
一般口演

一般口演2

病院情報システム2（ハードウェア，インターフェース）

2018年11月23日(金) 10:15 ～ 11:45 F会場 (5F 502+503)

[2-F-1-2] 多分散型電子カルテの基礎的研究

○本谷 崇之, 竹村 匡正（兵庫県立大学院応用情報科学研究科）

現在、電子カルテは診療情報の保存と閲覧に広く使用されている。一般的な電子カルテは中央サーバーに依存する集中型で構築されている。中央サーバーに対して依存するこの構成は、データ容量が増加した場合に備えた先行投資が必要となる。また障害が起きた場合、電子カルテの機能全体に影響を及ぼしてしまう。一方、データを集中的に管理するのではなく、複数のデータベースに分散して管理する多分散型データベースが存在する。この多分散型データベースは、内部的にリソースの追加が容易に行なえる。また、多分散型データベースはデータを水平に分散することができるため、データベース内のリソースを均一に保持することが可能である。ところで、集中型で構成されている電子カルテでは患者を軸にしたデータの保存を行っている。そのため、集中型で構成されている電子カルテの各オーダー情報は多分散データベースのインスタンスを捉えることが可能と考えられる。よって、各オーダー情報の量に対するリソースを最適化でき、リソースが不足している場合は追加することが可能となる。また、多分散型データベースのインスタンスに障害が発生した場合、該当するインスタンス以外は利用可能であるため、電子カルテへの障害の範囲が電子カルテの機能全体ではなく一部に留めることが可能になる。そこで本研究では、多分散データベースを用いた電子カルテ、すなわち多分散電子カルテの実現可能性の検討を行った。具体的には、今回は多分散型データベースである Redisを用いて多分散電子カルテ内のオーダーごとに分散されたデータを一般的な電子カルテと同様に、患者を軸に取得しその後、データの追加や編集を行った。この一連の動作によって、データの一貫性があるか検証を行った。

多分散型電子カルテの基礎的研究

本谷崇之、竹村匡正

兵庫県立大学大学院応用情報科学研究科

Fundamental research of multi-distributed EMR

Takayuki Mototani, Tadamasa Takemura

Graduate School of Applied Infomatics, University of Hyogo

EMR that are composed of centralized type is preserving data centered on patients. Therefore, it is considered that each order information of the centralized EMR can be regarded as an instance of the distributed database. Resources for the amount of each order information can be optimized, and if resources are insufficient, it can be added. As a result, it is possible to scale the storage capacity against the amount of generation of the order. And, if a distributed database instance fails, it is available except for the corresponding instance. The scope of obstacles to EMR can be stopped in part, not the entire function of the EMR. Therefore, in this research, we aim to investigate the feasibility of EMR using distributed database divided into each order information, that is, Multi-distributed EMR. Specifically, using Redis which is a distributed database, instances were created for each order information, and the data was distributed horizontally. The data dispersed for each order information was acquired on the patient axis similarly to general EMR, and then data was added and edited. Through this series of operations, it was verified whether the data was consistent.

Keywords: EMR, Redis, Database,

1. 背景

現在、電子カルテシステムは診療情報の保存と閲覧に広く使用されている。一般的な電子カルテシステムは、リレーショナル・データベースを用いて、中央管理のサーバを用いて構築されている。昨今の電子カルテシステムが扱うデータ量が增大するに伴い、ソフトウェア、ハードウェアの機能向上を持っても、アクセススピードの低下や、システム自体が高価格になる問題点がある。また、障害が起きた場合、電子カルテの機能全体に影響を及ぼしてしまう。また、中央サーバに対して依存するこの構成は、データ容量が増加した場合に備えた先行投資が必要となる。現在の電子カルテシステムのこのような状況は、現状のパフォーマンスを維持するために保守的に扱われることが多く、医療機器等のデータを扱わなくてはならないと考えられる院内 IoT 環境の実現に対して、制約条件になってしまう可能性がある。

