

تقسيم البيانات في قاعدة البيانات الموزعة MongoDB

الجزء النظري:

1. ما هي استراتيجيات تقسيم البيانات لتحقيق قاعدة بيانات موزعة؟ وضح حالات الاستخدام التي تناسب كل استراتيجية؟

• التقسيم الأفقي (Horizontal partitioning (sharding):

في هذه الإستراتيجية، يتم تقسيم البيانات إلى أقسام منفصلة تسمى shards، حيث يحتوي كل جزء على مجموعة فرعية محددة من البيانات، بحيث يتم التقسيم بناءً على الأسطر أي كل مجموعة أسطر تكون ضمن جزء (shard) معين بحيث يكون لجميع الأجزاء نفس البنية shema. ويتم التقسيم وفق منهجيتين وكلا المنهجيتين تعتمد على shard key:

- التقسيم وفق مجالات range-based sharding: بحيث يكون كل جزء shard مسؤول عن مجال من القيم، بحيث من أجل كل سطر يتم النظر إلى مفتاح هذا السطر وتحديد إلى أي مجال ينتمي وبناءً عليه يتم تحديد ال shard التي سيتم وضع هذا السطر ضمنها. تعاني هذه الطريقة من مشكلة أنه قد لا يتم توزيع البيانات على ال shards بالتساوي فقد تنتمي مثلاً جميع القيم إلى نفس المجال وبالتالي تكون جميعها ضمن نفس ال shard.
- التقسيم وفق دالة تجزئة (تهشير) hashed sharding: وضمن هذه الطريقة يتم تهشير (تجزئة) مفتاح كل سطر وفق دالة تجزئة فنحصل على قيمة مرتبطة بأحد ال shards فيتم تخزين هذا السطر على هذه ال shard الموافقة. تعد هذه الطريقة مناسبة لتقسيم المعطيات بشكل متساوي، مما يساهم في تحسين الأداء.

حالات الاستخدام:

- أنظمة المعاملات كبيرة الحجم: يتيح التقسيم الأفقي توزيع عبء العمل بالتساوي عبر الأجزاء shards، مما يتيح المعالجة الفعالة لعدد كبير من المعاملات المتزامنة.
- الأنظمة الموزعة جغرافياً: يمكن استخدام التقسيم الأفقي لتوزيع البيانات عبر مناطق مختلفة، مما يتيح الوصول المحلي (مثلاً يكون shard key هو عنوان البلد أو المدينة) ويقلل زمن انتقال الشبكة.

• التقسيم العمودي vertical partitioning:

في هذه الإستراتيجية، يتم تقسيم البيانات عمودياً أي بناءً على نمط استخدام الحقول أو الخصائص المختلفة columns ضمن مجموعة البيانات. يحتوي كل قسم على مجموعة فرعية من الحقول fields للعناصر الموجودة في مخزن البيانات. الهدف هو تحسين الإدخال/الإخراج والأداء من خلال فصل الحقول التي يتم الوصول إليها بشكل متكرر عن الحقول التي لا يتم الوصول إليها بشكل متكرر.

حالات الاستخدام:

- تحسين الأداء: من خلال وضع الحقول التي يتم الوصول إليها بشكل متكرر في قسم منفصل، يمكن للنظام استرداد البيانات الأساسية بشكل أكثر كفاءة، مما يقلل من عمليات الإدخال والإخراج وتحسين الأداء.

- خصوصية البيانات وأمنها: يمكن تخزين البيانات الحساسة في قسم منفصل مع ضوابط أمنية إضافية، مما يوفر طبقة إضافية من الحماية.

- فصل البيانات الديناميكية dynamic data (تتغير دوماً) عن البيانات بطيئة الحركة (لا تتغير بشكل مستمر وقد لا تتغير أبداً) slow-moving data: مثلاً في نظام بيع منتجات يمكن فصل البيانات بطيئة الحركة، مثل مواصفات المنتج واسم المنتج، عن البيانات الأكثر ديناميكية، مثل مستويات المخزون، لتحسين التخزين المؤقت والأداء العام للنظام.

• التقسيم الوظيفي functional partitioning:

يتضمن التقسيم الوظيفي تجميع البيانات بناءً على كيفية استخدامها من خلال سياقات محددة مختلفة في التطبيق. ويهدف إلى تحسين عزل البيانات وأداء الوصول إلى البيانات وفصل الاهتمامات.

حالات الاستخدام:

- بنية الخدمات المصغرة: يمكن أن يكون لكل خدمة صغيرة قسم أو أقسام مخصصة لتخزين البيانات الخاصة بمجالها، مما يعزز قابلية التوسع.

- فصل القراءة والكتابة عن القراءة فقط: يمكن استخدام التقسيم الوظيفي لفصل البيانات التي يتم تحديثها بشكل متكرر (للقراءة والكتابة) عن البيانات التي تكون في الغالب للقراءة فقط. يمكن أن يؤدي ذلك إلى تحسين الأداء من خلال تمكين استراتيجيات التخزين والتخزين المؤقت المختلفة لكل نوع من أنواع البيانات.

