

Um Levantamento sobre Sistemas de Gerenciamento de Bancos de Dados NoSQL Multimodelo

Amanda Christoval da Veiga de Aquino, Ronaldo dos Santos Mello

¹Universidade Federal de Santa Catarina (UFSC)
Campus Universitário Trindade – CP 476 – 88.040-900 – Florianópolis (SC), Brasil

amandacveiga01@gmail.com, r.mello@ufsc.br

Abstract. *The number of applications that deals with highly variable data is increasing. This is the case for systems that collect data from sensors or social networks. In this sense, it is necessary to develop data management technologies that support this data heterogeneity and offer a good performance. In order to achieve these objectives, database management systems (DBMS) called multi-model NoSQL have been proposed. For developers who need to use such DBMS it is not yet clear which solution best fits their needs as these technologies are considered new. In the literature we find out works comparing some solutions, however those comparing only multimodel NoSQL DBMS are few and do not cover many solutions. Thus, this paper presents a survey regarding the most popular multimodel NoSQL DBMS that are not completely covered by the related work. We also consider new comparison criteria.*

Resumo. *O número de aplicações que lida com dados bastante variados tem aumentado, como é o caso de sistemas que coletam dados de sensores ou de redes sociais. Nesse sentido, é necessário desenvolver tecnologias de gerenciamento de dados que suportem essa heterogeneidade de dados e apresentem um bom desempenho. Para atingir esses objetivos, sistemas de gerenciamento de banco de dados (SGBDs) chamados de NoSQL multimodelo vêm sendo propostos. Para desenvolvedores que pretendem utilizar tais SGBDs ainda não é claro qual solução se adapta melhor ao seu projeto, pois essas tecnologias são consideradas recentes. Na literatura existem trabalhos comparando SGBDs, entretanto os que comparam apenas SGBDs NoSQL multimodelo são poucos e não abrangem muitas soluções. Portanto, esse artigo apresenta um levantamento de SGBDs NoSQL multimodelo populares e não abordados na íntegra por trabalhos relacionados. Esse artigo apresenta também diferentes critérios de comparação.*

1. Introdução

Atualmente, uma grande quantidade de dados está sendo gerada a partir de fontes de dados diferentes como *smartphones*, sensores, câmeras, *sites* de redes sociais, transações comerciais e jogos [Kaur and Rani 2013]. Com a ajuda de ferramentas de análise é possível extrair, a partir desses dados, inúmeras informações úteis para a tomada de decisões. Contudo, essa não é uma tarefa fácil. Um dos desafios para a comunidade de pesquisa de Banco de Dados (BD) nos últimos anos tem sido o desenvolvimento de tecnologias para gerenciar essa grande quantidade de dados heterogêneos [Lomotey and Deters 2014].

A atual tecnologia de BD relacional não é adequada para lidar com dados altamente heterogêneos devido ao alto custo com controle de consistência. Isso incentivou o desenvolvimento de novas soluções, como os Sistemas de Gerência de BDs (SGBDs) NoSQL [Sadalage and Fowler 2012]. Eles lidam com armazenamento distribuído e não utilizam o modelo de dados relacional. Eles são uma família de modelos de dados, sendo os quatro principais o *chave-valor* e os *orientados a colunas*, *documentos* e *grafos*.

Mais recentemente, como uma resposta à necessidade de lidar com múltiplos formatos de dados, SGBDs que suportam vários modelos de dados simultaneamente vêm sendo disponibilizados [Lu and Holubová 2019]. Aplicações com tais requisitos realizam o que se denomina de *persistência poliglota*, uma vez que armazenam dados em diversos formatos [Sadalage and Fowler 2012]. O casamento de SGBDs NoSQL com tratamento poliglota de dados é aqui denominado de *SGBD NoSQL multimodelo*, ou seja, um único SGBD que armazena, indexa e consulta dados representados em vários modelos de dados NoSQL. Isso evita o acesso a vários SGBDs NoSQL monomodelo.

