卒業論文 20xx 年度 (令和 xx 年)

RG における卒論 LaTeX テンプレート 2019

慶應義塾大学大学院 政策・メディア研究科
阿部涼介

卒業論文要旨 - 20xx 年度 (令和 xx 年度)

RG における卒論 LaTeX テンプレート 2019

近年，書き始めが近年な論文が多い．ちゃんと特徴的な事象があるならそれを挙げて，“近年” なんて曖昧ワードを使うんじゃない馬鹿者．

キーワード:
1. 卒業論文, 2. 村井純研究室, 3. RG, 4. LaTeX

慶應義塾大学大学院 政策・メディア研究科
阿部涼介

Abstract of Bachelor's Thesis - Academic Year 20xx

LaTeXTemplate for RG 2019

I can't write English.

Keywords :
1. Thesis, 2. RG, 3. Jun Murai Lab., 4. LaTeX

Keio University Graduate School of Media and Governance
Ryosuke Abe

目 次

図 目 次

表 目 次

第1章　序論

本章では本研究の背景，課題及び手法を提示し，本研究の概要を示す．なお，Bitcoin [1] は関係ない．

1.1　地理的分散システムの発達

本節では，地理的分散システムの概要とその発達について説明する．

1.1.1　地理的分散システム

地理的分散システムとは主に P2P ネットワーク上で動作するシステムを指す．従来のサーバ・クライアント方式では，クライアントがサーバに対してリクエストを投げ受け取ったサーバが特定の処理をした上でクライアントに何らかのレスポンスを返す．対して P2P サービスでは，参加するノードはサーバでもありクライアントでもある。クライアントとして他のノードに対してリクエストを送信したり，逆に他のノードからのリクエストに対してレスポンスを返す．サーバ・クライアント方式に対して，耐障害性・冗長性・可用性において優れている．

1.1.2　地理的分散システムの発達

2000 年代初頭，Winny や Gnutella といった P2P アプリケーションが頭角を表した．それまでサービスの構成として一般的であったサーバ・クライアント方式とは異なるそれぞれのノードが対等な関係をもつ P2P アプリケーションについて注目が集まったが，サーバ・クライアント方式に置き換わるまでの隆盛はなく後退していった．しかし 2008 年，satoshi nakamoto により Bitcoin のために開発されたブロックチェーン技術が登場することによって再度地理的分散システムへの注目が集まり，開発や研究の勢いが盛んになってきている．

1.2　本研究の着目する課題と目的

本節では，本研究で着目しているステージング環境の概要とその必要性，ならびに地理的分散システムのステージング環境を構築する際の課題点について説明し，最後に目的を明確化する．

1.2.1 ステージング環境

ステージング環境とは，本番環境での運用をする前に実際の環境を想定してシステムの動作確認を行うための環境である．開発社が実際に開発を行うローカル環境と実運用する本番環境では，度々環境の差異から動作の違いが生じ，ローカル環境で正常に動作していたものが本番環境に反映した途端動作しなくなるといった事象が発生する．そのような自体を防ぐためにローカル環境と本番環境の間に，本番環境を想定したステージング環境を構築し，一度ステージング環境にて動作を確認することで予想外の障害が発生するリスクを抑えることができる．

1.2.2 ステージング環境の必要性

システムである以上，地理的分散システムにも本番環境での予想外の障害に備えたステージング環境が必要である．従来のサーバ・クライアント方式の場合であれば，開発者が任意に使用するデータセンターもしくはクラウド環境を選択しデプロイ作業を行えば，比較的簡単にステージング環境を構築することが可能できた．しかし参加ノードの物理的位置を開発者側が操作することが出来ない地理的分散システムでは，物理的に離れた地点にノードを設置し，実際のインターネット環境上で動作確認を行う必要性がある．

1.2.3 ステージング環境の構築における課題点

地理的に分散した地点間でステージング環境を構築するには，コミュニケーションコストとヒューマンリソースのオーバーヘッドが課題点になると考えられる．ステージングでの動作確認のためには，それぞれの地点でのノードの設置・サービスの動作確認に必要なリソースの共有・ステージング環境へのデプロイ作業，修正が発生した場合のアップデート作業といったタスクとそれらに伴う開発者同士のコミュニケーションが必要となるからである．コミュニケーション不足によるミスや単なる人為的ミスが発生する可能性もあり，地理的分散システムのステージング環境の構築は困難であると考えられる．

1.2.4 目的

本研究では，地理的分散システムの動作確認を行うためのステージング環境を世界規模のコンピュータネットワーク上に構築するための手法を提案することを目的とする．

1.3 本研究の仮説

コンテナオーケストレーションツールとして開発された Kubernetes は，アプリケーションのデプロイやスケーリングを自動化し，コンテナ化されたアプリケーションを統合管理するためのシステムである．Kubernetes では使用するサーバをクラスタリングし，実際

