3T-05

パブリッククラウドにおける 広域 State Machine Replication の特性評価

沼倉　正太†　　中村　純哉††　　大村　廉†††
豊橋技術科学大学†　豊橋技術科学大学††　豊橋技術科学大学†††

1. はじめに

State Machine Replication (SMR) [1] はシステムの耐故障性を向上させる技術として広く使用されている．SMR では，サーバクライアントモデルで通信するシステムにおいてレプリカと呼ばれるシステムのコピーを複数作成し，レプリカ間で処理順序を同期する．これによって，一部のレプリカに故障が発生してもシステム全体では正常動作を維持できる．

世界中にレプリカを配置して構成する SMR は**広域 State Machine Replication**（広域 SMR）と呼ばれる．広域 SMR は地震等の大災害に耐えられる高い耐障害性を備えたサービスを実現できることから，高い注目を集めている．既存研究 [2]では，広域 SMR がシステムの応答速度に与える影響について，システムの通信をモデル化した際の通信回数や，複数のレプリカを統率するリーダレプリカの配置などに着目した性能評価が行われた．

パブリッククラウドは，利用者の要望に応じて仮想マシンなどの計算資源をオンデマンドに貸し出すサービスである．仮想マシンを動かすデータセンタはリージョンと呼ばれ，世界各地から利用者が選択できる．

広域 SMR は，パブリッククラウドを利用することで従来よりも容易に実現できるようになると期待されている．しかしながら，パブリッククラウドを広域 SMR で実現した際の性能特性や課題について，これまで十分な研究が行われていない．本研究では，パブリッククラウド特有の性質であるリージョンの選択がレプリケーション性能に与える影響について，複数のレプリカ配置における性能を計測することでその特性を明らかにする．

2. State Machine Replication

SMR はサーバクライアントモデルを対象としたレプリケーション手法であり，複製対象のサーバは状態機械（State Machine）として記述される．サーバは複数台のレプリカに複製され，各レプリカは状態機械を実行する．SMR は次の順序でリクエストを処理する．

1. クライアントは全てのレプリカにリクエストを送信する．
2. レプリカは，SMR を構成するレプリカ同士で**ビザンチン合意**を行い，リクエストの処理順序について合意する．
3. 各レプリカはリクエストを実行し，状態機械の状態を更新する．
4. レプリカはクライアントにリクエストの実行結果を返す．

図 1 に代表的な SMR ライブラリ BFT-SMaRt [3]の通信パターンを示す．

故障モデルの一つに Byzantine 故障がある．Byzantine 故障したレプリカはウイルスや悪意のあるユーザからの攻撃によりシステムの動作が予測不可能となる．Byzantine 故障への耐性を持つ SMR は Byzantine Fault Tolerance (BFT)と呼ばれる．

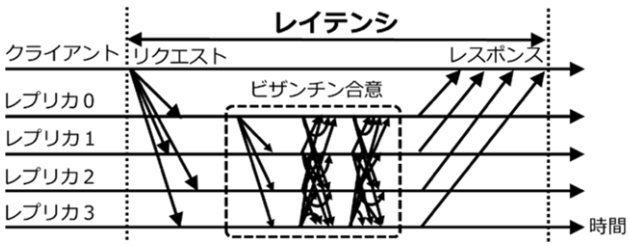

図 1. BFT-SMaRt 通信パターンの概要

3. 実験：利用リージョンによる影響

ここでは，広域 SMR を構成するクライアント及びレプリカの配置がレプリケーションに与える影響について，複数の配置で性能を計測することにより調査する．

3.1. 実験設定

評価対象の SMR 環境は，Java 言語で記述されたオープンソースの SMR ライブラリ BFT-SMaRt [3]を用いて構築する．実験では代表的なパブリッククラウドサービスである Amazon EC2 を用い，インスタンスタイプは t2.micro を使う．実験では 4 箇所のリージョンに，それぞれ 1 台の仮想マシンを起動する．各仮想マシンではレプリカ及びクライアントのプログラムを実行する．レプリカ及びクライアントのプ

Characterization of Geo State Machine Replication in Public Cloud
† Shota Numakura, Toyohashi University of Technology
†† Junya Nakamura, Toyohashi University of Technology
††† Ren Ohmura, Toyohashi University of Technology

ログラムは，BFT-SMaRt に含まれるベンチマークプログラム LatencyClient 及び LatencyServer を，それぞれ用いた．また，評価指標としてレイテンシ（クライアントのリクエスト送信からレスポンスが帰ってくるまでの経過時間．図 1 参照）を用いる．