Este trabalho apresenta um levantamento de SGBDs NoSQL multimodelo. Eles são comparados segundo critérios relevantes, como suporte à administração, manipulação e modelos de dados para, desta forma, auxiliar a tomada de decisão de profissionais na escolha de um SGBD que atenda melhor as suas necessidades. Alguns trabalhos relacionados analisam soluções NoSQL [Oliveira and del Val Cura 2016, Fernandes and Bernardino 2018, Zhang et al. 2018]. Entretanto, eles consideram poucas soluções e não enfatizam apenas SGBDs NoSQL multimodelo.

Este artigo possui mais quatro seções. A seção 2 introduz SGBDs NoSQL multimodelo e as soluções escolhidas para a análise. A seção 3 aborda trabalhos relacionados e a seção 4 detalha o levantamento. Por fim, a seção 5 é dedicada à conclusão.

2. SGBDs NoSQL Multimodelo

SGBDs NoSQL multimodelos são SGBDs concebidos com o intuito de gerenciar diferentes modelos de dados NoSQL em um único SGBD, além das vantagens inerentes aos SGBDs NoSQL monomodelo, como escalabilidade, disponibilidade e um modelo de dados não-relacional menos rígido [Sadalage and Fowler 2012]. Este trabalho considera quatro SGBDs NoSQL multimodelo comercialmente disponíveis e populares: *ArangoDB*¹, *OrientDB*², *CrateDB*³ e *Marklogic*⁴. Eles foram escolhidos por que são citados no *ranking* do site *DB Engines*⁵ e possuem documentação para os critérios de comparação considerados nos seus *Web sites*.

O ArangoDB é um SGBD de código aberto concebido para armazenar dados na forma de pares chave/valor, grafo ou documento, acessando dados em qualquer um destes modelos através de uma única linguagem de consulta declarativa denominada *AQL* (*Arango Query Language*). Além disso, ele possui suporte a transações ACID.

O OrientDB é um SGBD que também respeita as propriedades ACID e suporta os modelos chave-valor e orientados a grafo e documento. No caso do modelo orientado

¹<https://www.arangodb.com/>

²<https://orientdb.com/>

³<https://crate.io/>

⁴<https://www.marklogic.com/>

⁵<https://db-engines.com/en/ranking>

a documentos, dados são mantidos em JSON e relações entre documentos são definidas através de *links*. Esses links são utilizados para realizar junções. No caso do modelo orientado a grafos, o OrientDB apresenta uma estrutura composta por vértices e arestas. Já o modelo chave/valor do OrientDB é mais robusto que o modelo NoSQL chave/valor clássico. Ele suporta documentos e grafos como valores.

O Marklogic é um SGBD com armazenamento nativo para binários JSON, XML, RDF, dados geoespaciais e dados grandes (por exemplo, PDFs, imagens e vídeos). Já em nível lógico, operações de manipulação estão centradas em estruturas RDF e JSON. A primeira serve como abstração para dados orientados a grafos e a segunda para dados orientados a documentos. O Marklogic também é compatível com as propriedades ACID.

O CrateDB é um SGBD distribuído de código aberto que integra o modelo chave-valor com o modelo orientado a documentos. Basicamente, estruturas chave-valor são convertidas em documentos simples compostos de uma identificação e um conteúdo. Diferente dos SGBDs anteriores, o CrateDB não garante todas as propriedades ACID, pois não suporta consistência forte. Ele armazena dados no formato JSON em sistemas de arquivos distribuídos em nós de um *cluster*.

Mais detalhes sobre cada uma destas soluções são apresentadas na seção 4.

3. Trabalhos Relacionados

Algumas iniciativas na literatura possuem similaridade com a pesquisa apresentada neste trabalho. Uma delas avalia o ArangoDB e o OrientDB em um sistema com persistência poliglota [Oliveira and del Val Cura 2016]. Os autores comparam o desempenho destes SGBDs com os SGBDs NoSQL monomodelo Neo4J e MongoDB, recomendando o ArangoDB devido ao seu suporte simultâneo a grafos e documentos. Outra iniciativa compara o OrientDB com SGBDs relacionais e NoSQL com relação a facilidades de consulta, ressaltando os benefícios do OrientDB [Kosmerl et al. 2020].

