

オーガニックコンピューティング

Organic Computing

あらまし

近年、汎用コンピュータ部品の高性能化，低価格化が進んでいる。しかし，それらをそのまま組み合わせただけでは，高性能，高信頼，高可用なITシステムは構築できない。また，システムの大規模化，複雑化のために，人手に負えないほど運用管理が困難になってしまう。オーガニックコンピューティングとは，それらの汎用部品をベースとしつつ，高性能，高信頼，高可用なITシステムを構築，管理，運用する技術である。構成を工夫することによって，専用部品の使用を最小限に抑え，ソフトウェアとしてはシステム自身が最高性能を出すように制御する自律運用機能や，運用しながらハードウェアの取替えを可能にする新陳代謝機能を実現している。

本稿では，オーガニックコンピューティングの概要，試作したシステム，機能について説明する。

Abstract

Over the past few years, commodity computer components have become progressively faster and cheaper. Unfortunately, we cannot build large IT systems that have high performance, high reliability and high usability by simply combining these components as is. Moreover, even if this could be achieved, there would be many other difficulties and high management costs because such systems would be too big and complicated to be maintained by a human administrator. Organic computing is a technology to construct, manage, and operate such systems using commodity components. The use of special components is minimized by carefully combining them, and a self-optimization function and means to ease replacement of components are provided at the software level. This paper explains the concept of organic computing and describes several organic-computing functions.



小沢年弘
(おざわ としひろ)

ITコア研究所 所属
現在，オーガニックコンピューティングおよびコンパイラの研究に従事。



安里 彰
(あさと あきら)

ITコア研究所 所属
現在，オーガニックコンピューティングの研究に従事。



鈴木和宏
(すずき かずひろ)

ITコア研究所ITアーキテクトチャ研究部 所属
現在，オーガニックコンピューティングの研究に従事。



勝野 昭
(かつの あきら)

ITコア研究所ITアーキテクトチャ研究部 所属
現在，オーガニックコンピューティングおよび基幹サーバの研究に従事。

まえがき

ITシステムが社会の重要なインフラストラクチャになっている現在、ITシステムには高い処理能力と24時間365日ノンストップで動作し続ける信頼性を安価に提供することが要求されている。富士通では、このような要求に応えるため、IT基盤「TRIOLE」の開発・製品展開を進めている。著者らはこのTRIOLEのコア技術を開発すべく、2002年1月、国家プロジェクトの一環^(注)として、オーガニックコンピューティングの研究開発⁽¹⁾を開始した。

著者らが取り組んでいるオーガニックコンピューティングとは、安価なコンポーネントを組み合わせたつ、その構成や利用方法を工夫することにより、高性能、高信頼、高可用なITシステムを構築し、管理、運用するための技術である。

本稿では、オーガニックコンピューティングの概要、試作したシステムについて説明する。

オーガニックコンピューティングの概要

オーガニックコンピューティングでは、サーバとして、多数の安価な汎用プロセッサ（ノード）を高密度に集積したブレードサーバを想定している。汎用プロセッサの世代交代の速さから、その時点での最速の汎用プロセッサを適宜使用することが、価格的にも性能的にも有利だからである。しかし、ある程度の性能を得るためには、多数のノードを必要とすることから、様々な問題が出てくる。例えば、ノード故障による信頼性の低下や、汎用ネットワーク部品ではノード間の通信性能を得られない点である。また、運用管理面では、多数のノードを抱えることによる複雑さ、コストの上昇が挙げられる。

これらの問題に対して、ノードの信頼性を高めるためにノードを機能別に分離し、高性能化のために高速インタコネクタを備えた次世代ブレードサーバ「XION」{ X (ten) gigabit Interconnected Organic Nodes }の開発を進めている。ブレードサーバにおいては、インタコネクタが全体の性能を左右する重要な部分であるので、ここには富士通が開発した高性能スイッチを使用している。さらに、運用管理が

(注) NEDO（新エネルギー・産業技術総合開発機構）からの委託研究「高信頼・低消費電力サーバの研究開発」に基づく。

容易で、高信頼、高可用なシステムにするために、サービスを止めずに保守管理を行うためのインスタントノードマイグレーション技術と、サービスの自律運用機能を実現する「Phantom」システムを開発中である。インスタントノードマイグレーションは、サービスを実行させながら、稼働するノードを入れ替えることにより、古くなったノードをシステムから切り離し、新しいノードに換えていくといった新陳代謝機能の基礎技術である。Phantomは、システムの状態を監視し、最高の性能が出るように資源割当てなどを自律的に行うミドルウェアであり、運用管理の効率化、低コスト化を実現する。

