〈BIGLOBEのインフラストラクチャ〉

Linuxを活用したBIGLOBE基幹システム

BIGLOBE Mission Critical System Utilizing Linux

武下博英* Hirohide Takeshita
檜垣清志* Kiyoshi Higaki
深川泰介* Taisuke Fukagawa
長坂　茂* Shigeru Nagasaka
大縄陽一* Yoichi Ohnawa
菅原　淳* Jun Sugawara

要　旨

ITの急速な進歩によりサーバ&ネットワークが高性能で低コストなものに進化していくなかで，BIGLOBEの基幹システムも旧来の重厚長大で高価なサーバを利用した構成から脱却し，コストパフォーマンスに優れたサーバを組み合わせることでミッションクリティカルシステムを実現することが課題となっていました。

この課題を解決するために，NECは，独自の分散協調型アーキテクチャとトランザクション制御機構を新たに確立し，従来では高価なサーバでしか構築できなかったミッションクリティカル領域も最新のLinuxブレードサーバを多用できるようにすることで大幅なコストダウンを図るとともに，アプリケーションに依存しない一律なアーキテクチャによって品質，信頼性，保守性を向上させました。

With the rapid IT advances, servers and networking are becoming drastically more cost-effective. In such trends, it is necessary to break away, in development of BIGLOBE main system, from traditional large-and -monolithic structure with expensive servers to new combinational one with cost-effective servers.

To get this goal, NEC has established original architecture of distributed cooperative systems and transaction control mechanism. That resulted in serious cost reduction by expanding the applicable scope of Linux blade servers into mission critical systems, formerly constructed entirely of massive servers, and improved quality and reliability by applying unified architecture which is application system independent.

1．まえがき

BIGLOBEの基幹システムは当初，顧客情報管理システム，コンテンツなどの商品サービスを管理する商品管理システム，顧客の商品サービス契約状況を管理する商品契約管理システム，料金計算や請求処理を行う課金請求システム，および利用者のサービス利用可否を判断する認証システムといったプロバイダの各基本業務を，独立したシステムとして，それぞれに設計・開発・構築していました。また，これらのシステムは，従来のクライアント・サーバ型を基本としたそれぞれ固有のアーキテクチャで構築しており，各サーバもシステム特性に合わせた比較的処理性能の高い高価なミッドレンジやハイエンドのサーバを主に使用していました。

しかしながら，競争の激しいプロバイダ事業のなかにあっては，恒常的なコスト削減が必要であり，さらに，保守切れなどによるサーバリプレイスサイクルの短期化，基幹システムとして取り扱うアプリケーション範囲の拡大，サービス品質の向上などの課題もあります。

これらの問題を総合的に解決するために，システムごとの個別のアーキテクチャから脱却し，統一的で一貫性のあるアーキテクチャを開発しました。さらに，ハードウェアとして，BIGLOBE標準サーバであるLinuxブレードサーバを採用して，安価なミッションクリティカルシステム構築を進めています。

本稿では，このBIGLOBE基幹システムの新アーキテクチャについて紹介します。

2．BIGLOBE基幹システムのアーキテクチャ

2.1　MVC+アーキテクチャ

BIGLOBE基幹システムにおける最も基本的なアーキテクチャは，MVC＋（Model View Control+）です。これは，独自のフレームワークであるBizMAP（本誌掲載論文「BizMAPフレームワークによるBIGLOBE業務システムのプログラムレス開発」pp.94～98参照）に基づき，システム機能を6種類のコンポーネントタイプ（V層，P層，C層，M層，D層，T層）に分類し，それぞれのコンポーネント