Um outro trabalho avalia características que impactam no cotidiano de um administrador de BD (DBA) para a escolha de SGBDs NoSQL orientados a grafos, sendo alguns deles multimodelo [Fernandes and Bernardino 2018]. Eles concluem que Neo4J e ArangoDB se sobressaem em termos de funcionalidades, sendo as melhores opções. Uma comparação do ArangoDB e OrientDB utilizando um *benchmark* é o foco de outra pesquisa [Zhang et al. 2018]. A partir de dados sintéticos com formatos diversos, avaliou-se tempo de importação e desempenho de transações utilizando diferentes interfaces de acesso. A conclusão é que ambos os SGBDs são capazes de ingerir uma variedade de dados sem muito esforço adicional e suportar junções multimodelo, como grafo-JSON.

Mais recentemente, um *survey* comparando soluções mono e multimodelo foi produzido [Lu and Holubová 2019]. Esta pesquisa se diferencia de todos esses trabalhos ao analisar apenas SGBD NoSQL multimodelo e não enfatizar um modelo de dados específico suportado por eles. Além disso, os SGBDs NoSQL multimodelo aqui levantados não foram comparados por nenhum dos trabalhos e o conjunto de critérios de comparação proposto também é inovador.

4. Levantamento

Esta seção apresenta o levantamento dos SGBDs NoSQL multimodelo escolhidos. Os critérios de comparação foram definidos com base na documentação dos SGBDs. Eles

são os seguintes: (i) *modelos de dados*; (ii) *licenças de uso*; (iii) *sistemas operacionais*; (iv) *gerenciamento de esquemas*; (v) *formas de acesso*; (vi) *operações*; (vii) *índices*; (viii) *gerenciamento de backups*; (ix) *controle de concorrência*; e (x) *ferramentas de administração de dados*.

Cabe ressaltar que não é considerado neste trabalho uma avaliação de desempenho dos SGBDs utilizando, por exemplo, um *benchmark*. Este tipo de análise já foi considerado na literatura, sendo o foco aqui comparar os suportes de cada produto com base nas suas documentações. Uma análise de cada critério de comparação é detalhada a seguir.

4.1. Modelos de Dados

O ArangoDB suporta três modelos de dados: chave-valor, orientado a documento e orientado a grafo. Para tanto, ele utiliza a flexibilidade do JSON (comum em BDs orientados a documento) como formato de armazenamento, mapeando dados nos demais modelos para ele. O formato JSON permite armazenar dados complexos, por isso esta escolha.

O OrientDB suporta os mesmos modelos de dados do ArangoDB. Para o modelo orientado a documento, os dados são guardados em documentos que podem ser mantidos em coleções, utilizando os conceitos de classe e *cluster*. O modelo orientado a grafo é representado através de classes para vértices e arestas. Para o modelo chave-valor, assume-se que valores simples ou complexos são acessados a partir de uma chave.

O CrateDB suporta os modelos chave-valor e orientado a documento. Sua documentação informa apenas que sua unidade de armazenamento é o documento. Assim sendo, dados do tipo chave-valor são provavelmente mapeados para documentos. Já o Marklogic suporta os modelos orientados a grafos e documentos, sendo possível utilizar os formatos JSON para manter documentos que representam entidades, XML para dados textuais, bem como RDF para dados de grafos.

Tabela 1. Critério de Comparação: Modelo de dados

Modelo de Dados NoSQL	ArangoDB	OrientDB	CrateDB	Marklogic
orientado a colunas				
chave-valor	X	X	X	
orientado a documentos	X	X	X	X
orientado a grafos	X	X		X

A Tabela 1 compara os modelos de dados NoSQL adotados. O modelo de dados suportado por todos é o orientado a documentos e o modelo orientado a colunas não é suportado por nenhum. Percebe-se que o ArangoDB e OrientDB são os produtos que consideram mais modelos de dados NoSQL, levando vantagem sobre os demais.