次世代ブレードサーバ「XION」

本章ではオーガニックコンピューティングの研究開発用プラットフォームとして試作しているブレードサーバ「XION」の概要を述べる。

全体構成

XIONのシステム構成を図-1に、外観を図-2に示す。XIONは19インチラックに搭載可能な高さ4Uのシャーシと各種ブレードから構成される。16枚の機能別ブレードを実装することができ、それらはスイッチブレードを介して10ギガビットイーサネット（以下、10GbE）で接続される。それとは別に監視制御のための自律神経の働きをするネットワークがあり、管理ブレードに集線される。管理ブレードは、自律神経ネットワークを介してシステム内の様々なセンサ情報を収集し、システム全体を最適な状態に保つための諸々の制御を行う。

機能別ブレード

機能別ブレードとは、CPU機能、ディスク機能、

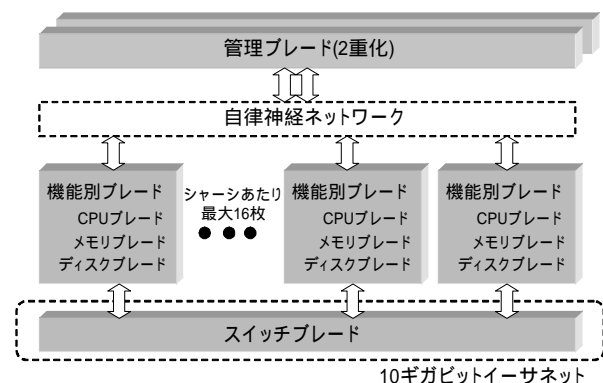


図-1 XIONのシステム構成
Fig.1-System structure of XION.



図-2 XIONの外観
Fig.2-XION system.

キャッシュ機能といった単機能に特化したブレードである。CPUブレードはハードディスクを搭載せず、CPUと最小限のメモリのみで構成される。一方、ディスクブレードには大容量のディスクの搭載が可能であり、メモリブレードには、10 Gバイトを超える大容量メモリを載せることができる。

これらの機能別ブレードを組み合わせることにより、アプリケーションの性質に応じた適切な計算能力、メモリサイズ、ストレージ容量を持つシステムを柔軟に構成できる。また、ディスクブレードのみを2重化して、効率良くシステムの信頼性向上を図ることも可能になる。

自律神経ネットワーク

XIONには、10GbEインタコネクタとは別に自律神経ネットワークと呼ぶIPMB (Intelligent Platform Management Bus) ⁽²⁾ インタフェースのネットワークがある。各機能別ブレードには多数のセンサが搭載され、チップの温度などを常に監視している。これらの情報は各ブレード上に搭載したBMC (Board Management Controller) によって収集され、自律神経ネットワークを通して管理ブレードに集約される。管理ブレードは、集められたデータに基づきシステム全体を最適な状態に保つために、例えばノードマイグレーションの起動、ファン回転数の制御などの指示を自律神経ネットワークを通じて各ブレードに伝える。

インタコネクタ

XIONの高性能化のため、各ブレード間を10GbEで接続している。シャーシあたり最大16枚の機能別ブレードが、バックプレーンおよびスイッチブ

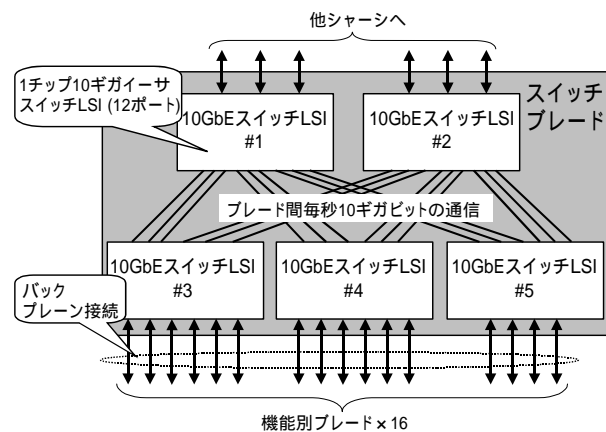


図-3 スイッチブレードの構成
Fig.3-Configuration of switch blade.