本実験ではリージョンの距離関係がレイテンシへどの様な影響を与えるか調査を行うため，3 種類のリージョンの組み合わせで測定を行う（表 1）．それぞれ，組み合わせ(1)は既存研究[2]の実験 E で行われたリージョンの組み合わせ，組み合わせ(2)は組み合わせ(1)の Oregon リージョンを最寄りの California リージョンに変更した組み合わせ，組み合わせ(3)は組み合わせ(1)の São Paulo リージョンを地理的に離れた Tokyo リージョンに変更した組み合わせである．実験結果を図 2 に示す．

表 1. 実験で利用するリージョンの組み合わせ

	利用するリージョン
組み合わせ(1)	Ireland, São Paulo, Oregon, Sydney
組み合わせ(2)	Ireland, São Paulo, California, Sydney
組み合わせ(3)	Ireland, Tokyo, Oregon, Sydney

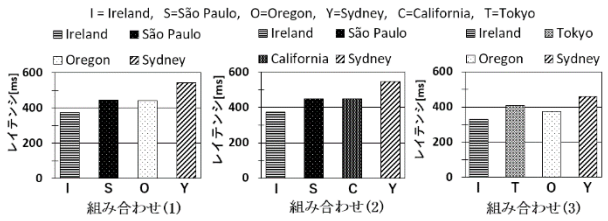

図 2. 各組み合わせの実験結果

3.2. 考察：既存研究と同じリージョンの組合せ

組み合わせ(1)の結果と，同じリージョンの組み合わせで実施された既存研究[2]の実験（3.E 節，図 7）との結果を比較したところ今回の実験結果ではレイテンシが全体的に低下した．原因の一つとして，実験に使用したインスタンスタイプが違うことが挙げられる（既存研究[2]では t1.micro を，本実験では t2.micro を使用した）．t2.micro はメモリ容量や CPU 処理性能が向上している．また，既存研究[2]の実験当時よりも本実験を実施した 2017 年では，リージョン間の通信速度が向上したことも実験結果に差が生じた理由の一つとして予想される．

3.3. 考察：リージョン毎のレイテンシの差

表 1 に示す各組み合わせで用いた 4 つのリージョンで計測したクライアントのレイテンシでは，一部のリージョン間で大きな変化が見られた．例えば組み合わせ(1)の Ireland と Sydney を比較すると，およそ 150ms 程度の差が生じた．

この原因を詳しく調査するため，今回使用した BFT-SMaRt の通信パターンに着目した（図 1 参照）．通信パターンは大きく 3 つの段階に分類できる．1 つ目はクライアントがレプリカへリクエストを送る**リクエスト送信段階**，2 つ目はリクエストがレプリカに届きレプリカ間でビザンチン合意が行われる**ビザンチン合意段階**，3 つ目はレプリカからクライアントへレスポンスが返される**レスポンス送信段階**である．

クライアントの位置が変わると，リクエスト送信段階及びレスポンス送信段階にかかる時間は変化する．リクエスト送信段階及びレスポンス送信段階にかかる合計の時間はある 2 つのリージョン間の Round Trip Time (RTT)に近似できる．そこで，図 2 の実験結果と実験で利用したリージョンの RTT の比較を行うと，RTT がレイテンシの測定結果へ影響を及ぼしていることが示唆された（表 2）．

表 2. Round Trip Time 測定結果[ms]

		受信側					
		I	S	O	Y	C	T
送信側	I	0.3	183.0	139.0	285.3	150.6	225.0
	S	183.0	0.4	181.0	330.0	181.0	269.0
	O	137.0	181.0	0.8	161.6	21.8	103.0
	Y	285.0	330.0	161.6	0.5	152.6	103.3
	C	144.6	181.0	21.8	152.6	0.3	113.0
	T	225.0	269.0	103.0	103.0	113.0	0.3

I:Ireland S:São Paulo O:Oregon Y:Sydney C:Carifornia T:Tokyo

4. まとめと今後の課題

本研究では，パブリッククラウド上に構築する広域 SMR において，リージョンの地理的な位置等のパブリッククラウド特有の性質がレプリケーションに与える影響について調査するため，3 つのリージョンの組み合わせを用意し，レイテンシの評価実験を行った．その結果，RTT やビザンチン合意がレイテンシの測定結果に影響をおよぼすことを明らかにした．

謝辞

本研究は JSPS 科研費 16K16035 の助成を受けて実施されました．

参考文献

[1] F. B. Schneider, Implementing fault-tolerant services using the state machine approach: A tutorial, ACM Computing Surveys, Vol. 22, No. 4, 299-319, 1990.

[2] J. Sousa and A. Bessani, Separating the WHEAT from the Chaff: An Empirical Design for Geo-Replicated State Machines, In Proc. of Symposium on Reliable Distributed Systems (SRDS), 146-155, 2015.

[3] A. Bessani, J. Sousa and E. P. Alchieri, State machine replication for the masses with BFT-SMaRt, In Proc. of the International Conference on Dependable Systems and Networks (DSN), 355-362, 2014.