	196
7-7-1 認証	196
7-7-2 権限管理	197
7-7-3 暗号化	197

7-7-4	データの監査	198
7-8	出来ないこと	199
7-9	主なバージョンと特徴	199
7-10	国内のサポート体制	200
7-11	ライセンス体系	201
7-12	効果的な学習方法	201
7-12-1	Cassandraの技術マニュアル	201
7-12-2	Cassandra トレーニング	202
7-12-3	Cassandraの技術情報、不具合情報	203

第8章 HBase

8-1	概要	206
8-2	データモデル	207
8-3	API	209
8-3-1	テーブルの作成	210
8-3-2	データの格納	212
8-3-3	データの参照	213
8-3-4	データの更新	214
8-3-5	データの削除	215
8-3-6	テーブルの削除	215
8-3-7	APIについての補足	217
8-3-8	部分的トランザクション	217
8-4	性能拡張	219
8-4-1	HBase クラスタのコンポーネント	219
8-4-2	データの分散とクエリの分散	221
8-5	高可用	222
8-6	運用	224
8-6-1	データのバックアップとリストア	224
8-6-2	監視と稼働統計	226
8-6-3	バージョンアップ	228
8-7	セキュリティ	228
8-7-1	データへのアクセス制御	228

8-7-2	操作記録	229
8-7-3	暗号化	229
8-8	出来ないこと	230
8-9	主なバージョンと特徴	231
8-10	国内のサポート体制	232
8-11	ライセンス体系	232
8-12	効果的な学習方法	232

第9章 Amazon DynamoDB

9-1	概要	236
9-1-1	概要	236
9-1-2	特徴	237
9-2	データモデル	238
9-2-1	アトリビュートのデータ型	239
9-2-2	DynamoDB JSON	240
9-2-3	キー	241
9-2-4	インデックス	242
9-2-5	DynamoDB Stream	243
9-3	API	244
9-3-1	APIとCRUD	244
9-3-2	アプリケーションから利用する	247
9-3-3	低レベルAPIと高レベルAPI	247
9-4	性能拡張	250
9-4-1	結果整合性	250
9-4-2	スループット、キャパシティーユニット	251
9-4-3	パーティション	252
9-5	高可用	253
9-5-1	レプリケーション、フェイルオーバー	253
9-5-2	クロスリージョンレプリケーション	254
9-6	運用	255
9-6-1	監視	255
9-6-2	バックアップ	259

9-7 セキュリティ	260
9-7-1 セキュリティの考え方	260
9-7-2 通信暗号化	260
9-7-3 アクセスコントロール	261
9-7-4 監査	262
9-8 出来ないこと	264
9-9 国内のサポート体制	264
9-10 効果的な学習方法	265
9-11 その他	266
9-11-1 バージョンと利用料	266

第10章 MongoDB

10-1 概要	270
10-1-1 MongoDBの主な特徴	271
10-2 データモデル	272
10-2-1 格納するデータの階層	272
10-2-2 格納できるデータ型	274
10-2-3 JSONのスキーマの事前チェック(ドキュメントバリデーション)	274
10-3 API	275
10-3-1 Mongo クエリ言語の概要	275
10-3-2 CRUDのサンプル	276
10-3-3 CRUDの特徴	278
10-3-4 集計	279
10-3-5 アプリケーションからの使い方	281
10-3-6 インデックス	282
10-4 性能拡張	284
10-4-1 シャーディングによる性能拡張	284
10-4-2 セカンダリ読み込みによる読み込み負荷分散	287
10-5 高可用	287
10-5-1 レプリケーションの概要	288
10-5-2 フェイルオーバー	290
10-5-3 セカンダリの種類	290
10-5-4 書き込み回数指定クエリ	291

10-6 運用	292
10-6-1 バックアップ	292
10-6-2 ヒューマンエラー対策(遅延レプリケーション)	293
10-6-3 監視・稼働統計	293
10-6-4 バージョンアップ	294
10-6-5 MongoDB Ops Manager	295
10-7 セキュリティ	296
10-7-1 通信暗号化	297
10-7-2 データ暗号化	297
10-7-3 アクセスコントロール	297
10-7-4 監査	297
10-8 出来ないこと	298
10-9 主なバージョンと特徴	299
10-9-1 バージョンのつけ方	299
10-9-2 主なバージョンとその機能	300
10-10 国内のサポート体制	300
10-11 ライセンス体系	301
10-12 効果的な学習方法	302
10-13 その他	302
10-13-1 便利な機能一覧	302
第11章 Couchbase	
11-1 概要	306
11-1-1 Couchbaseという言葉	306
11-1-2 Couchbase Serverの主な特徴	307
11-2 データモデル	310
11-3 API	311
11-3-1 データへのアクセス方法	311
11-3-2 クライアントライブラリの各API実行サンプル	320
11-4 性能拡張	324
11-4-1 データ分散	325
11-4-2 データアクセスの分散	326