にコンテナを動かすサーバはワーカーノードと呼ばれる．マスターノードはワーカーノードに対して一斉にコンテナをデプロイしたりスケーリングしたりすることが可能である．

そこで本研究では，地理的分散システムに参加するノードを Kubernetes によってクラスタリングすることで，で述べた課題点を解決し，で述べた世界規模のコンピュータネットワーク上に地理的分散システムのためのステージング環境を構築するという目的を達成できるのではないかと考えた．

1.4 本研究の手法

本研究では，地理的に分散したノード間で OpenVPN オーバーレイネットワークを形成することでノード同士が別セグメントに位置していたとしても相互に通信可能な状態にし，その上で Kubernetes クラスタを構築した．OpenVPN とは VPN ネットワークを構築することができるオープンソースソフトウェアである．別々のセグメントに位置するノード同士で Kubernetes クラスタを構築し，コンテナ化したアプリケーションをデプロイできていることを確認し，本システムの課題点を解決できているか推定することで要件を満たせることを確認した．

1.5 本論文の構成

本論文における以降の構成は次の通りである．

2 章では，地理的分散システムとその実験的運用方法およびそれに伴う課題点について議論し，本研究の背景を明確化する． 4 章では，本研究で着目する問題を解決するための要件，仮説と手法について説明する． 5 章では， 4 章で述べた提案手法について述べる． 6 章では，2 章で求められた課題に対しての評価を行い，考察する． 7 章では，本研究のまとめと今後の課題についてまとめる．

第2章　背景

本章では本研究の背景について述べる.

2.1　地理的分散システム

本節では，地理的分散システムの具体例ついて概説する.

2.1.1　Winny

Winny はソフトウェアエンジニア金子勇氏が開発し、2002 年に発表されたファイル共有ソフトである。システム上で中央集権的なサーバを保持せず、ノード同士が相互に接続することで実現される P2P アプリケーションとして注目を浴びた。ユーザはノード内に保持されたファイルを他のノードと共有することができるため、任意のファイルをアップロードしたり、逆に他のノードが保持しているファイルをダウンロードすることができる。Winny では、受信ファイルの送信元や送信ファイルの宛先をユーザが確認することはできず、バックグラウンドでの処理はユーザに見せないよう高い秘匿性が担保されていた。従来のサーバ・クライアント方式のシステムアーキテクチャとは打って変わって出た新しい形のアプリケーションであったが、高い匿名性も起因して、一部のユーザが違法な音楽ファイルや動画ファイル、コンピュータウイルスを Winny にアップロードしたことで著作権法違反が問われた。開発者である金子氏にも疑いがかけられ 2004 年に逮捕、その後画期的な発明であった Winny も衰退していった。

2.1.2　Gnutella

Winny に同じく Gnutella も中央集権型サーバに依存せず、P2P ネットワーク上のノード間の通信のみでファイルを送受信を行うファイル共有アプリケーションである。

2.1.3　Bitcoin

Bitcoin は 2008 年に Satoshi Nakamoto と名乗る人物によって論文にて提唱されたものである。2009 年にはソフトウェアとして実現されており、今では多くのユーザに使用されている上、仮想通貨の先駆けとして他の仮想通貨を生む大きな起点となった。同時に、

2000 年代後半に勢いを失っていた P2P システムの存在を再度世に知らしめ、開発の促進を促す起爆剤の役割を果たしたと考えられる。Bitcoin は基盤技術のひとつとして Winny や Gnutella と共通する P2P ネットワークを採用している。参加するノードはそれぞれがシステム上のデータを保持し相互にデータを検証しあうことで、第三者的監視機関を必要とせずにデータの堅の牢性を担保することが可能である。

2.2 地理的分散システムの技術

本節では，地理的分散システムを実現するための技術について概説する．

2.2.1 P2P

P2P とは”Peer to Peer”の略であり、ピアはコンピュータのことを指す。P2P ではコンピュータ同士が対等な関係を築いており、従来のサーバ・クライアント方式で中央集権的な役割を担うサーバを必要とせずに成り立っている主従関係のないシステムモデルである。サーバ・クライアント方式では、クライアントがリクエストを投げサーバがレスポンスを返すという明確な役割分担がなされている。そのためサーバはクライアントからのリクエストが来るまで何もせず、リクエストが来たときのみ必要な処理を行ってクライアントへ返答する。逆にクライアントはサーバに問い合わせる必要がないときは何もせず、データを要求したり変更する必要が生じたときのみクライアントとの通信を行う。よって通信は常にクライアントが起点となり、基本的にサーバ起点の通信は行われない。対象的に、P2P ではそれぞれのサーバが対等な関係で成り立っており、サーバ・クライアント方式のような明確な役割分担はシステム上されない。何故なら P2P では各ピアが状況に応じてサーバとクライアントの役割を担うからである。固定的な役割がない代わりに、臨機応変にピアがサーバとしてレスポンスしたり、クライアントとしてリクエストを投げたりと臨機応変な動的システムが特徴としてあげられる。サーバ・クライアント方式では、リクエストを発信する側をクライアント、それに対してレスポンスを返す側をサーバと呼んでいるが、P2P では前述した通り各ピアは動的に役割を変化させサーバとしてもクライアントとしても動くことからサーバントと呼ばれる。単にノードと呼ばれることもある。