4.2. Licenças de Uso

O ArangoDB possui duas edições: *community* e *enterprise*. A primeira é uma versão de código aberto. Já a segunda é uma versão paga voltada para projetos que requerem armazenamento altamente disponível, escalabilidade exclusiva e suporte incluído. O OrientDB é similar ao ArangoDB, sendo que a sua versão paga possui um período de teste gratuito de 30 dias. O CrateDB também disponibiliza duas versões. A versão paga inclui alguns benefícios, como acesso autenticado, monitoramento de *cluster* e gerenciamento

de usuários. Pode-se solicitar uma licença da versão paga sem custos caso o usuário for uma instituição sem fins lucrativos ou educacional.

Por fim, o Marklogic também oferece duas versões de uso. Deste modo, é possível constatar que todos os SGBDs possuem versões gratuita e paga. A versão gratuita possui menos benefícios, porém, é ideal para que DBAs e desenvolvedores se familiarizem com a solução e avaliem se ela se adapta às necessidades da aplicação. Vale ressaltar que o OrientDB oferece a oportunidade de um período de teste na edição paga e o CrateDB disponibiliza uma versão paga para instituições sem fins lucrativos ou educacionais.

4.3. Sistemas Operacionais

O ArangoDB e o CrateDB suportam os três Sistemas Operacionais (SOs) mais comuns: *Windows*, *Linux* e *MacOS*. O OrientDB executa em SOs que implementam o *Java Virtual Machine (JVM)*, como Linux, Mac OS, Windows (após a versão 95), Solaris, HP-UX e IBM AIS. Já o Marklogic suporta o Windows (versão 7 ou posterior), Red Hat Enterprise Linux 7, CentOS 7, MacOS (versão 10.14 ou posterior) e Linux.

Em suma, todos os SGBDs suportam os três SOs mais comuns. OrientDB é o SGBD que executa em um número maior de SOs, o que o torna mais robusto neste critério. Contudo, o Marklogic se diferencia pelo nível de detalhamento de sua documentação em relação aos SOs suportados e boas práticas.

4.4. Gerenciamento de Esquemas

O ArangoDB é classificado como *schemaless*, ou seja, não é necessário especificar nenhuma estrutura pré-definida para os dados. Todos os documentos podem apresentar uma estrutura integralmente diferente do outro em uma mesma coleção de documentos. O OrientDB também pode atuar como *schemaless*, porém, ele suporta também abordagens *schemafull* e híbrida. O Marklogic também oferece os modos *schemaless* e *schemafull*, enquanto que o CrateDB é *schemaless*, mas possui um esquema especial, chamado de *information schema*, no qual é possível conhecer a organização interna dos documentos.

Tabela 2. Critério de Comparação: Gerenciamento de Esquemas

Tipo de Gerenciamento	ArangoDB	OrientDB	CrateDB	Marklogic
<i>schemaless</i>	X	X	X	X
<i>schemafull</i>		X		X
híbrido		X		

Nenhum SGBDs impõe uma estrutura *a priori*, conforme mostra a Tabela 2. Neste critério, o ArangoDB e o CrateDB são mais limitados, uma vez que oferecem apenas o modo *schemaless*. Já o OrientDB suporta os três tipos de gerenciamento de esquema.

4.5. Formas de Acesso

O método de acesso mais conhecido do ArangoDB é a AQL. Além dela, existem outras formas de acesso como por *shell* e *drivers* oficiais para diversas linguagens de programação (Java, Java assíncrono, JavaScript, PHP e Go). Outra forma de acesso é a API HTTP, que possui uma vasta documentação. O OrientDB tem o Java como forma

de acesso nativo aos dados. Neste caso é necessário utilizar uma API *socket TCP/IP* apropriada à linguagem de programação utilizada pela aplicação. Acesso via SQL também é permitido e o OrientDB disponibiliza uma extensa lista de *drivers* de acesso: Java, Node, PHP, Net, Go, Gremlin, C, JavaScript, Ruby, Scala, R, Elixir e Perl. O CrateDB utiliza um dialeto SQL para consultar dados e suporta as APIs de acesso JDBC, ODBC e REST. Por fim, as formas de acesso do Marklogic são via XQuery, SPARQL, API REST e *drivers* para Java, Node e Javascript. Ele também permite a manipulação dos dados via SQL.

