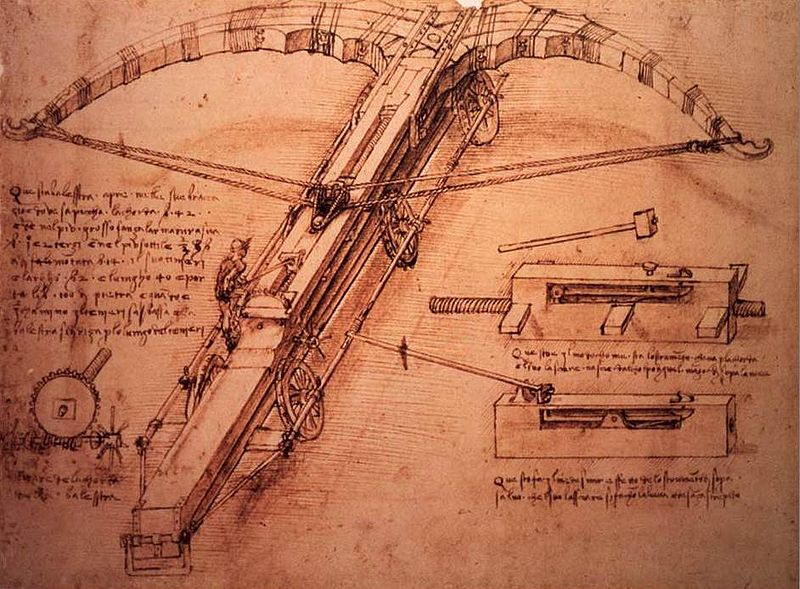
**Casssandra運用**



**XM-Megha-1.2.0**

**July25, 2018**

**Document Revision 1.0**



**改版履歴**

| 版数 | 発行年月日 | 改版内容 | 備考 |
| --- | --- | --- | --- |
| 1.0 | 2018/07/25 | 初版 |  |
|  |  |  |  |
|  |  |  |  |

**目次**

[1. はじめに 1](#_Toc520352126)

[1.1 入力文書 1](#_Toc520352127)

[1.2 関連文書 1](#_Toc520352128)

[1.3 制約事項 1](#_Toc520352129)

[2. Casssandra概要 2](#_Toc520352130)

[2.1 特徴 2](#_Toc520352131)

[2.2 データ構造 2](#_Toc520352132)

[3. MeghaでCassandraの役割 2](#_Toc520352133)

[3.1 役割 2](#_Toc520352134)

[3.2 Casssandra利用するイメージ図 3](#_Toc520352135)

[4. Meghaで何かを変わったか 3](#_Toc520352136)

[4.1 Capsule Teplatesの保存先 3](#_Toc520352137)

[4.2 Jobs情報の格納先は、 3](#_Toc520352138)

[4.3 XM　コンポーネント 3](#_Toc520352139)

[5. Casssandra運用ための設定 4](#_Toc520352140)

[5.1 Cassandraをインストール 4](#_Toc520352141)

[5.2 CassandraにKeySpaceとテーブルを生成 4](#_Toc520352142)

[5.3 MeghaのDB(Oracleか、Postgres)の設定 4](#_Toc520352143)

[5.4 BIO\_SERVER\_INFOにCDBを登録 4](#_Toc520352144)

[5.5 Jobを実行 5](#_Toc520352145)

[5.6 データをCasssandraに登録されるのを確認 5](#_Toc520352146)

[5.7 CassandraNode管理 5](#_Toc520352147)

[5.8 Cassandraキーポイント 5](#_Toc520352148)

Intentionally left blank

機能設計仕様書について

機能設計仕様書には、基本設計仕様書で明らかになった機能要件、非機能要件をどのように実装するかを明確にします。機能要件や非機能要件を実現するための機能を定義し、それらの外部仕様を明らかにすることで、利用者から見える範囲のソフトウェアの仕様を整理します。整理する外部仕様としては以下のようなものがあります。

