はじめに

EC サイト、企業のキャンペーンサイト、ニュースサイト、ブログ、SNS、ソーシャルゲームといった各種の Web システムは、今や我々の日常生活と切っても切り離せないものとなりました。そして、それらは 24 時間 365 日稼働し続けること（可用性の高いシステムであること）が当然のように求められるようになっています。

一方で、実際にそれらのサービスが常に利用可能であるかというと、残念ながらそうではありません。想定内のユーザーアクセスによってさえ反応がなくなってしまう Web システムは、今なおあります。それは個人が構築、運営するシステムだけにとどまらず、大手の企業や行政のシステムにおいても多く見受けられるのが実情です。

Web サービスを構築し、提供する側の立場としては、できるだけ 24 時間 365 日応答可能なシステムを提供したいのですが、そのためにコストを度外視して設備投資してしまうと、ビジネスの継続そのものが不可能となってしまいます。事業の継続のためには、システムは利用者のリクエスト数の増減に応じてその時々で最適なシステムリソースの使用量を選択できることが必要です。こういったシステムは「スケーラブルなシステム」と呼ばれ、現在では Web システムに求められる重要な要件の 1 つとなっています。このようなスケーラブルなシステムを構築、提供することは、今やシステム構築者の責務であると言っても過言ではないでしょう。

オンプレミスの時代にはそういったシステムの構築は難しかったのですが、現在では各種クラウド事業者が提供するさまざまなサービスを適切に組み合わせることで、可用性や処理能力をシステム要件に合わせて選択して構築できるようになっています。しかし、それらクラウドのサービスを用いて構築したシステムであっても、ユーザーのリクエストに対して安定して応答を返すことは、いまだに非常に難しい課題です。その点は、クラウド導入前と変わりません。また、オンプレミスですべて自社管理のサーバにより提供するサービスとは異なり、クラウド上で構築されたシステムを利用する場合はシステムや各サービスの多くがユーザー側から手を出せない部分にあるため、性能改善のために具体的にどうすればよいのかわからないケースもあるかもしれません。

本書では、よりスケーラブルで可用性が高いシステムを構築するという Web サービス構築者の責務を果たすために必要なことを、「クラウドを利用した設計」および「負荷試験の実施」の両面から、前提知識、具体的な実践方法、ケーススタディなどを通じて解説しています。また、実際に負荷試験を効率的に実施するためには、「直感に反する条件」を設定する必要がある個所などがたくさんあります。負荷試験結果のレポートを読む立場の人にも、それらを設定した理由を説明しなければなりません。本書がその際に、エンジニアが自信を持って説明するための助けとなれば幸いです。

2017 年　晩夏　仲川樽八

本書の特徴

本書は数ある負荷試験用の攻撃ツールやモニタリングツールのうち一部を取り上げて紹介していますが、ツールの解説本ではありません。ツールだけでは解決しない負荷試験の課題に焦点を当てています。攻撃ツールには非常に多くの種類があり、次々と新しいものが登場しています。しかし、それらのツールをもってしても負荷試験の課題の多くは解決できません。プロジェクトマネジメントからOSレベルまで、広い範囲に課題が存在するからです。

本書ではWebシステムのサーバ側における負荷試験で発生する課題に対して、問題解決に近づくためのベストプラクティスをまとめています。また、クラウド上にシステムを構築する前提としています[注1]が、オンプレミスが前提であった時代からの運用開発経験に基づいており、クラウドに限らず、オンプレミスやオンプレミスとクラウドでのハイブリッド構成のシステムの開発、運用に関しても適用できる内容となっています。

負荷試験とPDCAサイクル

PDCAサイクルとは、Plan（計画）- Do（実施）- Check（評価）- Action（改善）を繰り返し行うことです。先行きの見通しが立てにくい事業活動などにおいて、継続的に改善する手法として取り入れられている考え方です。多くの事業活動やアジャイル開発と同様、負荷試験も通常一度の実施では終わらず、まさにPDCAサイクルを回しながら進めます。