レードを経由して毎秒10ギガビットのスループットで通信可能となっている。スイッチブレード上には、富士通が開発した高性能1チップ10GbEスイッチLSI⁽³⁾が搭載されている。このチップは10GbEインタフェースを12ポート持つレイヤ2スイッチチップで、240 Gbpsのスループットと450 nsの低レイテンシを実現している。

図-3に示すように、スイッチブレードは10GbEスイッチLSIを5個搭載し、それらをツリー接続することにより、ブレード間毎秒10ギガビットのノンブロッキング通信を実現している。最大16枚の機能別ブレードと接続するためのダウンリンクポートに加え、シャーシ間接続用のアップリンクとして6ポートが設けられている。

インスタントノードマイグレーション

「ノードマイグレーション」とは、あるノード上で動作しているアプリケーションを、それが動作している環境(OS)ごと、ほかのノードに移し替える機能のことである。「インスタント」というのは、それを瞬時に行うことを意味している。つまり、アプリケーションの状態にかかわらず、ノードマイグレーションが行えることであり、たとえアプリケーションが通信中であっても、通信を途切れさせないで移動可能である。この機能により、ハードウェア保守のためにノードの電源を落とすときにも、そのノード上のアプリケーションを別のノードに移動させて実行を続けたまま保守を行うといった柔軟な保守管理が可能となり、システムの可用性を上げることができる。

以下では、インスタントノードマイグレーションの実現に関して、必要となるシステム構成要件と実現方法について、その概略を述べる。

システム構成要件

ノード上のアプリケーションをほかのノードに移動させた後、そのノードの電源を落とすには、関連するすべての情報をノード中に残しておけない。例えば、ファイルがノードの中のディスクに置かれている場合、アプリケーションがほかのノードで実行されるようになって、ファイルの内容を提供するために、元のノードの電源を落とすことができない。著者らは、各ノードはディスクレス構成であることを前提とし、ネットワークI/O以外のI/Oもネットワークを経由して行われる構成を想定している。

ノード移動方式

ノード移動の基本的な機構として、OSのsuspend/resume機構を応用している。この機構は、LinuxやWindowsなどのOSに既に組み込まれている機能であり、suspend時に全プロセスを停止させた後、メモリイメージのスナップショットをディスクに書き込み、resume時にはディスクから読み込むことで、同一ノードにおいて処理の中断を実現する機能である。著者らは、この機構をベースに、あるノードのsuspend時のスナップショットをほかのノードに通信して送り、そちらでresume処理をすることによりノードを移動させる機能を開発した。

移動先となるノードでは、スナップショットを受け取るための準備として、マイグレーションの開始に先立って、通信路確立のために接続待ちで待機する。移動元ノードでは、スナップショット作成後、通信路を確立し、スナップショットを送信する。

ノード移動時間の短縮

ノード移動にかかる時間の短縮のために、移動先ノードで実行を再開するまでに通信するデータ量の削減を図っている。つまり、移動元ノードでは、移動先ノードでとりあえず実行を再開するのに必要なデータだけでスナップショットを作成する。移動先ノードでは、スナップショットを受け取った時点で実行を再開し、スナップショットに含まれないデータが必要になったときには、移動元から別途データを送信してもらう。このような方式により、移動による中断時間の短縮を実現している。これらの機能をLinux OSのsuspend/resume機構を変更すること

で実装し、実験を行った結果、移動による処理中断時間を数秒程度に抑えることができ、移動によっても通信が途切れないことを確認した。

サービスの自律運用機能 “ Phantom ”

稼働中のアプリケーションやシステム全体をダウンすることなく運用を継続するためには、CPUノードを含めたリソースを仮想化する技術が必須となる。リソースの仮想化によってクラスタシステムの信頼性や可用性が向上することが期待されている。

そこで著者らはオーガニックコンピューティングに向けたクラスタシステムの効率的な運用を実現するために、サービスの自律運用機能 “ Phantom ” (4)を開発している。Phantomはクラスタ内のノードを仮想化することによって負荷変動などの外的刺激に対して自律的に適応するためのミドルウェアである。Phantomによって実際の計算機ノードとアプリケーションが操作するノードを切り離すことができ、アプリケーションからノードの構成の変化や故障などを隠ぺいすることができる。さらに各アプリケーションに割り当てるノード数を動的に制御することによって、柔軟なリソース割当てを実現できる。

ノード仮想化方式

Phantomによるノード仮想化方式の概要を図-4に示す。ユーザアプリケーションからはPhantomが提供する仮想的なノード（仮想ノード）だけを操作でき、物理的な計算機ノード（実ノード）を直接操作することができなくなっている。仮想ノードは実ノードとは異なるIPアドレスを持ったノードであり、適当な実ノードにマッピングされる。システム外部もしくはアプリケーション間での通信は仮想ノードのIPアドレスを用いて行われるが、実際に

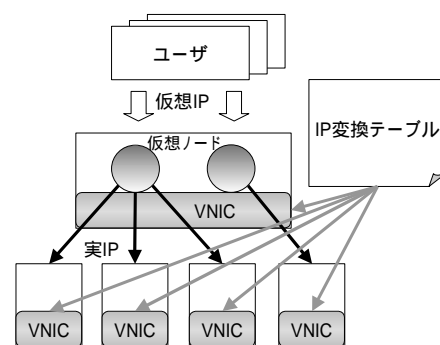


図-4 Phantomのノード仮想化方式の概要
Fig.4-Overview of node virtualization of Phantom.