2.2.2 従来のサーバ・クライアント方式との違い

2.3 地理的分散システムとステージング環境での動作確認

本節では、地理的分散システムのステージング環境と動作確認について概説する。

2.3.1 最小限の動作確認

最も簡単に行える動作確認は、ふたつのノード間で行うテストである。ネットワーク上の二点でそれぞれノードを立ち上げ、システムの機能が正しく動作するかを確認する。従来のサーバ・クライアント方式では、最低限ではあるが機能の保証ができる。サーバ・クライアント方式では、中央集権的サーバとクライアントが一対一の関係で繋がっており、開発者はクライアントとの通信ただひとつに注力すればいいからだ。ユーザが増加した場合の障害対策やレスポンスタイムの向上は確かに必要であるが、サーバとクライアントの一対一の関係性は不変であるため、ネットワーク自体が正常で有る限り問題は二点間に閉ざされておりテストがしやすい。一方、中央集権的サーバがなくノード同士がサーバにもクライアントにもなり得る地理的分散システムでは、この方法は十分ではない。ネットワークに参加するノードが増加すれば個々のノード同士の関係性は変化し、関係性が固定されないためである。もうひとつの理由として、ノード周辺のネットワーク環境によって動作に影響が出る可能性が考えられる。地理的分散システムの具体例として挙げたBitcoinでは、参加するノードは全て同じデータを保持する。データの送信や受信において遅延が発生すれば何らかの影響が出ることは簡単に予想可能である。例えシステム上でデータの不整合を防ぐロジックが組まれていたとしても、ロジックを表現したコードが実際の環境で正常な動作をすることを動作確認無しで担保することは難しい。以上の理由から、地理的分散システムの動作確認をするにあたって二点間でのステージング環境は不十分であり、より多くのノードを実際の世界規模のネットワーク上で動かしたステージング環境が必要であると考えられる。

2.3.2 地理的に分散したノードによる動作確認

上記で述べた通り、地理的分散システムのステージング環境は世界規模のネットワーク上で構築する必要性がある。しかしこの方法は、ステージング環境の構築ならびに動作確認の進行において多大なるコミュニケーションコストとヒューマンリソースが予想される。まず環境構築において離れた地点にノードを設置する必要性がある。地点ごとにノードを設置する人に加え、ノードのスペックやネットワークの構成等について共有するためのコミュニケーションが必要となる。必要な物理筐体が揃ったのち、地理的分散システム上で走らせるアプリケーションを各ポイントに配布し、各開発者は受け取ったアプリケーションファイルを設置したノードの上で走らせる必要がある。ステージング環境でシステムを走らせた後に関しては、機能面や性能面での動作確認を行い、修正箇所があれば開発者がパッチを適応した後、修正後のアプリケーションファイルを各ポイントに配布するところから再度やり直さなければならない。修正箇所が増加するに比例して、コミュニケーションコストと必要なヒューマンリソースは膨れ上がることが予想される。さらにコミュニケーションの不足や伝達ミス等の人的ミスにより理想的な動作確認が行えないケースも考えられる。以上の点から、地理的分散システムのステージング環境においてコミュニケーションベースの動作確認には多くの課題があり現実的に困難である。それ故、地理的

に分散したノードを任意のポイントから統合的に管理することによって各地点での作業や地点間でのコミュニケーションを削減する必要性があると考えられる。

2.3.3 独自実装のデバッグエージェントによる動作確認

既存の提案として、地理的に分散したノードを統合管理・操作するために別アプリケーションを独自で開発する手法がある。別アプリケーションとは、対象アプリケーションに対して命令を送信したり通信内容をログとして抽出するなどのデバッグエージェントして動作する。ノードを統合管理出来る点では要件を満たしており、コミュニケーションならびに工数の削減に繋がると考えられる。しかし対象アプリケーションにパッチを適用したい場合、同様にそれを操作するデバッグエージェントにも変更を加える必要があり、変更への弱さが窺える。アップデートへの柔軟性が不足している限り、それによって生じるオーバーヘッドを削減することが出来ず根本的な解決に繋がらないと思われる。分散したノードを一斉にコントロールだけでなく、アプリケーションの停止や更新といった変更においてもより少ない手間で抑えられることが求められ、それを満たした際に地理的分散システムの十分なステージング環境が成り立つと考えられる。

第3章　本研究における問題定義

第4章　提案手法

本章では提案手法について述べる.

4.1　概要

第5章　実装

本章では提案手法の実装について述べる.

5.1　概要

第6章　評価

本章では，提案システムの評価について述べる．

6.1　評価内容

第7章　結論

本章では，本研究のまとめと今後の課題を示す.

7.1　本研究のまとめ

7.2　本研究の課題

謝辞

俺に関わった全てに感謝

参考文献

[1] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system. http://www.cryptovest.co.uk/resources/Bitcoin%20paper%20Original.pdf, 2008.