負荷試験におけるPDCAとして最初に思い浮かぶのは、**試験全体の大きなPDCA**でしょう。

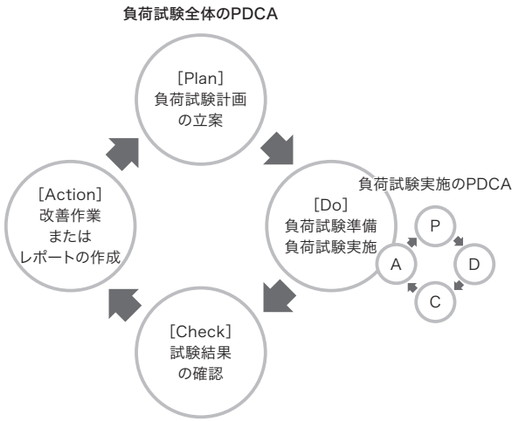
Plan　：負荷試験計画の立案
Do　：負荷試験の準備および実施
Check：計画時に立てた目標値と前提条件をクリアしたか、意味のある数字が取れたかの確認
Action：負荷試験レポートの作成、またはシステムの改善、目標値や前提条件の見直しをして次のPDCAにつなげる

通常、システムの「試験」というと試験項目書があり、その手順どおり進めて全項目が適合するかをチェックします。もし適合しない場合はシステムを修正する、という流れが一般的です。負荷試験も基本的には同じ流れですが、場合によっては要件や設計を見直したり、新しい試験項目を追加したりと、結果に応じて比較的ダイナミックに変化します。そのため、このPDCAは全体をいきなり回すには大きすぎて手戻りが発生し、いたずらに期間と労力を浪費するだけとなるリスクが高くなります。

本書においては、上記のPDCA中のDoにあたる「負荷試験の準備および実施」に関してさらに細かいチェックポイントを設けて、それぞれのチェックポイントをクリアするための細かいPDCAを回すことで、負荷試験をより効率的に実施することを目指しています。

注1　本書ではクラウドサービスとしてAmazon Web Services（AWS）を利用することを想定し、AWS用語を使用しています。主な用語は第11章「11.1　用語説明」を参照してください。

▼**図0-1**　負荷試験におけるPDCA



負荷試験は、単に仕様に基づく試験項目の合否だけを見るのではなく、試験を行う過程でシステムの内部と対話しながら状況に応じて手段を講じていく開発工程の一部でもあります。このような事前予測の難しい環境で、ある一定の性能要件を達成するのが負荷試験であり、小さなPDCAを積み立ててシステムを改善していくと考えるのは理にかなっていると言えます。

対象読者

本書は、Webシステム企画、構築、運用に関わる次のような方々を幅広く対象とし、実際に負荷試験を実施する立場にない方であっても、一般的なWebシステムに対する負荷について実用的な考え方を学べる書籍となることを意図しています。

- 安定したWebシステムの運用を必要とする立場の方
- 今まさにWebシステムの負荷対策に困っている方
- 負荷試験を初めて実施しようとしている方（負荷試験初心者）
- 負荷試験を過去に何度か実施しているが、その実施内容に自信のない方
- システム設計から構築まで、もう一度初心に戻って体系的に学習したい方

初心者や入門者だけでなく、非エンジニアであっても理解できるように、できるだけ丁寧な解説を心がけています。一方で、熟練したエンジニアでもハマってしまう負荷試験の問題を扱っており、負荷試験に関わる多くの方にとって有益な内容であると考えています。

各章の概要

第1章　間違いだらけの負荷試験とWebシステムの失敗事例

この章では、よくある開発の現場を想定しながら、"あえて"間違いだらけの負荷試験のケーススタディを紹介します。また、間違った結果により引き起こされた失敗事例も紹介します。

第2章　Webシステムの設計手法

可用性が高く、スケーラブルなシステムを構築するための設計について、オンプレミス時代からの設計手法と、クラウド登場以降の設計手法の両方を紹介します。

第3章　負荷試験の基本知識

負荷試験を行うための基礎知識や基本的な考え方について解説します。

第4章　負荷試験のツール

負荷試験を行うために必要なツールである、攻撃ツール、プロファイリングツール、モニタリングツールに関して、さまざまなツールがある中での共通的な考え方や選択基準、使用方法などを解説します。また、ここでは攻撃ツールを使った負荷生成と、実際のユーザーのアクセスによる負荷の違いを解説し、負荷試験時に攻撃ツールを使うと生じるものの実際には発生しない問題などを紹介します。