一方で、NoSQL データベースに見られるような、Key-Value 型のデータ構造を利用して、大規模なデータを高速に取扱うことが可能なアーキテクチャが提案されている。また、これらは分散アーキテクチャの考え方に基づいて構築されることが多く、複数のデータベースサーバやノードが停止した場合でも、システム内の他のノードがデータを失い、システム全体が停止することはなく運用を継続することが可能であり、耐障害性が高いと考えられる。また、分散アーキテクチャでは、関連したデータが別々のデータベースに保存されるため、データの拡張性があり、規模に応じた設備投資を行うことが可能である。しかし、一般的に CAP 定理として言われるように、分散データベースのように可用性 (A: Availability) と分断耐性 (Partition-tolerance) を満たすものの、一貫性 (C: Consistency) を満たさない場合など、電子カルテシステムに適用可能なのか、という問題がある。しかし、昨今の NoSQL データベース、また分散データベース製品では可用性に制限を設けて一貫性、分断耐性を保持するものもあり、電子カルテシステムの基盤として検討するに値すると思われる。

2. 目的

そこで本研究では、NoSQL 型の分散型データベースの電子カルテへの適用について検討を行う。

3. 方法

3.1 電子カルテシステムと分散型データベース

電子カルテシステムは、基本的に患者を軸にしたデータ構造である。これに対して、各オーダーや SOAP 記載等の項目が紐付いてデータを構成していることから、各オーダー情報をクラスターとして構成する。また、患者ごとのデータをクラスターとして保持することで、電子カルテシステムにおける表示速度を上げることが可能であると考えられる。概念図を図1に示す。

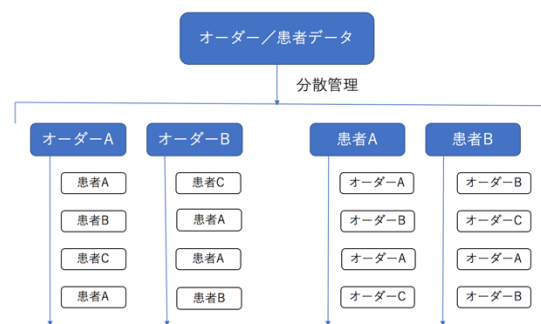


図2 分散型データベースによる電子カルテの概念図

3.2 Redis

今回分散型データベースとして Redis を用いて構成した。Redis は NoSQL に分類されるインメモリ型のデータベースである。インメモリ型のデータベースとは、コンピュータのメインメモリ上でデータの管理を行うことである。しかし、Redis には一

一般的なリレーショナルデータベース同様、データの永続化機能も備わっている。さらに、リレーショナルデータベース同様トランザクションを実行することが可能であり、データの衝突や競合に対して安全に処理が行える。

一方、リレーショナルデータベースと異なり、Redis でのデータ保存形式は、Key-Value ストアという形式をとっている。Key-Value ストアとは、キーと値の組み合わせを指定して保存する形式であり、データを二次元表で管理するリレーショナルデータベースとは大きく異なる特性がある。

また、Redis はデータの水平分散と容量追加が容易に行える特徴がある。通常のデータベースでは、保存できる容量はデータベースが存在するサーバに依存する。しかし、Redis の場合は一つの Redis サーバをノードとして捉え、複数の Redis ノードを一つに束ね、クラスタを構成することが可能であり、ノードを追加することで、容量の追加を可能にしている。Redis クラスタ内部では、クラスタを構成している複数のノードに対し、データを水平に分散する。こうすることでクラスタ内部での容量の偏りを無くすることが可能となっている。一度構成した Redis クラスタは、ノードの削除や追加が容易に行える。これより、容量限界が予想された場合、随時追加を行うことが可能になる。

さらに、Redis クラスタからデータを取得する場合、構成している Redis サーバのいずれか一つに接続することで取得することができる。これは、データを水平に分散し、リソースを均一に保持することと矛盾しているように考えられる。しかし、Redis クラスタは、接続した Redis サーバに要求されたデータの Key が存在しなかった場合、Key を保持している Redis サーバへ接続をリダイレクトする。これにより、Key を保持していない Redis サーバへデータを要求しても取得することが可能となっている。

