

15 - Performance and Troubleshooting

SEARCH AND SPEED PROBLEMS

- Pode ter múltiplas causas
 - Hardware:
 - Alocar SSD e Memória abundantes ajudam a evitar gargalos no Lucene subjacente
 - Memória Heap armazena objetos novos
 - Mais Heap evita que se encha e requer menos garbage collection, dá tempo para dados serem movidos à outra memória.
 - Recomendado é metade da memória do node vai à aplicação como heap.
 - Mais shards pequenos aumenta necessidade de comunicação entre os nodes
 - Pode gerar gargalo de rede.
 - Mais réplicas melhora desempenho de read, mas consome memória e pode gargalar e ser difícil de gerir.
 - Tamanho de shard deve ser equilibrado
 - Document Modeling:
 - ES é NoSQL
 - Dados devem ser denormalizados (i.e. sem joins)
 - Melhor não haver hierarquia entre dados, cada documento é autossuficiente.
 - Também é bom evitar buscar em muitos campos ao mesmo tempo, operação cara.
 - Melhor combinar vários campos em um só e buscar nesse, se for o caso.
 - Podemos usar o copy_to no mapping do index para isso.
 - Usar Keyword type no lugar ao junto de text type para busca de termos exatos pode melhorar desempenho
 - Poupa etapa de análise do texto

INDEX SPEED PROBLEMS

- Ids aleatórios gerados pelo elastic poupam trabalho de checar unicidade do id.
 - Mas sujeito a duplicação
- Request bulk é bem melhor que um documento por vez
 - Mas tem um limite de tamanho até ser prejudicial.
 - Aumentar refresh rate pode melhorar isso, usa melhor o disco.
- Ajuste da refresh rate
 - Taxa com que elastic verifica se existe algum documento novo e o indexa.
 - Refresh pode ser desabilitado por completo para inserir vários documentos de uma vez só.
 - _refresh pela API para refresh manual.
 - Ajuste do thread_pool para o write ou para o search também pode influir.

UNSTABLE CLUSTER

- Estado RED é uma emergência.
 - Múltiplas causas (Rede, hardware, arquivos corrompidos etc)
- Se o ping do master não for bem sucedido, nova eleição do master é feita imediatamente.
- Pode haver shards não alocados de um index existente.
 - Geralmente durante a etapa de rebalanceamento e/ou após falha de algum node
 - Read e Write são bloqueados até que o shard seja alocado corretamente.
 - API de _explain explica por que um shard não pode ser alocado
- Limites de uso de disco
 - Por default, num node o low-disk watermark acontece com 85% do disco utilizado.

- ES não aloca novos shards a este node, neste caso
- Volta a poder alocar quando disco volta para abaixo deste nível
- High-disk watermark
 - Idem para 90% do disco utilizado.
 - Shards neste node são realocados para outros nodes
- Flood-stage-disk watermark
 - 95% do disco utilizado.
 - Todos os index nesse node se tornam não-writable, viram read-only
 - Volta ao normal quando disco é liberado

CIRCUIT BREAKERS

- Alguma função que é ativada quando ocorre algum problema na cadeia de chamadas.
 - e.g. disparos de funções que evitam falha de node quando há um aumento no uso de memória.
- Erro de Out of Memory antes que a operação seja realizada.

Table 15.1 Types of circuit breakers and their memory settings

Circuit breaker	Description and minimum memory limit	Property
Parent	The total memory that can be used across all the other circuit breakers. Defaults to 70% of JVM heap memory if real-time memory is taken into account (<code>indices.breaker.total.use_real_memory=true</code>); otherwise, it is 95% of JVM heap memory.	<code>indices.breaker.total.limit</code>
Inflight requests	The sum total of memory of all inflight requests, which is not to be exceeded by a threshold. The default is 100% of JVM heap memory, although it derives the actual percentage from the parent circuit breaker.	<code>network.breaker.inflight_requests.limit</code>
Request	Helps prevent exceeding the heap memory to serve a single request. The default is 60% of the JVM heap.	<code>indices.breaker.request.limit</code>
Field data	Helps prevent exceeding memory when loading fields into the <code>fields</code> cache. The default limit is 40% of heap memory.	<code>indices.breaker fielddata.limit</code>
Accounting requests	Avoids accumulating memory after the request is served. The default is 100%, which means it inherits the threshold from the parent.	<code>indices.breaker.accounting.limit</code>
Script compilation	While all other circuit breakers look after memory, this circuit breaker limits the number of inline compilations in a fixed time period. The default is 150/5 min (150 script compilations in a period of 5 minutes).	<code>script.max_compilations_rate</code>

- Assunto de troubleshooting é muito extenso
 - Este capítulo é apenas um panorama raso.
- Importante ter prática e ler documentação

Summary

- Elasticsearch is a complex search engine, and maintaining and managing healthy clusters requires expertise.
- Slower search speeds are a common problem that clients complain about when searching for data with Elasticsearch. Provisioning modern hardware with appropriate allocation of memory and computing resources helps alleviate speed problems. Options like choosing keyword data types and tweaking the document data model also help increase search speed.
- Indexing speeds are a concern as well, especially when the system is ingestion heavy. If allowed, measures like using system-generated IDs for documents can speed up the indexing process. Loading data using bulk APIs (rather than single-document APIs) is a surefire way to improve ingestion performance.
- We can turn off or increase the refresh rate during indexing and then switch it on when indexing completes. Doing so makes the documents unavailable instantly for searches but improves indexing performance by reducing I/O hits.
- Due to the distributed nature of Elasticsearch, things can go wrong in many areas, from the cluster to the filesystem, node communication, memory, and so on. Keeping a healthy cluster is paramount, and administrators must strive to maintain a GREEN cluster state. Religiously following the traffic light management-based cluster health system helps with smooth running of the cluster.
- Occasionally, shards are left unassigned, and troubleshooting to find the exact cause helps us reattach nodes if necessary or take the appropriate action.
- Elasticsearch provides thresholds on the available disk memory, including low-, high-, and flood-stage-disk watermarks. The low- and high-disk watermarks allow Elasticsearch to manage shard reallocation and alert the administrator about upcoming cluster problems. The flood-stage watermark is a dire warning about an out-of-disk memory problem on the node. To keep the cluster alive when this happens, Elasticsearch makes all shards read-only and does not allow any indexing operations on the shards of that node.
- Elasticsearch is a memory-hungry application, and appropriate controls should be implemented so that errors are instantly propagated to the client. It uses circuit-breaker patterns to create circuit breakers so the client doesn't need to wait longer than necessary. Each circuit breaker triggers Elasticsearch if the memory for a certain action grabs more memory than expected. A child circuit breaker inherits the memory thresholds from a parent circuit breaker.