

MongoDB 3.4 New Feature

InBum Kim

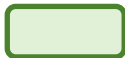
MongoDB Korea

with



Lineup

Before Started



3.4 inside



After 3.4



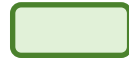
Suggestion



Q&A



Before Started



MongoDB?

Document Oriented

key-value 로 구성되는 document는 MongoDB의 기본 unit

Sharding / Replication.

제품 초기부터 언급된 MongoDB의 대표적인 기능

Dynamic schema

schemaless 로 주로 알려져 있지만 정확히는 Dynamic schema에 가까움

Latest release

최신 release는 3.4.2 ('17.2.1 기준)

MongoDB History

2.2 Aggregation TTL Collection

2.4 Hash-Based Sharding Kerberos

2.6 Index Intersection Text Search Field-Level Redaction LDAP & x509 Auditing

3.0 Doc-Level Concurrency Compression Storage Engine API ≤50 replicas Auditing ↑ Ops Manager

3.2 Document Validation \$lookup Fast Failover Aggregation ↑ In-Memory Storage Engine BI Connector
MongoDB Compass APM Integration Profiler Visualization

3.4 Intra-cluster compression Views Log Redaction Linearizable Reads Graph Processing Decimal Collations
Faceted Navigation Spark Connector ↑ Aggregation ↑ BI Connector ↑ Compass ↑

3.6 ?

Release update 별 약 6개월~1년 정도의 시간이 소요되었으며,
특이하게 release 2.8은 존재하지 않는다.
(2.6 → 3.0으로 바로 update)

WiredTiger 엔진이 3.0부터 도입되어 3.2부터는 default로 설정됨

3.4 inside



3.4 Main

Multi-model done right.

다양한 model이 하나의 데이터베이스에서 가능해짐

: document, graph, key value, and search with faceted navigation.

Mission-critical apps.

강화된 security, platform support 범위 증가는 mission-critical deployments를
가능하게 함

Modernized tooling.

operation, 분석 등을 위한 tool 들의 진화

Graph Processing

- **\$graphLookup**
graph 또는 tree 구조 데이터의 처리를 가능하게 만들어 줌
- 데이터 간의 간접적인 관계를 파악하기에 좋음
 - 추천 엔진, social networks, etc.
- Single collection & across multi collection 상에서 모두 recursive한 쿼리 가능
 - 추천 엔진, social networks, etc.

Graph Processing

```
{
  "_id" : 1,
  "name" : "Tanya Jordan",
  "friends" : [ "Shirley Soto", "Terry Hawkins", "Carole Hale" ],
  "hobbies" : [ "tennis", "unicycling", "golf" ]
}
{
  "_id" : 2,
  "name" : "Carole Hale",
  "friends" : [ "Joseph Dennis", "Tanya Jordan", "Terry Hawkins" ],
  "hobbies" : [ "archery", "golf", "woodworking" ]
}
{
  "_id" : 3,
  "name" : "Terry Hawkins",
  "friends" : [ "Tanya Jordan", "Carole Hale", "Angelo Ward" ],
  "hobbies" : [ "knitting", "frisbee" ]
}
{
  "_id" : 4,
  "name" : "Joseph Dennis",
  "friends" : [ "Angelo Ward", "Carole Hale" ],
  "hobbies" : [ "tennis", "golf", "topiary" ]
}
{
  "_id" : 5,
  "name" : "Angelo Ward",
  "friends" : [ "Terry Hawkins", "Shirley Soto", "Joseph Dennis" ],
  "hobbies" : [ "travel", "ceramics", "golf" ]
}
{
  "_id" : 6,
  "name" : "Shirley Soto",
  "friends" : [ "Angelo Ward", "Tanya Jordan", "Carole Hale" ],
```

Graph Processing

```
db.people.aggregate( [
  { $match: { "name": "Tanya Jordan" } },
  { $graphLookup: {
    from: "people",
    startWith: "$friends",
    connectFromField: "friends",
    connectToField: "name",
    as: "golfers",
    restrictSearchWithMatch: { "hobbies" : "golf"
  }
  },
  { $project: {
    "name": 1,
    "friends": 1,
    "connections who play golf": "$golfers.name"
  }
  }
] )
```