Ao analisar esse critério, fica evidente os diversos métodos de acesso que o OrientDB possui, possuindo conectividade com mais linguagens de programação que os demais SGBDs. Todavia, o ArangoDB e Marklogic também possuem um número considerável de opções. O menos robusto neste critério é o CrateDB.

4.6. Operações de Acesso

O ArangoDB suporta as operações básicas *CRUD* (*Create, Read, Update e Delete*). Em particular, ele disponibiliza um comando REPLACE, que substitui um documento por completo, e um comando UPSERT, que insere ou atualiza condicionalmente documentos em uma coleção. A AQL não permite junções, mas permite agrupamentos similares à cláusula GROUP BY da SQL através dos comandos COLLECT e AGGREGATE.

Por utilizar SQL como linguagem de consulta, o OrientDB suporta as operações básicas CRUD, além da operação MATCH e agregação/agrupamento. A operação MATCH é semelhante ao SELECT, porém, é utilizada para buscar dados por correspondência de padrões. O OrientDB permite ainda junções entre dados. O CrateDB provê as operações básicas CRUD. Além disso, também oferece uma operação similar à UPSERT através da cláusula ON CONFLIT DO UPDATE SET. Operações de junção e agrupamento, bem como subconsultas também são oferecidas. Ainda, o Marklogic suporta operações CRUD e agrupamento. Entretanto, não suporta junções e operações com subconsultas que envolvem agrupamento.

Tabela 3. Critério de Comparação: Operações de Acesso

Operação	ArangoDB	OrientDB	CrateDB	Marklogic
CRUD	X	X	X	X
Junção	X	X	X	
Agrupamento	X	X	X	X
UPSERT	X		X	

A Tabela 3 compara esse critério considerando as operações mais comuns de manipulação de dados. ArangoDB, OrientDB e CrateDB oferecem um amplo conjunto de operações. O Marklogic fica em desvantagem por não suportar junções e UPSERT.

4.7. Índices

O ArangoDB oferece diversos tipos de índices. O *Primary index* é um índice para coleções, mantendo as chaves dos seus documentos. Já o *Edge index* é utilizado nas coleções de arestas, indexando campos que definem relacionamentos entre dados. O *Hash index* é utilizado para buscas rápidas de documentos com base em valores de atributos específicos. O *TTL (Time-To-Live) index* é utilizado internamente para expirar documentos

de uma coleção com base em uma configuração de tempo de validade. Além disso, o ArangoDB oferece os índices *Skiplist*, *Fulltext* e *Geo*. O primeiro mantém uma estrutura de dados ordenada que garante boa eficiência em pesquisas por intervalo e para conjuntos de dados que sofrem muita atualização, uma vez que não requer balanceamento de estruturas de árvore, como exigido pelos índices B-tree ou AVL. O segundo é utilizado para encontrar palavras ou prefixos em documentos, e o terceiro é utilizado para encontrar locais específicos na superfície da Terra.

O OrientDB suporta alguns índices similares aos do ArangoDB: *Fulltext*, *Spatial* e *Hash*. Além disso, ele oferece o *SB-Tree index*, que é baseado em índices B-Tree, porém adaptado com otimizações relacionadas à inserção de dados. O CrateDB suporta índices *Primary*, *Geo* e *Fulltext*. Por fim, o Marklogic oferece o *Universal index*, utilizado para indexar palavras, frases, relações e valores. Além desses, disponibiliza outros tipos de índice: *Range index*, *Triple index* e *Reverse index*. O primeiro é indicado para buscas por desigualdade. O segundo é um índice de três níveis para buscas de termos que aparecem em sequência. O terceiro reúne o conjunto distinto de termos de consulta do nó folha de uma árvore de processamento de consulta, ou seja, os termos de consulta não compostos. Para cada nó folha é mantido um conjunto de IDs de documento para nomear como uma possível correspondência inversa de consulta quando esse termo está presente em um documento, e outro conjunto de IDs para nomear quando o termo não está presente.