＊ BIGLOBE構築運営本部
BIGLOBE Design and Operations Division

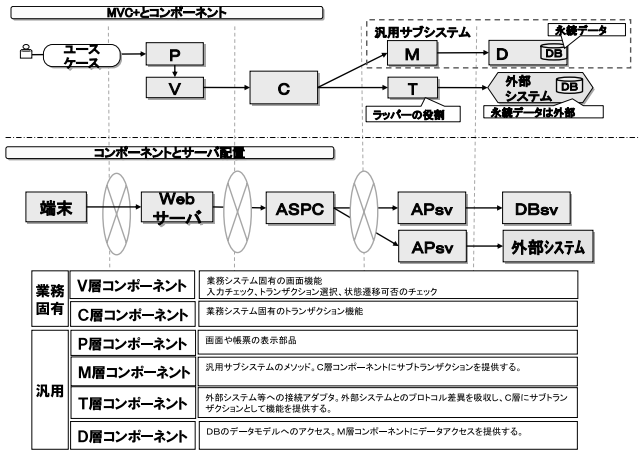


図1　MVC＋とサーバ実装
Fig.1　MVC+ and server deployment.

を，どのサーバに配置するかを規定するものです（**図1**）。

BIGLOBEでは，どのようなアプリケーションを構築する際にも，このアーキテクチャを適用することにより，DBsv以外はすべて安価なサーバを水平分散する実装方式を採れるようになりました。

2.2　トランザクション制御機構（ASPC）

アーキテクチャの中心となるのは，分散協調型トランザクション処理制御機構であるASPC（Application Service Processing control）です。ASPCは，すべてのトランザクション処理を各サブシステムや外部システムと連携することで実現するトランザクション制御HUBとなっており，Hub＆Spoke型アーキテクチャを構成しているといえます（**図2**）。

このASPCは，100台以上の1CPUサーバを水平分散型で構成しており，順次Linuxブレードサーバへ切り換えています。

またASPCでのトランザクション処理は，サーバ認証および利用者認証が必須となっており，トランザクション処理実行において高いセキュリティを実現しています。

2.3　トランザクション種類とサブシステムタイプ

次に重要な概念であるサブシステムタイプについて説明します。ASPCから処理を依頼されるサブシステムは，顧客管理，契約管理から，認証やDWH（Data Ware House）といった様々なものがあります。

BIGLOBEでは，更新トラヒックよりも認証や参照トラヒックの方が格段に多いという特長がありますので，この特長から，サブシステムを，顧客DBのようにマスタとして1つしか持つことができない「マスタ系」，認証DB，参照専用DB，DWHのようにマスタ系からのコピーやデータ加工して作られる「レプリカ系」，短期間に何度もアクセスされるデータをキャッシュしておく「キャッシュ系」の3タイプに分類しています。

このように，トランザクションの種類でサブシステムタイプを分離することによって，互いに重要なトランザクション処理が，負荷影響を受けないよう工夫しています（**図3**）。

加えてこのことによって，レプリカ系は大量に同一サーバを並べる方式（水平分散）が利用できるため高拡張性を実現でき，キャッシュ系はセッション情報などをキャッシュしておくことで，サーバ当たりの負荷低減を実現し，高レスポンスを実現できるようになっています。

2.4　高可用性の実現

BIGLOBE基幹システムは，サービス無停止を実現するための冗長化方式として，「水平分散型」と「ACT/STB型」の2種類のみでシンプルにシステム全体を構築しています。

水平分散型とは，どのサーバが故障しても，故障サーバのみ自動閉塞し，残りのサーバで同じ処理を無停止で継続できるようにしてある構成のことです。この水平分散型構成は，データ蓄積をしないWebサーバ（V層），ASPC（C層），APサーバ（T層，M層）での採用に加え，DBサーバについても，サブシステムタイプがレプリカ系とキャッシュ系の場合に，採用しています。この場合，マスタ系のトランザクション更新時の性能確保のために，全レプリカ系およびキャッシュ系は，非同期に更新をするようにしています（**図3**）。

また水平分散型では市販の負荷分散装置を使用して常に適切な負荷バランスをとるとともに，故障サーバの自動切り離しを実現しています。

ACT/STB型構成は，マスタDBにのみ適用していますが，今後OracleのRAC機能などを適用検討中であり，DBサーバも水平分散型にし，さらなる可用性向上を図ってい