Graph Processing

```
{
  "_id" : 1,
  "name" : "Tanya Jordan",
  "friends" : [
    "Shirley Soto",
    "Terry Hawkins",
    "Carole Hale"
  ],
  "connections who play golf" : [
    "Joseph Dennis",
    "Tanya Jordan",
    "Angelo Ward",
    "Carole Hale"
  ]
}
```

Faceted Navigation

- 관련이 깊은 카테고리 데이터 묶어 사용자가 검색 조건을 제시하기 전에 선택할 수 있는 범위를 좁혀주는 기능
 - 검색, 분석 애플리케이션에서 사용됨
- 검색 결과가 너무 많아 특정 범위로 범위를 한정 지어야 할 때 사용자가 별도의 검색을 하지 않아도 자연스럽게 원하는 방향으로 정보를 얻을 수 있게 해준다
 - 검색, 분석 애플리케이션에서 사용됨
- **Facet** - 사물의 측면과 단면
Facet Navigation – 사물의 측면과 단면을 고려하여 고객에게 방향성을 제시하는 내비게이션

Faceted Navigation

```
db.artwork.aggregate( [
  {
    $facet: {
      "categorizedByTags": [
        { $unwind: "$tags" },
        { $sortByCount: "$tags" }
      ],
      "categorizedByPrice": [
        // Filter out documents without a price e.g., _id: 7
        { $match: { price: { $exists: 1 } } },
        {
          $bucket: {
            groupBy: "$price",
            boundaries: [ 0, 150, 200, 300, 400 ],
            default: "Other",
            ...
          }
        }
      ],
      "categorizedByYears(Auto)": [
        {
          $bucketAuto: {
            groupBy: "$year",
            buckets: 4
          }
        }
      ]
    }
  }
]
```

Faceted Navigation

```
{
  "categorizedByYears(Auto)" : [
    // First bucket includes the document without a year, e.g., _id: 4
    { "_id" : { "min" : null, "max" : 1902 }, "count" : 2 },
    { "_id" : { "min" : 1902, "max" : 1918 }, "count" : 2 },
    { "_id" : { "min" : 1918, "max" : 1926 }, "count" : 2 },
    { "_id" : { "min" : 1926, "max" : 1931 }, "count" : 2 }
  ]
  ...
  "categorizedByTags" : [
    { "_id" : "painting", "count" : 6 },
    { "_id" : "oil", "count" : 4 },
    { "_id" : "Expressionism", "count" : 3 },
    { "_id" : "Surrealism", "count" : 2 },
    { "_id" : "abstract", "count" : 2 },
    { "_id" : "woodblock", "count" : 1 },
    { "_id" : "woodcut", "count" : 1 },
    { "_id" : "ukiyo-e", "count" : 1 },
    { "_id" : "satire", "count" : 1 },
    { "_id" : "caricature", "count" : 1 }
  ]
}
```

Advanced Analytics

- **MongoDB Connector for BI**

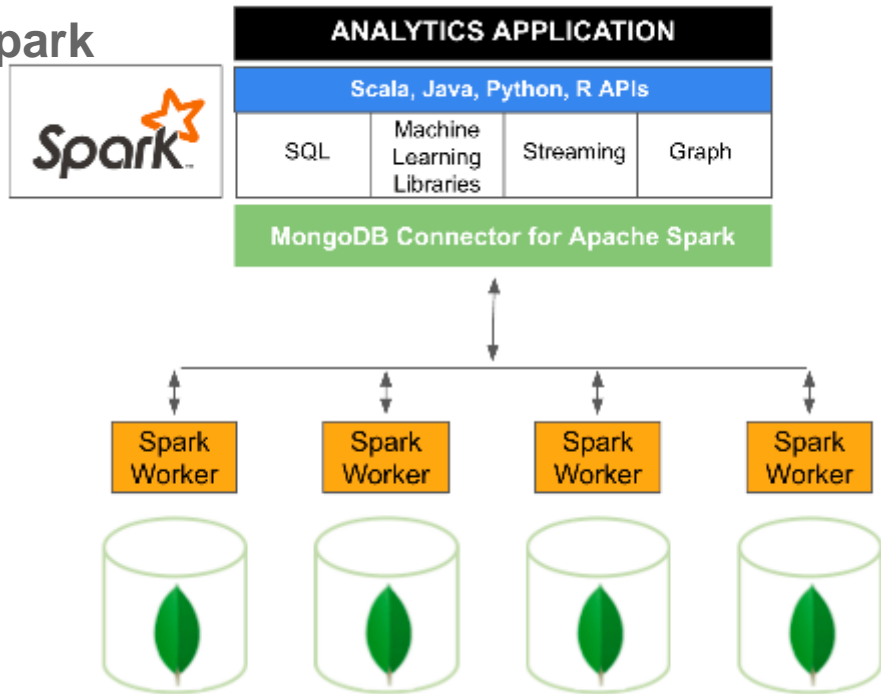
- SQL 기반의 BI 도구로서 visualizations과 분석을 위한 툴
- 2015년 11월에 처음으로 소개되어 꾸준히 업데이트 중