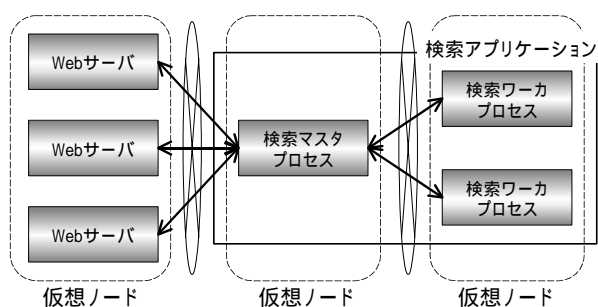


図-5 検索アプリケーションシステムの構成
Fig.5-Structure of search application system.

は実ノードのIPアドレスに変換されてやり取りされる。この変換は各ノードに実装されているVNIC (Virtual Network Interface Card) と呼ばれる仮想的なネットワークインタフェースが行う。VNICはカーネルモジュールとして実装され、仮想ノード宛のパケットを監視して、IP変換テーブルを参照し宛先を実ノードのIPアドレスに書き換える働きをする。

これによってアプリケーションにはあたかも仮想ノード上で動作しているように見せることができ、ユーザは実ノードの構成や台数にかかわらずアプリケーションを動作させることができるようになる。

仮想ノードに実ノードをマッピングする場合、一つの仮想ノードに対して複数の実ノードをマッピングすることができる。つまり仮想ノードへのリクエストを複数の実ノードに分散することによってロードバランシング型のクラスタシステムを構成できる。

機能検証

図-5に示すような検索アプリケーションシステムを構築して、Phantomの自律運用機能を検証した。図中の検索アプリケーションは、DB検索を担当するワーカプロセスと、複数ワーカに検索リクエストを分散し、結果をWebサーバに返すマスタープロセスから構成される。本実験ではPhantomの自律運用機能により、Webサーバと検索ワーカ数を動的に変更した。毎秒約400リクエストの負荷をかけたときの様子を図-6に示す。初期状態としてWebサーバ、検索ワーカとも1台ずつとしたが、CPU負荷、応答時間が一定以下になるようにPhantomが自律的に制御してノード数を増やした結果、最終的にすべてのリクエストを処理できていることが分かる。また反対に負荷を止めると、これを検出して初期状態に

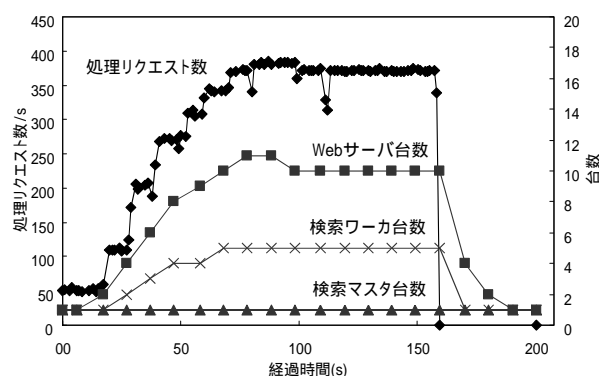


図-6 ノード数と処理リクエスト数の時間変化
Fig.6-Time change of number of nodes and processing requests.

戻っていることが確認された。

む す び

本稿では、安価なコンポーネントを組み合わせつつ、その構成や利用方法を工夫することにより、高性能、高信頼、高可用なITシステムを構築し、管理、運用するための技術であるオーガニックコンピューティングについて、その開発プラットフォームである、次世代ブレードサーバ“XION”と、システムを新陳代謝させるためのインスタントノードマイグレーション技術、サービスの自律運用機能“Phantom”を説明した。今後、これらを統合してシステム全体としての完成度を高めていきたい。

参考文献

- (1) 西川克彦ほか：オーガニックサーバ．*FUJITSU* , Vol.54 , No.4 , p.298-304 (2003) .
- (2) Intelligent Platform Management Bus Communications Protocol Specification v1.0, 1998, Intel Corporation, Hewlett-Packard Company, NEC Corporation, and Dell Computer Corporation . <http://developer.intel.com/design/servers/ipmi>
- (3) T. Shimizu et al . : A Single Chip Shared Memory Switch with Twelve 10Gb Ethernet Ports. *Hot Chips 15 - August 19, 2003* .
- (4) 鈴木和宏ほか：クラスタ計算機システムにおけるノード仮想化方式．並列/分散/協調処理に関するサマー・ワークショップ (SWoPP2003松江), 電子情報通信学会・情報処理学会共催, 松江, 2003年8月, p.67-72 .