- عزل منطقة العمل: من خلال تقسيم البيانات بناءً على اهتمامات مختلفة (مجال عمل مميزة)، يمكن للتقسيم الوظيفي تحسين إدارة البيانات والاستقلالية وقابلية الصيانة داخل كل مجال.

• يمكن دمج هذه الاستراتيجيات معاً لإنشاء نظام تقسيم أكثر شمولاً يناسب متطلبات محددة وعبء عمل البيانات.

2. في حال وجود عدد كبير من الاستعلامات على جزء من البيانات (مثلا الاستعلام عن كتاب معين، في حال كانت قاعدة البيانات تخزن كتب) ما هو التصميم المناسب لمنع الاختناق واستيعاب هذا العدد الكبير من الاستعلامات؟

لمنع الاختناق واستيعاب عدد كبير من الاستعلامات حول جزء معين من البيانات، يمكن استخدام أساليب التصميم التالية:

1. التخزين المؤقت **caching** : يمكن تنفيذ التخزين المؤقت لتخزين في الذاكرة لجزء البيانات الذي يتم الوصول إليه بشكل متكرر. عند تلقي استعلام عن هذا الكتاب (الكتاب الذي يحدث عليه استعلام بشكل كبير)، نقوم بالتحقق مما إذا كانت بيانات الكتاب متوفرة في ذاكرة التخزين المؤقت. إذا كان الأمر كذلك، نقوم بتقديم الاستعلام مباشرة من ذاكرة التخزين المؤقت، وتجنب الحاجة إلى الوصول إلى قاعدة البيانات. يمكن أن يؤدي التخزين المؤقت إلى تحسين أوقات استجابة الاستعلام بشكل كبير وتقليل الازدحام في قاعدة البيانات.

2. التقسيم الأفقي (**horizontal partitioning or sharding**) باستخدام التجزئة: نقوم بتطبيق التجزئة لتقسيم البيانات أفقياً عبر أجزاء متعددة **shards**. من خلال توزيع بيانات الكتب عبر أجزاء مختلفة بناءً على مفتاح التقسيم، يمكن توزيع حمل الاستعلام بين خوادم متعددة. يمكن لكل جزء التعامل مع مجموعة فرعية من الكتب، وسيتم توجيه الاستعلام عن كتاب معين إلى الجزء المناسب. يساعد التقسيم على توسيع نطاق النظام وتجنب الازدحام في قاعدة بيانات واحدة.

3. النسخ المتماثل **replication** : يمكن استخدام النسخ المتماثل للجزء **shard** الذي يحدث عليه استعلام كبير من قاعدة البيانات لإنشاء نسخ متماثلة متعددة من هذا الجزء من البيانات. يسمح النسخ المتماثل بتوزيع حمل الاستعلام عبر نسخ متماثلة متعددة، مما يزيد من قدرة النظام على التعامل مع الاستعلامات المتزامنة. يمكن لكل نسخة متماثلة أن تخدم استعلامات القراءة، بينما يمكن توجيه استعلامات الكتابة إلى نسخة متماثلة أساسية. بهذه الطريقة، يمكن استيعاب الحجم الكبير من استعلامات القراءة دون ازدحام قاعدة البيانات الأساسية. (يجب الأخذ بعين الاعتبار العبء الناتج عن تحقيق الاتساق بين هذه النسخ **consistency**).