11-4-3	リバランスによる無停止でのクラスタ伸縮	328
11-5	高可用	329
11-5-1	クラスタ内レプリケーション	329
11-5-2	物理構成を意識したレプリケーション	330
11-5-3	複数クラスタ間のレプリケーション(XDCR)	331
11-6	運用	334
11-6-1	バックアップ	334
11-6-2	監視・稼働統計	335
11-6-3	バージョンアップ	336
11-7	セキュリティ	337
11-7-1	通信暗号化	337
11-7-2	管理者ユーザ、LDAP連携	338
11-7-3	監査ログ	338
11-8	出来ないこと	339
11-9	主なバージョンと特徴	340
11-9-1	バージョンの振り方	340
11-9-2	主なバージョンとその機能	341
11-10	国内のサポート体制	341
11-11	ライセンス体系	342
11-12	効果的な学習方法	343
11-13	その他	344
11-13-1	モバイルソリューション	344
11-13-2	便利な機能	346
11-13-3	ロードマップ	348
11-13-4	Couchbase Serverアーキテクチャ詳細	349

第12章 Microsoft Azure DocumentDB

12-1	概要	358
12-1-1	Microsoft Azure	358
12-1-2	Microsoft Azureのデータベース関連のサービス	360
12-1-3	Microsoft Azure DocumentDB	362

12-1-4	DocumentDBを使ってみよう	365
12-1-5	Azureの他の機能との連携	368
12-2	データモデル	369
12-2-1	リソースモデル	369
12-2-2	データモデル	371
12-3	API	374
12-3-1	REST API	374
12-3-2	クライアントSDK	378
12-3-3	インデックス	379
12-3-4	SQLクエリ (DocumentDB SQL)	383
12-3-5	ストアドプロシージャ、トリガ、UDF (ユーザ定義関数)、 トランザクション	389
12-3-6	文字列の検索	393
12-4	高可用	394
12-4-1	高可用性のためのアーキテクチャ	394
12-4-2	整合性レベルとレプリケーション	396
12-4-3	クライアントからの接続	401
12-5	性能拡張	403
12-5-1	コレクション	403
12-5-2	パーティション分割	405
12-5-3	パーティション分割に対応したアプリケーションの開発	407
12-5-4	.NET SDK を使用したパーティション分割	409
12-6	運用	412
12-6-1	管理と監視	412
12-6-2	バックアップ/リストア	413
12-7	セキュリティ	414
12-8	出来ない事	416
12-9	国内のサポート体制	417
12-10	効果的な学習方法	417

第13章 Neo4j

13-1 概要	420
13-1-1 グラフDBに向いている処理	421
13-1-2 グラフDBに向いていない処理	422
13-2 データモデル	422
13-2-1 グラフを構成する要素	423
13-2-2 グラフデータの格納形式	425
13-2-3 グラフデータモデル	427
13-3 API	430
13-3-1 Cypherクエリ	430
13-3-2 アプリケーションからのアクセス方法	441
13-4 性能拡張	443
13-4-1 HA クラスタによる処理性能向上	444
13-4-2 キャッシュシャーディングによる処理性能向上	444
13-5 高可用	445
13-5-1 HA クラスタのアーキテクチャ	445
13-5-2 システム構成	446
13-6 運用	449
13-6-1 バックアップ	449
13-6-2 リストア	449
13-6-3 バルクロード	450
13-6-4 監視	450
13-6-5 ログ出力	452
13-6-6 稼働統計	452
13-7 セキュリティ	452
13-8 出来ないこと	453
13-9 国内のサポート体制	454
13-10 主要バージョンと特徴	455
13-10-1 Neo4jのエディション間の比較	455
13-10-2 ライセンス体系	456
13-11 効果的な学習方法	457
13-11-1 公式ドキュメント	458

