*クラウド時代にこそ再注目！「インフラ設計のセオリー」～後編～

株式会社JIEC
基盤エンジニアリング事業部
中村　圭吾

*アジェンダ

1．はじめに
2．本書の構成内容
3．なぜ非機能要求は難しいのか
4．非機能要求グレードの活用
5．インフラ設計のセオリー
6．おわりに

*1. はじめに

*1.はじめに

ITシステムに対する要求事項は、機能要求と非機能要求が存在します。

非機能要求は、業務機能と異なり、あいまいな要求をいかにまとめていくかが非常に重要なポイントとなります。

また、非機能要求の大部分は、システムインフラの設計・実装により実現することができます。

*1.はじめに

そこで我々は、インフラに求められる要求を項目化、分類化することで「見える化」を行い、過不足なく網羅できるようにしました。

そして、各項目をどのような観点で検討し整理すればよいかを「インフラ設計のセオリー」として出版させていただきました。

これから、その内容について簡単にお伝えさせて頂きます。
短い時間ですが、どうぞおつきあいよろしくお願い致します。

*2. 本書の構成内容

*2. 本書の構成内容

本書は、大きく分けて2部構成です。

＜設計前工程＞

何を設計するべきか？

序章　システムインフラについて
１章　インフラ構築の流れ
２章　インフラの要件定義と非機能要求
３章　要件定義から設計へ

＜設計工程＞

どのように実現するか？

４章　可用性設計のセオリー
５章　性能・拡張性設計のセオリー
６章　運用・保守性設計のセオリー
７章　セキュリティ設計のセオリー
付録　商用システムにおけるシステム構成の変遷

*3. なぜ非機能要求は難しいのか

*3.なぜ非機能要求は難しいのか

非機能要求は業務要求と異なり、ITへの専門性が高く、かつプロジェクトの初期段階ではユーザーが関心を持ちにくい項目です。

そのため、非機能要求は、ユーザーがベンダへ要件として提示しにくく、曖昧なまま次工程へ進んでしまうというケースが発生しやすいのです。

*3.なぜ非機能要求は難しいのか

ユーザ	ベンダ
具体的な要求提示が困難 （専門性が高い、業務でないため関心を持ちにくい）	**ベンダ主導での提案が難しい** （特定手段の必要性・理由が説明できない）

ギャップ

結果

・具体化が進んでいない上流工程で非機能要求を扱うことは困難
・ユーザ/ベンダで共通認識を持てない、合意できない

リスク誘発

システム開発	システム運用上のリスク拡大
プロジェクト運営成否へ影響 （基盤の変更、下流工程での問題発生）	想定外、あるいは極限状態での利用運用

*3.なぜ非機能要求は難しいのか

もう１つの難しい点として、要件が具体的な数値として提示されないことが多い、ということです。曖昧な表現の要求だと受け取り手の感覚によってズレが生じます。

たとえば耐障害性について・・・

ユーザ

可能な限り早く
復旧するようにして
ほしいな！

じゃあ１時間後に
復旧できれば十分だな

ベンダ

*3.なぜ非機能要求は難しいのか

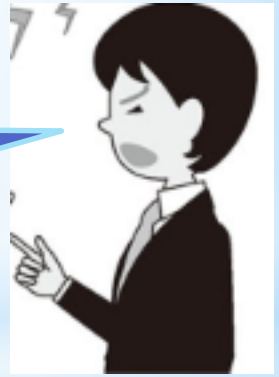
ところが・・・

ユーザ

なんで復旧に
1時間もかかるの？
遅すぎる！

初めから具体的に
言ってもらえないと
分かりませんよ！

ベンダ



このような事態は、後工程に入って発覚すればするほど、修正に必要な手戻りの工数も膨らんでしまいます。

4. 非機能要求グレードの活用

*4.非機能要求グレードの活用

前項で説明した非機能要求におけるユーザとベンダの行き違いや異なる理解を防止するため、本書では「非機能要求グレード」を活用することを推奨しています。

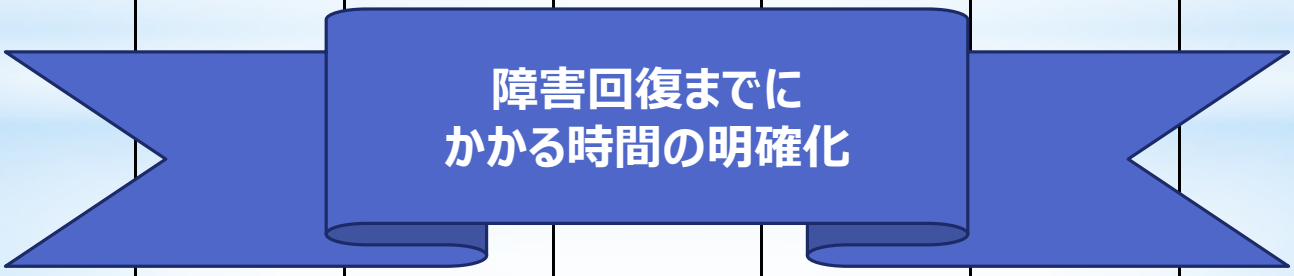
非機能要求グレードは、これまで見えにくかったシステム要求を具体的に「見える化」するための、ユーザとベンダ間の共通ツールです。