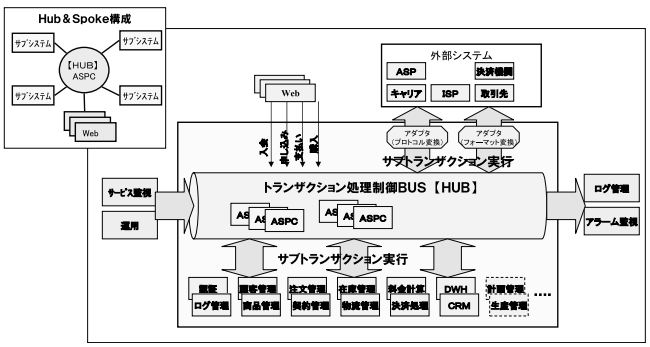


図2　トランザクション制御BUSとHub＆Spoke構成
Fig.2　Transaction control BUS and Hub&Spoke configuration.

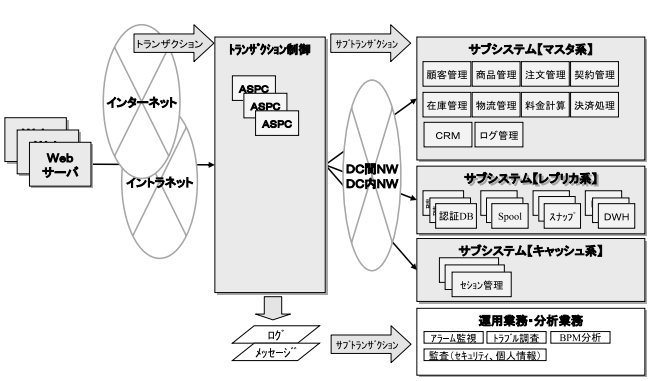


図3　トラヒックの特長とサブシステム分類
Fig.3　Characteristic of transaction and subsystem category.

く予定です。

2.5 高拡張性の実現

上記のとおりBIGLOBEの基幹システムは，そのほとんどが水平分散型で構成できるため，トラヒックの増加には，単純にサーバ増設で対応が可能となります。

一方マスタ系サブシステムのDBサーバは，ACT/STB型であるため，従来は，トラヒック増加に対して，高性能なサーバへのスケールアップしかできませんでした。これを解決するため，MVC＋アーキテクチャに基づいた機能ごとのサブシステム分割により，サブシステム単位でDBクラスタを組めるようにしました。これはASPCが，IPネットワーク（DC間NW，DC内NW）経由で，サブシステムのサービスを順次呼び出すことでトランザクション処理を実現しているために可能なわけです（**図4**）。

このようにシステム全体を，適切な単位でサブシステムに分割し，サブシステムごとにDBクラスタを構築することでDB負荷増に対応できるため，比較的安価なミッドレンジサーバでのシステム構築が可能となります。

2.6 リソース管理

MVC＋アーキテクチャにより，コンポーネントタイプ（V，C，M，T）ごとにサーバクラスタが分かれており，さらにAPサーバについては，オンライン，バッチ処理，外部コネクタといったアプリケーション種類によって，サーバクラスタをグルーピングしているため，サーバごとの負荷変動は一律であり，負荷増加に対しても，シンプルで容易なリソース管理を実現しています（図4）。

2.7 統合的なログ管理

ASPCでは，必ず認証付きでトランザクションを実行するようにしているので，トランザクションごとのロギングをこのポイントで集中的に取得することができます。このログには，誰が，いつ，どのトランザクションを実行し，どのデータを修正したかまで記録していますので，各種問題調査，監査ログ，オペレータの生産性などを把握することができます。

BIGLOBE基幹システムでは，このようにASPCでロギングを集中的に行っているため，システム開発ごとにログ採取の方式や方法を開発する必要がないのに加え，複数のアプリケーションシステムのログを一元管理できるというメリットがあります（図3）。