Com relação a esse critério, a maioria dos SGBDs têm uma abrangência relativamente boa e algumas peculiaridades. O ArangoDB e o Marklogic se destacam por oferecer um maior número de tipos de índices. O CrateDB é o que menos se sobressai, inclusive por não suportar índices baseados em estruturas B-tree.

4.8. Gerenciamento de *Backups*

O ArangoDB oferece três métodos de *backup*: físico, lógico e quente. O primeiro pode ser realizado quando o servidor não estiver rodando (*backup offline*). O terceiro, por sua vez, pode ser invocado com o servidor rodando, mas só está disponível na versão paga (*backup online*). Já o *backup* lógico é utilizado para mover ou arquivar um BD (registros e esquemas). Ele copia os dados, porém não os arquivos físicos.

No caso do OrientDB, *backups* geram uma cópia comprimida de todo o BD. Os processos de *backup* e restauração (*restore*) podem ser realizados manualmente pelo DBA ou serem pré-programados através de uma API. O OrientDB executa também esses processos de forma incremental, porém essa funcionalidade está disponível apenas na versão paga. O CrateDB também permite a execução de processos de *backup* e restauração de dados manuais ou incrementais. Já o Marklogic disponibiliza *backup* imediato ou programado. O usuário pode escolher a melhor hora para realizar o *backup* e quais são os documentos e esquemas a serem copiados. Ambos os tipos podem ser executados de forma *online*. Além disso, ele também oferece *backup* incremental.

A Tabela 4 compara esse critério de análise. Todos os SGBDs apresentados possuem suporte à *backup* e restauração de dados. Em relação a documentação sobre recuperação de falhas, o CrateDB apresenta uma documentação escassa e não muito clara. O ArangoDB deixa a desejar por ser o único a não suportar *backup* incremental. Já a vantagem do Marklogic é que ele suporta diversos tipos de *backup*.

Tabela 4. Critério de Comparação: Recuperação de Falhas

Funcionalidade	ArangoDB	OrientDB	CrateDB	Marklogic
<i>Backup online</i>	X			X
<i>Backup incremental</i>		X	X	X
<i>Backup programado</i>		X		X
<i>Restore</i>	X	X	X	X

4.9. Controle de Concorrência

O ArangoDB delega o controle de concorrência para dois produtos: *MMFiles* e o *RocksDB*. No primeiro caso, uma coleção é bloqueada, no início da transação, no modo solicitado (leitura ou escrita). Caso várias coleções forem manipuladas, é realizado o bloqueio em ordem alfabética. Assim, quando a transação termina, os bloqueios são liberados na ordem inversa. Durante um bloqueio de escrita, outras transações que desejam modificar as mesmas coleções são impedidas, ou seja, cada transação executa de forma isolada para fins de escrita. O *RocksDB*, por sua vez, não realiza o bloqueio de nenhuma coleção que participa de uma operação de leitura. Leituras podem ser executadas paralelamente a outras operações. Contudo, no caso de duas transações tentarem modificar o mesmo documento, ambas são abortadas. O isolamento, no caso do *RocksDB*, é do tipo *snapshot isolation*, ou seja, uma transação enxerga a última versão consistente do BD.

O OrientDB pressupõe que várias transações possam executar concorrentemente sem interferir umas nas outras. Se ocorrer algum conflito de acesso, então algumas transações são abortadas com base nos seus *timestamps*. Para operações atômicas é utilizada a abordagem *MVCC (MultiVersion Concurrency Control)*, que evita bloqueios. Nesta abordagem, transações que atualizam um dado geram uma nova versão deste dado que fica disponível assim que a transação terminar com sucesso. Porém, enquanto a transação não finaliza, a última versão do dado é disponível apenas para leitura [Neumann et al. 2015]. O CrateDB também utiliza a abordagem *timestamp* e o Marklogic também utiliza a abordagem *MVCC*.