こちらは、IPA（情報処理推進機構）の専用サイトからどなたでもダウンロードすることができます。

*4.非機能要求グレードの活用

先程の障害回復を例に取ると、下記のような形で明確に数値化（見える化）されることになります。

メトリクス（指標）	レベル					
	0	1	2	3	4	5
サービス切替時間	24時間以上	24時間未満	2時間未満	60分未満	10分未満	60秒未満



*5. インフラ設計のセオリー

*5.インフラ設計のセオリー

要件が明確になった後の設計工程。本書ではこちらに重点をおいて説明しています。

特に設計の肝となる、

◆可用性
◆性能・拡張性
◆運用・保守性
◆セキュリティ

の4項目について設計における主要パターンと選択基準及び設計における考慮点などを、図を交えてわかりやすく説明しています。

*5.インフラ設計のセオリー

今回は１例として、可用性設計の１つであるシステムの冗長化についてご説明します。

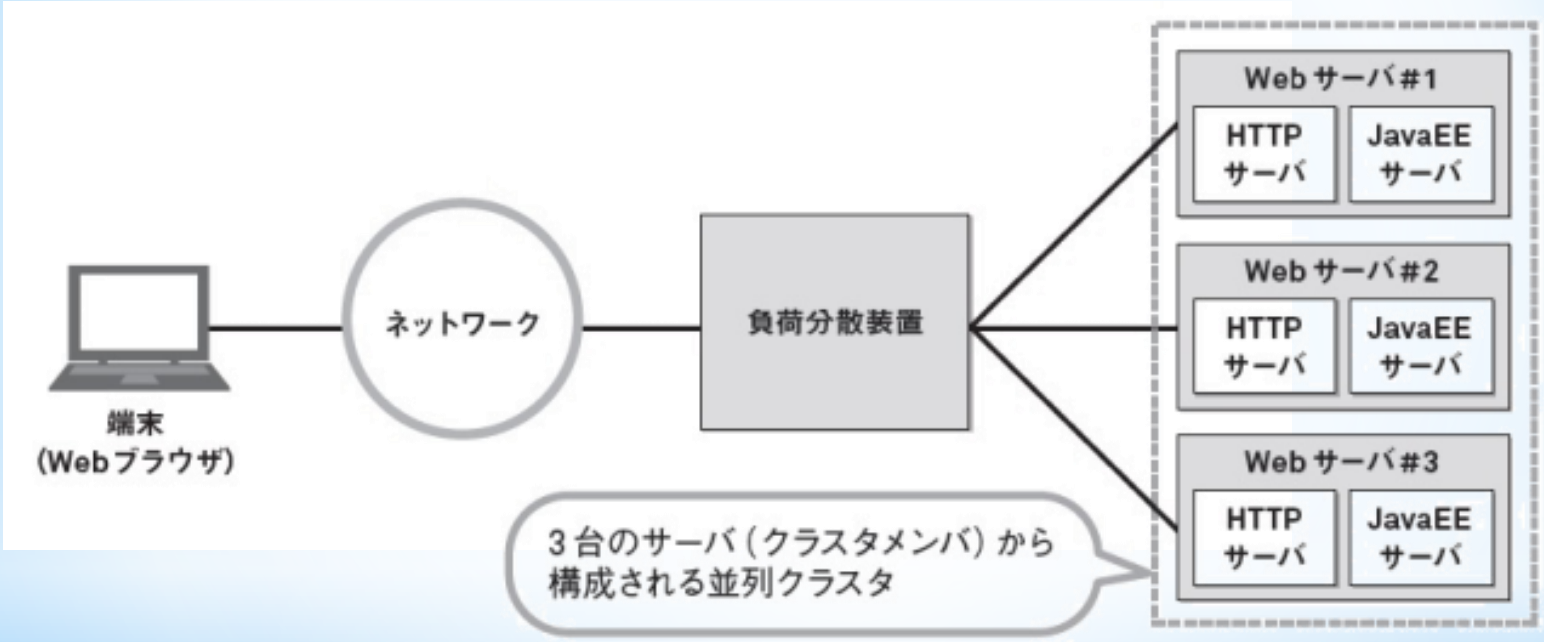
サーバの冗長化方式には、大きく分けて

①並列クラスタ方式
②HAクラスタ方式

の２通りが存在しますが、実装においてそのどちらを選択するべきか？という点について考えてみます。

*5.インフラ設計のセオリー

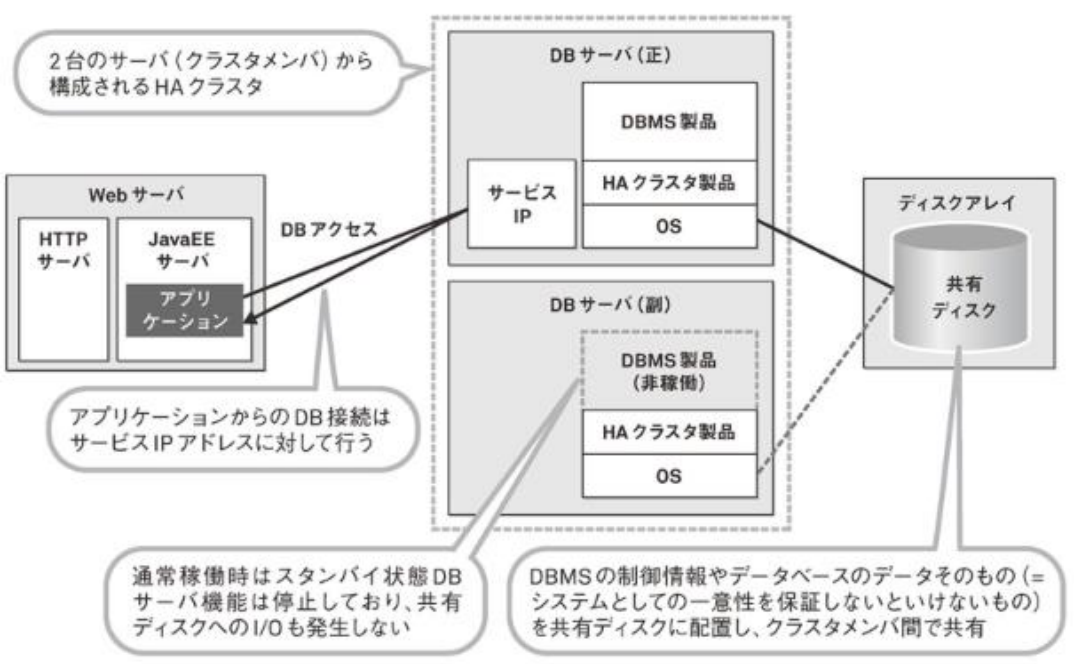
並列クラスタ方式の実装



並列クラスタ方式は、同じ機能と役割を持つ複数台のサーバを並列稼働させて、そのうちのどれかが障害になっても残りのサーバでシステムを継続できる構成です。

*5.インフラ設計のセオリー

HAクラスタ方式の実装



HAクラスタは、同じ機能・役割を持つ2台のサーバを稼働系と待機系という形で稼働させ、通常は稼働系のみで処理を行い、稼働系に障害が発生した場合に待機系が稼働系へと昇格することでシステム処理を継続する構成です。

*5.インフラ設計のセオリー

では、実際に構成を行う際にはどちらの構成を選択すべきでしょうか？