- 13-11-2 ユーザ会 …… 458
- 13-11-3 書籍 …… 459
- 13-11-4 他の日本語の資料 …… 459

第14章 想定されるNoSQLのユースケース

- 14-1 キャッシュ (Redis) …… 462**
 - 14-1-1 RDBのスケールアップ・スケールアウトによる対処 …… 462
 - 14-1-2 RedisによるWebアプリケーションキャッシュ …… 463
 - 14-1-3 まとめ …… 465
- 14-2 IoT (モノのインターネット) 基盤 (Cassandra) …… 465**
 - 14-2-1 RDBを用いた場合の課題 …… 465
 - 14-2-2 NoSQLによる課題解決 …… 467
- 14-3 メッセージ基盤 (Cassandra) …… 471**
 - 14-3-1 なぜNoSQL向きなのか …… 472
 - 14-3-2 NoSQLを用いたメッセージ基盤の具体例 …… 474
 - 14-3-3 まとめ …… 477
- 14-4 Hadoop連携 (HBase) …… 478**
 - 14-4-1 RDBの課題 …… 478
 - 14-4-2 HBaseによる解決 …… 479
 - 14-4-3 HBaseとMapReduceアプリケーションの連携 …… 480
- 14-5 モバイルアプリケーションに代表されるアプリケーションでの利用 (DynamoDB) …… 483**
- 14-6 AWSサービスとの連動性を意識した利用 (DynamoDB) …… 484**
- 14-7 ログ格納システム (MongoDB) …… 485**
 - 14-7-1 RDBだと大変 …… 486
 - 14-7-2 MongoDBだと楽 …… 487
 - 14-7-3 まとめ …… 489
- 14-8 ECサイトのカタログ管理 (MongoDB) …… 490**
 - 14-8-1 RDBだと大変 …… 490
 - 14-8-2 MongoDBだと楽 …… 491
 - 14-8-3 まとめ …… 492
- 14-9 高速開発 (MongoDB) …… 493**

14-9-1	RDB だと大変	493
14-9-2	MongoDB だと楽	493
14-9-3	Web フレームワークに組み込まれる MongoDB	495
14-9-4	まとめ	495
14-10	業界横断型アプリ (MongoDB)	496
14-10-1	RDB だと大変	496
14-10-2	MongoDB だと楽	497
14-10-3	まとめ	498
14-11	Web アプリ (ユーザプロフィール/セッションストレージ) (Couchbase)	498
14-11-1	RDB で実現しようとした時の課題	499
14-11-2	Couchbase による解決	499
14-12	Web アプリ (オムニチャネル/パーソナライズ) (Couchbase)	500
14-13	データベースのグローバル展開/ディザスタリカバリ (Couchbase)	502
14-14	モバイルとサーバのデータ同期 (Couchbase)	503
14-15	リアルタイム詐欺摘発システム (Neo4j)	505
14-15-1	概要	505
14-15-2	Neo4j による解決	506
14-15-3	RDB では実現が難しい	507
14-16	適材人材の検索システム (Neo4j)	508
14-16-1	概要	508
14-16-2	Neo4j による解決	508
14-16-3	RDB では実現が困難	511
14-17	経路計算システム (Neo4j)	511
14-17-1	概要	511
14-17-2	Neo4j による解決	512
14-17-3	RDB では非効率	513

第15章 NoSQLの選び方

15-1	データ処理の課題を見極める	517
15-1-1	NoSQL で解決するのが最適な課題はどれか?	518

15-1-2	NoSQLでは解決できないRDB (OLTP) の課題	520
15-1-3	NoSQLで解決するかわからないRDB (OLTP) の課題	521
15-1-4	NoSQLで解決が期待できるRDB (OLTP) の課題	524
15-2	高い処理性能を出すためのNoSQLの選び方	525
15-2-1	小規模なキーバリュースタイルならRedis	525
15-2-2	マルチデータセンタでどこでも書き込めるようにしたいなら Cassandra か Couchbase	526
15-2-3	MongoDBは柔軟なデータ分散やレンジ指定クエリを速くしたい場合	528
15-2-4	Hadoopと一緒にHBase	528
15-2-5	クラウド上でのスケーラビリティ獲得ならばDynamoDBや Microsoft Azure DocumentDBを検討	529
15-3	半構造化データを処理しやすいNoSQLの選び方	530
15-3-1	ドキュメントDBはどれを選ぶべきか	531
15-4	その他の選定の観点	536
15-4-1	可用性の高いNoSQLの選び方	536
15-4-2	セキュリティの高いNoSQLの選び方	537
15-5	本書にないNoSQLを選ぶ時のポイント	537
15-5-1	ありがちな悪い文句に踊らされない	538
15-5-2	性能比較を当てにしない	539
15-5-3	最新ドキュメントを見る	541
Index	索引	542
Profile	著者プロフィール	547

