

Element Computing 概念に基づく新しい分散処理システム構成法の提案

中村和敬[†] 日比野靖[†]

1. 概要

近年、ユーザあたり計算機資源数の飛躍的な増大により、管理が困難になり様々な非効率が発生している。また多様な装置にコンピュータが組み込まれ、もはや単なる計算機資源の枠に収まりきれない、処理要素ともいべきデバイスが出現してきており、しかもその種類は増大している。今、それらを統合的に利用する為の新たな分散システム環境への要求が高まっている。しかし既存の OS の提供するシステム機能はこのような状況に対して柔軟性を欠いている。

我々はこれからの分散システムが備えるべき要件を導く為、実装レベル、処理要素間接続、利用形態の三つの観点から考察した。それによって導き出された要件を満たす分散システム形態を検討し、Element Computing として提案する。さらに、Element Computing の実現性と利用可能性を検証するための方法を示す。

2. システム要件の考察

実装レベル、処理要素間接続、利用形態の三つの観点から現状を考察し、これからの分散システムが備えるべき要件を導く。全ての要件を満たす関連研究及び製品は存在しないが、個々の要件を満たすものを挙げていく。

2.1 実装レベルとユーザインターフェース

既存の分散コンピューティング環境は OS より上のレベルで実装されている。この事はユーザに特定の OS、乃至 Application の利用を迫る。システムの移行の際には Software 資産 / ユーザ知識の継承の困難さをもたらす。

OS に依存しないという点に於いては、Java VM や、inferno の VM がこの要件を満たす。しかし OS の果たしている役割は大きく、このような部分的解決法のみでは不十分であると考える。

以上から以下の要件を定めた。

要件 1: Software レベルでなく Hardware Component レベルで実装を行うことにより、OS に依存しない分散コンピューティング環境を提供する事。

2.2 処理要素間接続

処理要素間の通信路には、レジスタ経由、メモリ経由、PCI バス、USB、インターネット、電話網、など様々な形態がある。その目的は、いずれも処理要素間の通信を可能とし、利用効率を高めたり、処理要素同士の連携によって新しい機能を作り出す事にある。しかし、既存のシステムでは複数の通信路を跨いだ通信が全く考慮されていない為、複数の通信路に跨る経路の設定は不可能な場合が多い。この為処理要素間接続の本体の目的が達成できない。

この問題を解決する為、一つのネットワークに全ての処理要素を接続しようという試みもある。しかしこれはセキュリティに悪影響をもたらす。通信路は秘匿出来るべきである。

これらの要件に関しては、RPC や、micro kernel を採用している OS は、同様の方向性を持っていると考えられるが、筆者の知る限り、上記のような一般的な形での接続を必須とすることを打ち出したものは存在しない。

以上から以下の要件を定めた。

要件 2: 様々な通信路を越えて処理要素同士が通信可能な、抽象的なネットワーク (Abstract Network) を提供する事。またこのネットワークは、アクセス制御機能を備え、さらに通信路の存在を秘匿可能なセキュア機能を提供しなければならない。

2.3 利用形態

現状は、ユーザは複数の管理者から処理要素を提供され、それらを組み合わせ利用し、さらに他のユーザにサービスの提供を行うという形態になっている。その為、誰もが“ユーザ”かつ“管理者”であるという状況になっている。

しかし、既存システムは計算機資源が高価であった頃の一つのコンピュータを複数のユーザが共同利用するという利用形態に適応した構成になっており、これはいわゆる“分散システム”も同様である。

[†] 北陸先端科学技術大学院大学 情報科学研究科

このようにシステムの想定する利用形態と現状との間に乖離があるため、様々な歪みや非効率が発生している。

既存の OS では、Plan9 や、inferno がこの要件を満たす。また、Java VM もこの利用形態を想定している。

以上から以下の要件を定めた。

要件 3: 集中管理を排除し、ユーザ自ら処理要素の収集/管理、利用/提供を行える事。

3. Element Computing

前章でまとめた三つの要件を備えたシステムを、我々は Element Computing と名付けた。システムで利用可能な処理要素を Element と呼ぶ。p

要件 1 から、Element の最小単位を Hardware Component である。Element は要件 2 の Abstract Network に接続される。また Element はアクセス制御機構や、セキュア機能に対応している必要がある。要件 3 から、Element を組み合わせて新しい機能を作ることが出来、同時にこの機能も Element して扱うことができる。

4. プロトタイピングの構想

以下の通りにプロトタイピングを行い、Element Computing の実現可能性を探る。

4.1 システム構成

TCP/IP で接続された、複数の IBM-PC/AT 互換機を用いる。この上で、Hardware Component である、CPU、Memory、HDD、Graphic board、Display、Mouse、Keyboard を Element 化する。また、これらを接続する通信路としての Abstract Network およびその入り口となる、Abstract Bus の実装を行う。Abstract Bus/Abstract Network の機構を用いて、実装した Hardware Component Element から仮想的な IBM-PC/AT 互換機 Element を構成する。

4.2 実装方法

(1) 利用する PC は 2 台とし、Ethernet および TCP/IP を用いて相互接続する。

(2) ふたつの PC の CPU の上で VM を動作させ、Abstract Bus を実装する。VM 上には通常の OS を動作させる。VM 上には通常の OS を動作させる。OS が発行する入出力デバイスに対するコマンドは、ローカルの存在するデバイスに対するものはそのままデバイスに伝えられる。ローカルに存在しないデバイスに対するものは、VM が捕らえ、Abstract Bus を実装しているモジュールに処理を任せる。

(3) Abstract Bus を実装しているモジュールでは入出力コマンドをネットワークに乗せてリモート PC に伝える。(ローカル側仮想ドライバ) リモート PC では、ネットワーク経由で伝えられた入出力コマンドを入出力デバイスに伝える。(リモート側仮想ドライバ)

4.3 利用可能性の検証

主に応答遅延が主要な問題となると思われるので、これに注目して、“素の”PC と、Element Computing による PC Element の比較を行う。Element の内、ローカル接続からリモート接続へ分離することを優先するのは、ストレージである HDD、ユーザインタフェースである、Mouse、Key-board、Display である。CPU と Memory をリモートに分離するのは遅延の観点から非現実的である。

5. むすび

我々はこれからの分散システムに要求される機能として、三つの仮定をおき、Element Computing という概念を提案した。また、この概念の実現性および利用可能性を検証するための方法と手順を示した。

今後は、実装に向けて、VM の選定と、Element、Abstract Bus、Abstract Network の仕様、設計を進める。