どちらの構成にも特徴があります。

カテゴリ	評価	並列クラスタ	評価	HAクラスタ
停止時間	○	同一機能のサーバが複数稼働しているためほとんどない	△	待機系を稼働系に昇格させるため一定時間の停止が発生する
性能	○	負荷分散されるため個別サーバの負荷が軽減される	△	1つのサーバで稼働するため負荷が集中する
一意または順序保証	△	並列処理前提なので、保証されない	○	単独サーバーなので保証される
データの共有	△	個別に情報を保持しているため共有できない	○	共有領域を保持することが可能

今回は一例としてシステムの冗長化を取り上げましたが、本書内では各項目ごとの設計のポイントや構成選定の考慮点などを、未経験者でもご理解頂けるよう、解説しています。

*5.インフラ設計のセオリー

では、実際に構成を行う際にはどちらの構成を選択すべきでしょうか？どちらの構成にも特徴があります。

カテゴリ	評価	並列クラスタ	評価	HAクラスタ
停止時間	○	同一機能のサーバが複数稼働しているためほとんどない	△	待機系を稼働系に昇格させるため一定時間の停止が発生する
性能	○	負荷分散されるため個別サーバの負荷が軽減される	△	1つのサーバで稼働するため負荷
一意または順序保証	△	並列処理前提なので されない		る
データの共有	△	個別に情報を保持して め共有できない		とが可能

先刻の障害対応の場合は、こちらを考慮の上構成を決定する必要があります

今回は一例としてシステムの冗長化を取り上げましたが、本書内では各項目ごとの設計のポイントや構成選定の考慮点などを、未経験者でもご理解頂けるよう、解説しています。

*6. おわりに

*6.おわりに

いかがでしたでしょうか。

オープン系システム構成実現イメージをほんの少しでもご理解頂くことができたでしょうか？
また、本書に対する興味を抱いて頂くことができましたでしょうか。

是非、手に取ってお読み頂き、その知識が読者の方々の手助けとなることができれば、著者としてこれほど幸せなことはありません。

ご清聴ありがとうございました