2.8 サービス監視と高度自動切り離し

負荷分散装置で自動的に検出できないサーバ障害（たとえば，アプリケーションレイヤでの無応答）については，サーバ1台ごとのサービス応答をサービス監視サーバから監視することで，障害サーバの切り離しを実現しています。サーバのレスポンス異常を外部のサービス監視サーバから定期的に監視し，ルールベースで検証の上，動作異常と判断したものを，負荷分散装置に通知し切り離します。これによって，単純判断できないような障害のケースも自動的に切り離すことができるようしていく予定です（**図5**）。

2.9 統一バックアップ

BIGLOBE基幹システムでは，サブシステムごとではなく，共通バックアップサーバで一元的にDBなどのデータバックアップを行っています。

バックアップの方法は，ディスク装置内部に3rd Mirrorを作り，3rd Mirrorを共通バックアップサーバでマウントしてバックアップを取得するという方法を採っています。

3rd Mirrorからバックアップすることで，オンライン処理に影響を与えることなく，バックアップを行っています。

3．DRと認証サーバの地域分散

3.1 DR（ディザスタリカバリ）

基幹システム構築において，災害時に処理の継続を実現するDRは最も重要な要件の1つです。BIGLOBEの基幹システムでは，V層，C層，M層という単位でサブシステムを分離しており，さらにASPCは，ほかのASPCと連携してトランザクション処理を実行できるためHub＆Net型のアーキテクチャを実現することができます（**図6**）。

このアーキテクチャにより，サブシステムを複数のデータセンタに分散配置しDRを実現することが可能です。

以下では優先的に取り組んでいる認証サブシステムの地

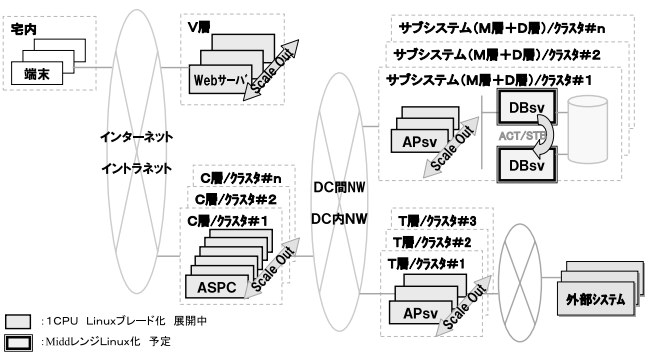
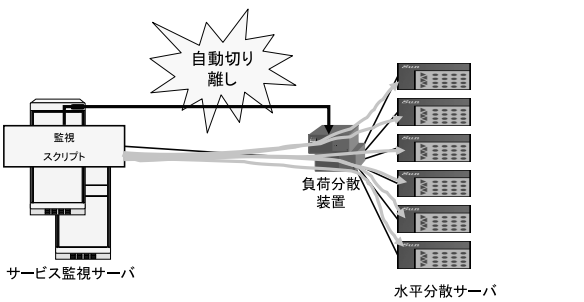


図4 MVCアーキテクチャと冗長化構成・スケーラビリティ
Fig4. MVC architecture and redundant configuration.



水平分散サーバ1台ずつのサービス応答を監視し、障害を検知した場合に、障害サーバを自動切り離し。

図5 サービス応答監視によるサーバ自動切り離し
Fig5. Separation of fault server by monitoring service response.

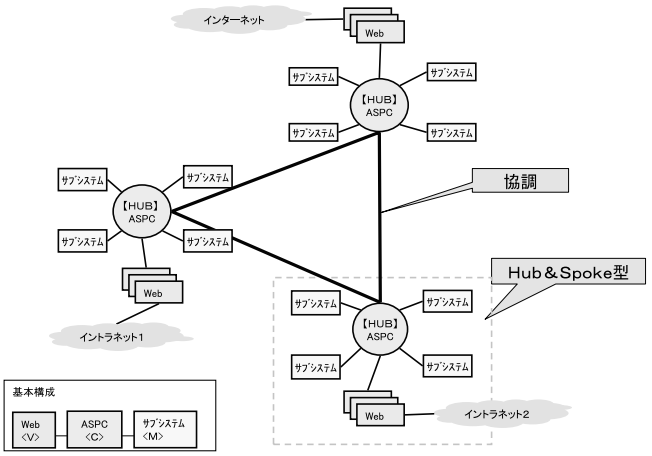


図6　ASPCによるHub & Net型の実現
Fig.6　Realization of Hub&Net configuration with ASPC.