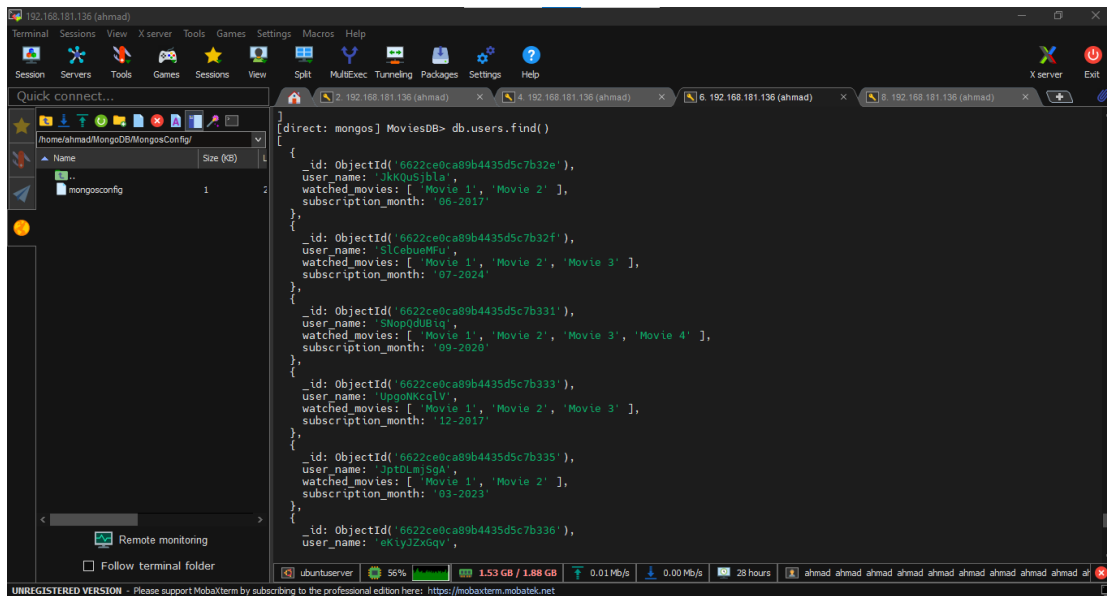
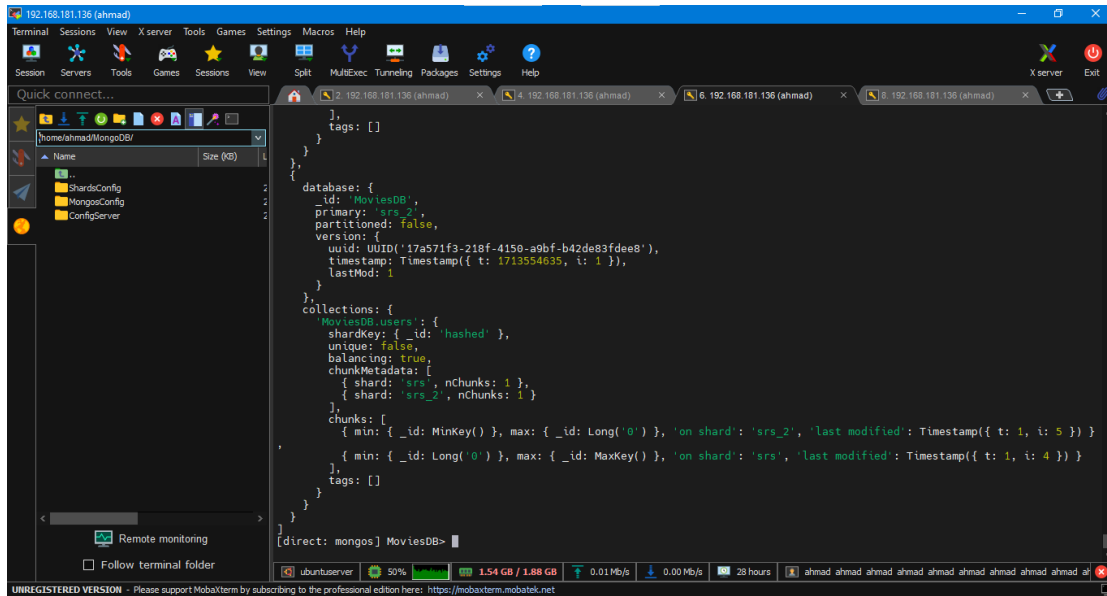
4. موازنة التحميل: استخدام موازن الحمل **load balancer** لتوزيع الاستعلامات الواردة عبر خوادم قاعدة بيانات متعددة. يمكن لموازن الحمل توزيع الاستعلام بالتساوي على العقد أو الأجزاء المختلفة، مما يمنع الازدحام على أي خادم. تضمن موازنة الحمل أن يتم توجيه كل استعلام إلى خادم متاح، مما يسمح للنظام بمعالجة عدد أكبر من الطلبات.

5. تحسين الاستعلام **indexing** : تحسين الاستعلام واستخدام الفهارس الملائمة يساهمان في تسريع استعلامات البيانات المرتبطة بالكتب المحددة. الفهرس هو هيكل بيانات يتم إنشاؤه على مجموعة محددة من الحقول في قاعدة البيانات، ويمكن استخدامه للعثور بسرعة على السجلات التي تطابق شروط الاستعلام. باستخدام الفهارس المناسبة، يمكن الوصول السريع إلى البيانات وتقليل الحاجة إلى فحص كل سجل في قاعدة البيانات. يمكن إنشاء فهرس على الحقل "اسم الكتاب"، مما يسهل البحث بشكل فعال عن الكتب باستخدام أسمائها. تحسين الاستعلام واستخدام الفهارس المناسبة يقللان وقت الاستجابة للاستعلامات ويحسنان أداء النظام بشكل عام، مما يتيح للنظام التعامل بكفاءة مع عدد كبير من الاستعلامات المتزامنة حول الكتب المحددة.

الجزء العملي:

ضمن القسم العملي تم إعداد ما يلي:

- Config server replica set: يضم ثلاث عقد.
- Shard replica set: اسمه srs مكون من ثلاث عقد.
- Shard replica set: اسمه srs_2 مكون من عقدتين.
- Mongos router.



الطالب: علي حسان سعيد.