- 処理内容

- 画面レイアウト

- 画面遷移

- コマンドI/F

- 設定ファイル

- ログメッセージ

また、外部に公開するインターフェースがある場合はそれらを明確にします。

1. はじめに

Section 1

本書はXM Megha1.2.0にて実装した(Necamソースの取り込む)、Casssandraの仕様について、説明するものである。

* 1. 入力文書
  2. 関連文書
  3. 制約事項

1. Casssandra概要

Section 2

ビックﾃﾞｰﾀを扱う分散型NoSql製品である。データは「Key-Value」という形で保存。

各ノードは同じ機能を持つ、マスターノードという概念がなし。詳細は以下のWikiに纏めおります：

<http://dev01a/pukiwiki/?Apache%20Cassandra>

* 1. 特徴

1. RelationDBではないですが、RelationDBと似ている。
2. 普通のデータだけではなく、Map、Listの挿入もできる。
3. クラスター(複数Node)の運用が可能
4. 高いパフォーマンス
5. 分散型で伸縮自在なスケーラビリティ
6. 単一故障点（SPOF）がないアーキテクチャ
   1. データ構造
7. Cassandraのデータはカラム中心である
8. データCenterはKeyspacesで構成
9. Keyspaces(database)は column families(table)から構成
10. column families(table)は、row（keys,columns)から構成
11. columnはデータの最小の単位、name,value,timestampから構成
12. SuperColumnは複数の集まり

1. MeghaでCassandraの役割
   1. 役割
2. Capsule Templates の格納
3. Job情報の格納
   1. Casssandra利用するイメージ図

LobStorageType=CASSANDRA

TemplateStorageType=CASSANDRA

Megha

TemplteData、

TemplateData

JobInfo

JobInfo

設定確認

Insert

Search

DB

VN

SN

EN

1. Meghaで何かを変わったか
   1. Capsule Teplatesの保存先

FILE, DB, FILE\_TO\_DB,以外に、CASSANDRAを追加される。

* 1. Jobs情報の格納先は、

FILE以外に、CASSANDRAを追加される。

* 1. XM　コンポーネント

XM　コンポーネントにCDBを追加される。

CasssandraはSC、SNのように、独立のコンポーネントの扱いである。

1. Casssandra運用ための設定
   1. Cassandraをインストール

　　　その前にJDK8、Python2.7.xを先にインストールされることを確認

* 1. CassandraにKeySpaceとテーブルを生成

CREATE KEYSPACE bio\_matcher WITH REPLICATION = { 'class' : 'SimpleStrategy', 'replication\_factor' : 1 };

use bio\_matcher;

CREATE TABLE bio\_template\_data\_info (TEMPLATE\_DATA\_ID varchar PRIMARY KEY, CREATE\_DATETIME timestamp, UPDATE\_DATETIME timestamp, TEMPLATE\_DATA blob);

CREATE TABLE bio\_lob\_data\_info (LOB\_ID varchar, LOB\_TYPE varchar, CREATE\_DATETIME timestamp, LOB\_DATA blob, PRIMARY KEY (LOB\_ID, LOB\_TYPE));

CREATE INDEX bio\_lob\_data\_info\_idx01 ON bio\_lob\_data\_info (CREATE\_DATETIME);

* 1. MeghaのDB(Oracleか、Postgres)の設定

BIO\_PARAMETERSテーブルにて、「TEMPLATE\_STORAGE\_TYPE」、「LOB\_STORAGE\_TYPE」を「CASSANDRA」に設定

* 1. BIO\_SERVER\_INFOにCDBを登録

Insert into BIO\_SERVER\_INFO

(SERVER\_ID, SERVER\_HOSTNAME, SERVER\_TYPE, SERVER\_STATE, COMPONENT\_TYPE,

SERVER\_GROUP\_ID, MAX\_JOB\_COUNT, CREATE\_DATETIME)

Values

('CDB001', '10.10.10.11', 'CASSANDRA\_DB', 'ACTIVE', 'CDB',

'CDBG01', 0, CURRENT\_TIMESTAMP);

COMMIT;

* 1. Jobを実行

普通の通り必要なJobを実行。特にかわることはなし。

* 1. データをCasssandraに登録されるのを確認

Casssandraマシンで、cqlを実行して確認、以下はcql サンプルです：

./cqlsh 192.168.22.106 9042

use bio\_matcher;

select \* from bio\_template\_data\_info;

select count(\*) from bio\_template\_data\_info;

select \* from bio\_lob\_data\_info;

select count(\*) from bio\_lob\_data\_info;

* 1. CassandraNode管理

以下のCommandで、CasssandraNodeのStatusなどを認できる：

./nodetool status

./nodetool cfstats

./nodetool tpstats

./nodetool netstats

./nodetool cfhistograms bio\_matcher bio\_lob\_data\_info;

./nodetool cfhistograms bio\_matcher bio\_lob\_data\_info;

* 1. Cassandraキーポイント

CAP

Dataの保存

データのReplication　Policy

Partitioners

consistency level

QUORUM

並列実行

データのCache

どのNodeに接続か

ConnectionPool

＝＝＝以上＝＝＝