第 1 章

前提

NoSQL

1-1

この本で伝えたいこと

本書で伝えたいことは3つあります。

- NoSQL というバズワードの実態を理解
- エンタープライズ視点の NoSQL 活用方法
- NoSQL の最新情報

1-1-1

NoSQL というバズワードの実態を理解

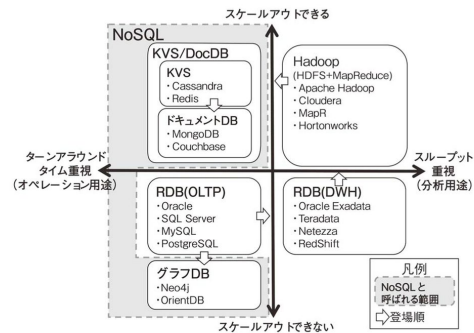
世の中のデータベースのはほぼ全ては RDB です。読者の方も RDB だけを使ってきた人がほとんどでしょう。しかし、近年 NoSQL と呼ばれる RDB ではないデータベースが人気になってきています。昔は、NoSQL といえばオープンソースソフトウェアしかなく、最先端の技術というイメージでしたが、近年では大手ベンダから NoSQL の商用製品やクラウドサービスが続々と登場しており、いまや NoSQL は当たり前の技術の一つになってきました。

しかし NoSQL とはなんなのでしょうか？それを正しく説明するのはかなり困難です。言葉通り「RDB 以外のデータベース」と思ってしまうと、大きな誤解になります。

NoSQL はバズワードです。NoSQL に明確な定義はなく NoSQL と呼ばれるデータベースの中でも各々性質が大きく異なります。

本書では、NoSQL という言葉はできるだけ使わずに、具体的な KVS、ドキュメント DB、グラフ DB といったデータベース種別で分類します。そし

に関連するHadoopやDWHも絡めて、図1-1のように整理しています。この図を見てわかる通り、NoSQLはRDB以外の全てのデータベースを含んでいるわけではなく、かといってある特定の性質があるわけでもないのです。これがNoSQLがバズワードであることを物語っています。この図については本書で詳しく説明しますので、これを理解することでNoSQLの実態がはっきりと浮かび上がるでしょう。



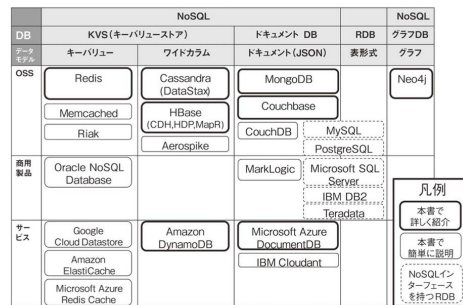
● 図1-1 NoSQLと呼ばれるデータベースが所属する範囲

[1-1-2]

エンタープライズ視点のNoSQL活用方法

NoSQLに分類されるデータベースは数多く存在します。オープンソースだけでなく商用製品やクラウドサービスでもNoSQLと名のつくプロダクトが数多く登場しています。また、既存のデータベース製品もNoSQLのインターフェースを設けるようになってきました。NoSQLが氾濫していると言ってもいいでしょう。

そこで本書では、NoSQLの中からエンタープライズの本番運用に耐えるプロダクトに絞り込んで詳細に紹介します。一部の人が使っていないマニアックなNoSQLや、開発体制やサポートが十分でないNoSQLは、エンタープライズでは利用できませんので紹介からは外します。広く普及して品質が安定しており、サポートが受けられ、エンタープライズでの事例があるNoSQLだけに絞り込んで紹介します。具体的には、キーバリューストアからはRedis、Cassandra、HBase、Amazon DynamoDB、ドキュメントデータベースからはMongoDB、Couchbase、Microsoft Azure DocumentDB、グラフデータベースからはNeo4jを紹介します(図1-2)。



● 図1-2 NoSQL一覧と紹介するプロダクト

各プロダクトの説明は、プロダクトを選定する立場の読者を意識して、アーキテクチャや処理方式といった高い目線から説明します。よって、実作業で必要となるインストール手順やサンプルコードなどの詳細な情報は本書では省略します。プロダクトを選定するのに最低限必要な知識を提供する本だと思ってください。