域分散について説明します。

3.2　BIGLOBE認証システムの現状

BIGLOBEの認証システムは用途に応じて大きく2つに分類されます。1つは，ユーザがインターネットに接続する際に行う「接続認証」です。もう1つは，BIGLOBEの提供するサービスコンテンツをユーザが利用する際に行う「サービス認証」です。これらの認証システムは分散協調型アーキテクチャのもと，レプリカ系サブシステムの位置付けとなり，水平分散型システムにより高可用性が実現されています。

また「接続認証」については地域分散をすでに展開中です。

3.3　認証システムの地域分散

BIGLOBEではMVC＋アーキテクチャの特性を生かし，今後，マスタ系サブシステムからのレプリケーションによる地域分散，および認証システムの地域分散を行うことで，認証システムのDRを実現していきます（**図7**）。

これによって，データセンタ機能が停止に陥るような地域災害が発生した場合においても，インターネットへの「接続認証」，サービスコンテンツの「サービス認証」の維持が可能となり，高信頼サービスの提供が可能となります。

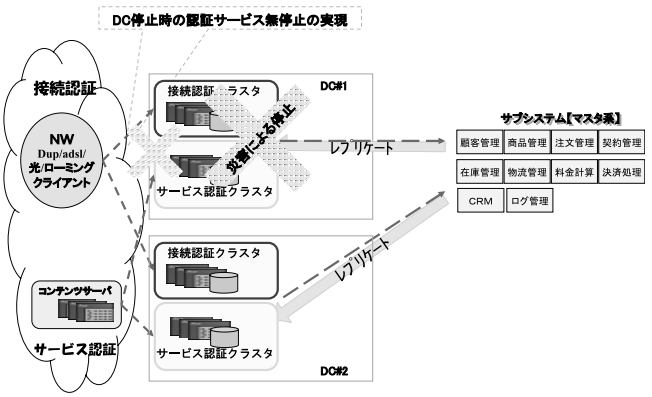


図7　認証システム地域分散の概要
Fig7. Outline of geographically-distributed authentication system.

4．まとめ

本稿では，MVC＋アーキテクチャに基づいた，Linuxブレードサーバを多用したBIGLOBE基幹システムのアーキテクチャを紹介しました。

今後さらに，RAC（Real Application Cluster）などを活用した新しい技術の確立により，DBサーバの水平分散化，認証システムを始めとした基幹システムの地域分散化を進め，高可用性を実現するとともに，BIGLOBE全体として，さらなる費用削減，信頼性向上を図っていきます。

筆者紹介

Hirohide Takeshita
武下　博英（たけした　ひろひで）　1995年，NEC入社。現在，BIGLOBE事業本部BIGLOBE構築運営本部主任。

Kiyoshi Higaki
檜垣　清志（ひがき　きよし）　1987年，NEC入社。現在，BIGLOBE事業本部BIGLOBE構築運営本部グループマネージャー。

Taisuke Fukagawa
深川　泰介（ふかがわ　たいすけ）　1983年，NECソフトウェア北陸入社。現在，NEC BIGLOBE事業本部BIGLOBE構築運営本部勤務。

Shigeru Nagasaka
長坂　茂（ながさか　しげる）　2000年，NEC入社。現在，BIGLOBE事業本部BIGLOBE構築運営本部主任。

Yoichi Ohnawa
大縄　陽一（おおなわ　よういち）　1990年，NEC入社。現在，BIGLOBE事業本部BIGLOBE構築運営本部マネージャー。

Jun Sugawara
菅原　淳（すがわら　じゅん）　2000年，NEC入社。現在，BIGLOBE事業本部BIGLOBE構築運営本部勤務。情報処理学会会員。