Tabela 5. Critério de Comparação: Controle de Concorrência

Abordagem	ArangoDB	OrientDB	CrateDB	Marklogic
Bloqueio	X			
<i>Snapshot Isolation</i>	X			
MVCC		X		X
<i>Timestamp</i>		X	X	

Conforme mostra a Tabela 5, o Marklogic e OrientDB utilizam a abordagem *MVCC*, que é barata e garante as propriedades *ACID*. O CrateDB possui a abordagem mais simples de controle de concorrência (*timestamp*) que, diferente das demais, é recomendado para ambiente de baixa contenção de dados, onde conflitos são raros. O ArangoDB se destaca por oferecer duas abordagens de controle de concorrência, deixando a cargo do usuário escolher qual mecanismo melhor se adequa à sua aplicação. OrientDB também lida com dois protocolos, conforme o tipo de operação.

4.10. Ferramentas de Administração

O ArangoDB possui uma ferramenta denominada *Arangod*. Seu painel de controle apresenta janelas para consultar estatísticas do SGBD e dos seus *clusters* (transações por segundo, número de conexões do cliente, etc), recursos do sistema (número de processos, uso da memória e da CPU, etc) e replicações de dados. A *Arangod* disponibiliza ainda um editor de AQL e um painel de grafos. O painel permite uma fácil interação com dados nesse formato. Por fim, a *Arangod* possui painéis para controle de usuários e de *logs*.

A ferramenta do OrientDB se chama *Orient Studio*. Ela oferece diversas opções de interação para o DBA: *Browser* (consultas a dados); *Schema* (gerência de metadados); *Security* (gerência de usuários); *Graph* (manipulação de dados na forma de grafos); *Functions* (gerência de funções do SGBD); *Backup*; *Database Management* (gerência de clusters e configurações); e *Dashboard* (estatísticas sobre os dados).

O CrateDB possui uma ferramenta chamada *Crate admin UI*. Ela apresenta as seguintes abas com funcionalidades: *Console* (gerência de consultas); *Tables* (visualização de dados e metadados de documentos); *Clusters* (gerência de *clusters*); *Privileges* (gestão de usuários); e *Monitoring* (estatísticas sobre os dados). Por fim, as funcionalidades da ferramenta do Marklogic se resumem à gerência de configuração do SGBD, grupos de usuários, BDs e coleções, *backup* e *restore*, caminhos de acesso ao servidor Web e ao Java, bem como ajustes de desempenho e consulta aos status dos seus recursos.

A partir dessas informações, percebe-se que o ArangoDB possui uma ferramenta mais robusta para DBAs. OrientDB e CrateDB também disponibilizam diversos dados estatísticos sobre seus BDs, entretanto, em menor quantidade se comparado ao ArangoDB. O Marklogic fica em desvantagem por não disponibilizar estatísticas. ArangoDB e o OrientDB se destacam ainda pela visualização na forma de grafos.

4.11. Comparativo Geral

O ArangoDB se destaca nos critérios modelos de dados, formas de acesso, índices e ferramentas de administração. Ele considera o maior número de modelos de dados, juntamente com o OrientDB, além de suportar diferentes formas de acesso, as principais operações de acesso e muitos tipos de índices. Entretanto, seus pontos fracos são o gerenciamento de esquemas, pois ele suporta apenas o modo *schemaless*, e controle de concorrência, uma vez que utiliza principalmente técnicas de bloqueio, que são custosas.

O OrientDB se destaca nos critérios modelos de dados, licenças de uso, SOs, gerenciamento de esquemas, controle de concorrência e ferramenta de administração. Além de suportar diversos modelos de dados, ele também oferece um período de teste para sua versão paga, executa no maior número de SOs, gerencia o maior número de tipos de esquemas, lida com mais de uma técnica de controle de concorrência, sendo que ambas são técnicas com bom desempenho, e sua ferramenta de administração é uma das que oferece maior número de recursos.