第5章　負荷試験の計画

負荷試験計画の立て方や、システムの目標値の設定方法を紹介します。

第6章　負荷試験の準備

負荷試験を実際に行うための事前準備を解説します。また、段取りを踏まない負荷試験を行った際に問題が発生しやすい部分を紹介します。

第7章　負荷試験の実施1 ～ 試験実施とボトルネックの特定

実際の負荷試験実施部分をさらに段取りごとに分解したPDCAサイクルとしてとらえて、実際にどういう順番でそのサイクルを回すとより効果的で手戻りの少ない試験を実施できるかを紹介します。

第8章　負荷試験の実施2 ～ 原因分析とシステムの改善作業

負荷試験実施により特定されたボトルネックごとに、該当のボトルネックが生じる原因を分析します。その後、分析した原因別にボトルネックの解消に向けてどういう対策が取れるのかを紹介していきます。

第9章　負荷試験レポートの作成

負荷試験の完了後に行うべきシステム構成の決定およびレポートの作成方法を紹介します。

第10章　負荷試験実践のケーススタディ

実際にAWS上で構築したアプリケーションに対してPDCAサイクルでの負荷試験を行い、負荷試験レポートをまとめるケーススタディを次の2つの事例で紹介します。

- PHP + JMeter + Xhprof
- Node.js + Locust + New Relic

第11章　巻末資料

用語説明

本書の図表中で断りなく利用しているAWSアイコンや、本文中の用語を解説します。

JMeterシナリオ解説

第10章で紹介した事例で利用したJMeterシナリオに関して、その作成方法や内容を解説します。

Locustシナリオ解説

第10章で紹介した事例で利用したLocustシナリオについて解説します。

間違いだらけの負荷試験 解説編

第1章「1.1　間違いだらけの負荷試験」のどこがどう間違いなのかを解説します。

謝辞

本書執筆のきっかけは、2015 年 2 月にレバレジーズ株式会社様主催の勉強会で、負荷試験に関する単独講演の機会をいただいたことでした。この講演自体、元上司である株式会社スピカの國府田 勲氏のお膳立てにより（酒を飲んだときのノリで）実施に至ったものでした。

人前での単独講演は初めてで講演そのものはガチガチでしたが、講演時のスライド資料を SlideShare で公開したところ、日本オラクル株式会社の奥野幹也氏に取り上げていただいたことで数多くのエンジニアの皆様にも好評をいただきました。そのスライドを見た技術評論社の池本公平氏より連絡を受けたのですが、そこでも奥野氏による推薦があったということでした。その後、池本氏から『Software Design』誌の特集で「負荷試験実践入門」の記事を連載（全 4 回）するチャンスをいただいたうえで、さらに今回の書籍執筆のお話をいただくこととなりました。

私自身のエンジニアとしてのスキルは、ほぼ携わってきた業務中心で培われてきたことで偏りが生じているため、執筆にあたっては当時同僚であった森下氏に共著を依頼しました。快く引き受けてもらえたことで、多くの部分が補完され、書籍として完成させることができました。

最初の舞台を準備していただいたレバレジーズ株式会社様。すべての局面で常に「やっちゃえよ！」と後押ししていただいた國府田 勲氏。ネット上ですぐに埋もれるかと思われたスライドに光を当てて推薦していただいた奥野幹也氏。誌面連載および書籍執筆の機会を与えていただいた池本公平氏。共著の森下氏。書籍執筆活動そのものをエンジニアに有益な業務とみなして全面的に協力してくれた株式会社ゆめみと、全体の査読・校正に多大なるご尽力をいただいたゆめみ関係者の皆様。そして勉強会、インターネット、雑誌、書籍といったすべての場でより良い技術・情報を求めノウハウをシェアするエンジニア文化と、それらに携わるすべての人々に深く感謝します。

仲川 樽八

本書を刊行するにあたり、お世話になった方々に感謝いたします。

特に執筆の機会を作っていただいた株式会社スピカの國府田 勲氏、技術評論社の池本公平氏に厚く御礼申し上げます。

また、共著者ではありますが、執筆のお誘いをいただいた仲川氏にも心から感謝いたします。この本書を執筆するに至るまでのさまざまな失敗や成功の経験を与えてくれた株式会社ゆめみと、査読して有益なコメントをいただいたゆめみ関係者の方々、これまで多くの支援をいただき本当にありがとうございます。

最後に、いつも私を支えてくれている妻や子どもたちと周囲の方々にあらためて感謝します。

森下 健