3.3 分散型データベースを用いたシステム概要

今回構築したシステムの概要図を図2に示す。

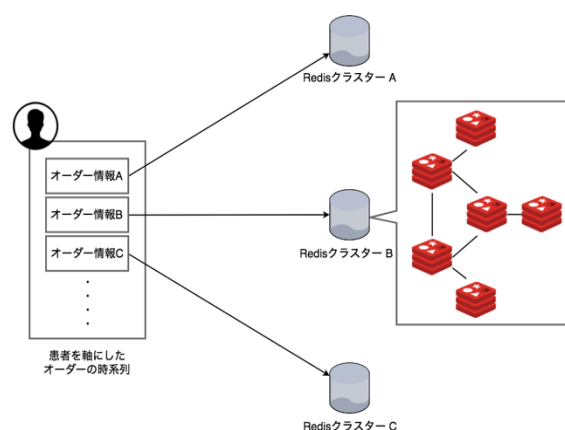


図2 分散型データベースを用いたシステムの概要図

今回構成したシステムでは、オーダー情報ごとに分散型データベースである Redis を用いてデータの保存を行った。Redis のパフォーマンスを最大限に発揮させるため、Master と Slave の構成を持ったクラスタ構成とした。こうすることで、データを取得する際は Slave から参照されるため、Master の負荷を Slave へ分散できる。また、オーダー情報ごとの Redis クラスタとは別に、オーダー情報ごとの Redis クラスタ同様に、患者の Redis クラスタを作成した。このクラスタでは、患者のタイムラインとして、患者へのオーダーの Key 一覧を日付ごとの配列で

保持するように構成した。

3.4 システム環境と評価

今回構成したシステムを模擬電子カルテとしてシステムの評価を行なった。構築したシステムの環境を下記に示す。

表1 システム環境

OS	Mac OS X 10.12.5
メモリ	16GB
HDD	2TB
プロセッサ	2.4 GHz Quad-Core Intel Xeon

今回は、構築したシステムに対して、レコードを蓄積させ、レコードの取得とレコードの書き込みのパフォーマンス測定を行なった。これは、通常の電子カルテ同様に、患者を軸に時系列での取得を行い、その後オーダーの追加を行うことを想定しているものである。蓄積されたレコードを5億レコードとし、測定項目として、レコードの取得と書き込みに対し、単位時間あたりのリクエストの実行数を測定した。

4. 結果

患者を軸に時系列でデータを取得し、その後オーダーの追加を行う一連の動作を 1000 レコードの読み込み処理と書き込み処理で測定したところ、秒間あたりのリクエスト実行速度が 4103.5491(request/sec)、2000 レコードで測定したところ 4135.8091(request/sec)、4000 レコードで測定したところ 4129.6428(request/sec)の結果となった。

5. 考察

結果より、今回構成したシステムは、一連のレコードの取得と書き込みに対して、高いパフォーマンス性能を発揮しており、電子カルテとしては十分な結果が得られたと考えられる。すなわち、各オーダー情報に分散した Redis クラスタを用いた電子カルテシステムにおいては、データ取得と書き込みに関しては、高いパフォーマンスを期待できる。

また、Redis の特性に見られるように、患者の増加やオーダーの種類の増加、また今後の院内 IoT 環境の実現による機械的なレコード数増加に対しても、対応が可能になると考えられる。

一方で、リレーショナルデータベースでは一つのレコードに対して複数のクライアントが書き込みを奪い合う場合が想定される。その際にトランザクションを用いて該当するレコードに対して安全に処理を行なっている。今回用いた Redis も同様に、トランザクション処理は行えるが、データベースに不整合を起さない排他制御についても、「悲観的排他制御」「楽観的排他制御」を考慮してトランザクション処理を実装することが可能であり、これらの実装が電子カルテシステムとして利用可能かについては、今後検討を行う。

参考文献

- 1) Redis Documentation. Redis.io
[<https://redis.io/documentation> (cited 2018-Sept-3)].
- 2) Pengcheng Huang, Zuofei Wang. Redis 4.x Cookbook. Packt Publishing, 2018