O CrateDB se destaca nos critérios licenças de uso, operações de acesso e ferramentas de administração. No primeiro caso, ele disponibiliza versões pagas (mais completas) para instituições sem fins lucrativos. No terceiro caso, ele oferece diversos tipos de interações e informações úteis aos usuários. Entretanto, ele apresenta diversos pontos fracos, como o fato de suportar poucos modelos de dados, não gerenciar esquemas para os dados e disponibilizar poucas formas de acesso.

O Marklogic se destaca nos critérios formas de acesso, índices e recuperação de falhas. Suas formas de acesso são bastante variadas, incluindo linguagens, APIs e *drivers*. Além disso, ele oferece a maior variedade de operações de *backup* de dados. Entretanto, o Marklogic suporta poucos modelos de dados, é o mais limitado em termos de operações de acesso e sua ferramenta de administração oferece poucos recursos.

5. Conclusão

Este trabalho apresenta um levantamento de SGBDs NoSQL multimodelo populares. Ele inova por analisar apenas SGBDs multimodelo e empregar um conjunto de critérios de comparação diferente. Os critérios inéditos considerados neste trabalho são: (i) *modelos de dados*; (ii) *licenças de uso*; (iii) *SOs*; (iv) *formas de acesso*; (v) *índices*; (vi) *controle de concorrência*; e (vii) *ferramentas de administração de dados*. A motivação é favorecer o trabalho de DBAs que desejam utilizar essas novas tecnologias em seus projetos.

O SGBD NoSQL multimodelo de maior destaque é o OrientDB. Também considera-se o ArangoDB um bom produto, principalmente em termos de modelos de dados suportados, formas de acesso e indexação. Já o SGBD de pior destaque é o CrateDB. Mesmo assim, vale lembrar que não existe uma solução perfeita, e sim a que melhor se encaixa nas necessidades de cada aplicação. Como trabalhos futuros, sugere-se outros tipos de análise: processo de instalação, desempenho e SGBDs com os mesmos modelos de dados. Por fim, sugere-se uma comparação entre SGBDs NoSQL mono e multimodelo.

Referências

- Fernandes, D. and Bernardino, J. (2018). Graph databases comparison: Allegrograph, arangodb, infinitegraph, neo4j, and orientdb. In *7th International Conference on Data Science, Technology and Applications (DATA)*, pages 373–380.
- Kaur, K. and Rani, R. (2013). Modeling and querying data in NoSQL databases. In *International Conference on Big Data*, pages 1–7. IEEE.
- Kosmerl, I., Rabuzin, K., and Sestak, M. (2020). Multi-model databases - introducing polyglot persistence in the big data world. In *43rd International Convention on Information, Communication and Electronic Technology, MIPRO*, pages 1724–1729. IEEE.
- Lomotey, R. K. and Deters, R. (2014). Towards knowledge discovery in big data. In *IEEE 8th International Symposium on Service Oriented System Engineering (SOSE)*.
- Lu, J. and Holubová, I. (2019). Multi-model databases: A new journey to handle the variety of data. *ACM Comput. Surv.*, 52(3):55:1–55:38.
- Neumann, T., Mühlbauer, T., and Kemper, A. (2015). Fast serializable multi-version concurrency control for main-memory database systems. In *ACM SIGMOD International Conference on Management of Dats*, pages 677–689.
- Oliveira, F. R. and del Val Cura, L. M. (2016). Performance evaluation of nosql multi-model data stores in polyglot persistence applications. In *20th International Database Engineering & Applications Symposium (IDEAS)*, pages 230–235.
- Sadalage, P. J. and Fowler, M. (2012). *NoSQL Distilled*. Pearson.
- Zhang, C., Lu, J., Xu, P., and Chen, Y. (2018). Unibench: A benchmark for multi-model database management systems. In *Performance Evaluation and Benchmarking for the Era of Artificial Intelligence - 10th TPC Technology Conference (TPCTC)*, pages 7–23.