Advanced Analytics

- **MongoDB Connector for Apache Spark**

- Native Scala 커넥터
- Exposes all Spark APIs & libraries (including Scala, Java, Python, and R)
- Efficient data filtering with predicate pushdown, secondary indexes, & in-database aggregations
- Spark 2.0 까지 지원 가능



And then...

- **Collation**
 - 100가지 이상의 언어를 이용한 Text 비교 및 sorting 가능
 - Over 2x as many as offered by most RDBMs
- **Sharding**
 - Range Sharding
 - Hash Sharding
 - Zone Sharding
- **Geographically Distributed Clusters**
- **Ops Manager, MongoDB Compass**
- **Read Only Views**

After 3.4



Release Pattern

Release 패턴(1)

v3.0 이전은 기존에 지적 받았던 문제점을 고치는 등의 수세적인 release update

v3.0 이후에는 장점을 극대화하는 다소 공세적인 release update

Release 패턴(2)

Public version 과 Enterprise version의 격차가 조금씩 발생하기 시작함

After 3.4

Multi & Multi

다양한 Open Source와의 연동

다양한 Platform 지원

다양한 도구들의 등장

Ops Manager, MongoDB Atlas, MongoDB Compass, ... + ∂

Another Empire?

자체 Platform으로의 진화 과정

다양한 Open Source & Platform과의 연계를 통해 점점 범용성을 갖추어 감

기존 제품군을 활용한 자체 생태계 구축

MongoDB + 다양한 Tool + ∂

Suggestion



3.2 or 3.4 ?

Release 선택의 어려움

일반적으로 Open Source는 가급적 빠르게 최신 release에 맞추는 것을 권고하지만, 항상 그렇지는 않으며 상황에 따라 다른 선택이 가능

선택의 기준은?

최신 API(기능)를 빠르게 활용해야 하는지,
정해진 몇 가지 기능 위주로 서비스가 운영이 될 수 있는지,
빠른 도입 + 빠른 교정 vs 단계별 도입 + 안정적 update

Compatibility Check

잘 드러나지는 않지만, 반드시 확인이 필요한 **compatibility check**

2.x → 3.x 때 급격한 변화가 발생했으며, 이후 release부터 주요 check point

주로 바뀌는 부분

sharding & replication

deprecated operation

security

storage engine

synergy

연계되는 생태계의 증가

초기 Hadoop을 넘어 다양한 Open Source 생태계와 연계되고 있는 중.

Integrated with AWS, MS, IBM etc.

다양해지는 조합

MongoDB + Hadoop, MongoDB + Spark, MongoDB + Redis, ...

MEAN (MongoDB, Express, AngularJS, Node.js) – full stack development toolkit

suggestion

서비스의 역할, 범위, 한계를 명확히 할 것

서비스 유형에 따라 항상 최신 **release**를 유지할 필요가 없을 수도 있음

번거로워도 **compatibility check**는 확실하게 할 것

사소한 변화로 거대한 **side effect**를 만들어 내기 쉬운 부분

다양한 **Open Source**와 연동

다양한 생태계와의 연동은 생각지 못한 상승 효과를 불러 일으킬 수 있음

Q & A



MongoDB Korea

MongoDB Korea

- <https://www.facebook.com/groups/krmug/>
- inbum85@gmail.com
- [whoever, whenever, whatever !](#)