また、各プロダクトを横並びで比較しやすいように、各プロダクトの説明では観点をそろえています。具体的には、以下の観点でプロダクトを説

明します。

- 概要
- 機能
 - ・ データモデル
 - ・ API
- 非機能
 - ・ 性能拡張
 - ・ 高可用
 - ・ 運用
 - ・ セキュリティ
- その他有用な情報
 - ・ 出来ないこと
 - ・ 主なバージョンと特徴
 - ・ 国内のサポート体制
 - ・ ライセンス体系
 - ・ 効果的な学習方法

加えて、NoSQLの想定されるユースケースを数多く紹介します。ユースケースは「商品カタログ管理」や「リアルタイム詐欺検出」など、可能な限りアプリケーションに近い視点で紹介します。この説明を読んでもいただくことによりどのプロダクトがどのユースケースに向いているか理解できるとともに、NoSQLを利用するイメージがアップするでしょう。

以上のように、本書を読んでもいただければ、読者が直面している問題を解決するのに最適なNoSQLプロダクトを選択して活用できるようになるでしょう。

[1-1-3]**NoSQLの最新情報**

現在、NoSQLは非常に活発に開発されており、多くのプロダクトは年に数回バージョンアップしています。また変化も激しく、1年前はサポートされていなかった機能が今ではサポートされていることは多いです。

本書では、NoSQLの代表プロダクトについて、それぞれのスペシャリストを集め最新の情報を記載しています。他のNoSQLの日本語書籍もありますが、それらとの違いは、本書が2015年12月時点の最新情報を集めた書籍であるという点です。具体的にはRedis 3.0、HBase 1.1、Cassandra 3.0、MongoDB 3.2、Couchbase 4.1、Neo4j 2.3、Amazon DynamoDB(2015年12月時点)、Microsoft Azure DocumentDB(2015年12月時点)であり、これらの日本語の正確な情報というのはあまり多くありません。

また、各プロダクトの最新情報については、各プロダクトをエンタープライズでサポートしているスペシャリストの方々に執筆してもらっています。そのため、情報は正確で的確です。インターネットのブログの情報や英語のドキュメントを読むよりも遥かに効率的に最新情報をキャッチアップできるでしょう。

1-2**想定する読者**

本書が想定する読者は、RDBだけでシステムを開発しており、NoSQLについては初心者の方を対象としています。RDBの概念や基本用語については理解していることを前提としていますので、データベースを全く触ったことのない読者には向いていません。

また、NoSQLをエンタープライズの本番環境で運用しようと考えている方を前提とします。そのため、純粋な技術以外の学習コストや国内保守体制といった内容についても言及します。エンタープライズ以外の領域でNoSQLを学習する方にとっては少し的外れな内容になってしまいますので、その点はご了承ください。

最後に、想定している読者の職域は、技術を選定する立場にある方です。現場のアプリケーション開発者やデータベース管理者の方に対しては、ある程度有益な情報はあると思いますが、具体的な構築手順や操作手順など細かい内容は掲載されていません。

1-3

本書の構成

2章「イントロダクション」では、NoSQLの必要性や、NoSQL自体の簡単な説明、NoSQLを使うと嬉しいこと、NoSQLのよくある勘違い、など気軽に読める内容になっています。

3章以降は、体系立てた説明をしていきます。

3章「データベースの中のNoSQLの位置づけ」では、データベース全体を俯瞰した際にNoSQLがRDB (OLTP)、RDB (DWH)、Hadoopとどのように違うか説明します。また、NoSQLというあいまいな言葉をKVS、ドキュメントDB、グラフDBの三つに分けてそれぞれの特徴を説明します。

4章「データモデルごとのNoSQLプロダクト紹介」ではNoSQLをデータモデルで分類し、分類の中の具体的なプロダクトを紹介します。

5章「NoSQLの代表プロダクト紹介を読む前に」では代表プロダクトの詳細な説明をする上で、先に読んでおいてほしい点について記載していま

す。そして、6章「Redis」以降から13章「Neo4j」までは具体的なプロダクトの詳細な説明になります。

14章「想定されるNoSQLのユースケース」ではNoSQLの想定されるユースケースを多数紹介します。これにより、NoSQLを利用するイメージをより具体化してもらえましょう。

最後に、15章「NoSQLの選び方」では、RDBの課題を整理した上で、それがNoSQLで解決できるのか、解決できるとしたらどのNoSQLを選ぶべきなのかを説明します。

無料サンプルはお楽しみいただけましたか？

この続きをお読みになるには...

[今すぐ購入](#)

または

[Kindleストアで本の詳細をチェック](#)
