分散システムにおける柔軟な構成情報による障害復旧方式

1C－5

相澤 雅彦，宮内 直人，中川路 哲男，勝山 光太郎
三菱電機(株) 情報システム研究所

1　はじめに

従来の集中処理ではホストが一つであるため、構成情報には起動されているアプリケーション名が記述されていた。これに対し、分散処理ではサーバが複数であるため、アプリケーション名に加えサーバ名つまり位置情報が必要となる。

一方、分散処理では複製が可能であり、あるサーバの障害発生時に障害の発生したサーバで動作していたアプリケーションを他のサーバ上で動作させることができる。

本稿では分散処理においてアプリケーションの動作場所を固定した構成情報とアプリケーションの動作場所を選択する柔軟な構成情報を初期化時とサーバのダウン障害時において比較し、両者の長所を生かした方法を提案しそのシステム例を示す。

2　システムの構成

想定するシステム構成を図1に示す。システムは業務を行なうデータベースサーバ、サーバ、クライアントと管理を行なうマネージャ、エージェントから構成される。業務A、B、Cは業務内容が異なるが、使用するデータの一部が共通であるため一つのデータベースに業務A、B、Cのデータを格納している。各サーバのディスクには複数の業務アプリケーションがインストールされている。マネージャは各サーバのエージェントと通信を行ない管理を行なう。

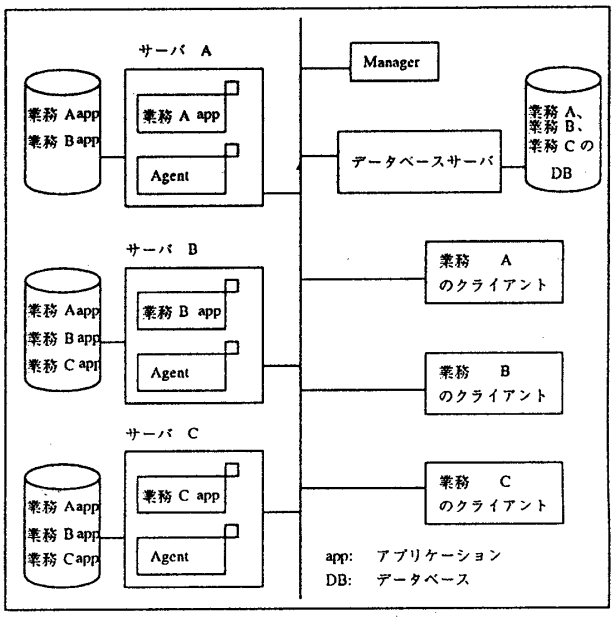

図 1: システム構成図

3　構成情報の検討

3.1　構成情報

上記のような分散システムについて次に示す固定的な構成情報と柔軟な構成情報について比較した。

固定的な構成情報は、ひとつのサーバに対して起動するアプリケーションのリストが記述されているものとする。これらのアプリケーションは、初期化時にすべて起動される。例えば図2の固定的な構成情報に基づくとサーバAでは業務A app(アプリケーション）が初期化時に起動される。

柔軟な構成情報は一つのアプリケーションに対し複数のサーバ名のリストからなるものとする。つまりアプリケーションがインストールされ起動可能な複数のサーバ名が構成情報に記述される。この複数のサーバからマネージャがアプリケーションを起動するサーバを決定する。例えば、図2ではアプリケーション業務A appはサーバA, サーバB、サーバCのうちから選択され起動される。

3.2　初期化時の検討

初期化時の起動時間について検討する。固定的な構成情報では構成情報通りに起動するが、柔軟な構成情報では、起動するサーバを選択する時間が必要である。その選択の組合せはアプリケーションの数とサーバの台数が増えるしたがって飛躍的に増加する。システム全体でア

A recovery method using a flexible configuration information in distributed systems
Masahiko AIZAWA, Naoto MIYAUCHI,
Tetsuo NAKAKAWAJI, Kotaro KATSUYAMA
Computer & Informatinon Systems Laboratory,
Mitsubishi Electric Corp.

固定的な構成情報

マシン名	アプリケーション名
サーバA	業務A app
サーバB	業務B app
サーバC	業務C app

柔軟な構成情報

アプリケーション名	マシン名		
業務A app	サーバA	サーバB	サーバC
業務B app	サーバA	サーバB	サーバC
業務C app	サーバB	サーバC	

両者の利点を持つ柔軟な構成情報

アプリケーション名	マシン名		
	初期化時	障害時	
業務A app	サーバA	サーバB	サーバC
業務B app	サーバB	サーバA	サーバC
業務C app	サーバC	サーバB	

図 2: 構成情報

プリケーションがM個あり、それぞれがインストールされているサーバ数をm(i) (アプリケーション番号i=1,2, …,M) としそのうち一つのサーバだけで起動するとすると初期化時の組合せは $\prod m(i)$ となる。そのため初期化時にアプリケーションの配置を自動的にかつ最適に決定するには多大な時間が必要となる。

3.3 障害時の検討

次にサーバがダウンした時の障害対策を検討する。固定的な構成情報を用いた障害対策には初期化時に複数のサーバ上で同一のアプリケーションを立ちあげておく方法がある。しかし、この方法では資源 (CPU, メモリ) の無駄が発生する。また、サーバAで作業の重いアプリケーションを実行し、サーバBでは作業の軽いアプリケーションを実行したとする。このとき障害対策のため、サーバBにも作業の重いアプリケーションを実行するとサーバBの負荷が高くなり、作業の軽いアプリケーションの動作に影響がでる問題が発生する。

それに対して柔軟な構成情報は、サーバがダウンする障害時に複数のサーバから現在の状況 (CPU負荷・キュー・メモリ容量・ディスク容量) を得て、障害が発生したサーバのアプリケーションを実行するサーバを選択することができる。ここでシステム全体でアプリケーションがM個あり、それぞれがインストールされているサーバ数をm(i) (i=1,2, …,M) とする。障害が発生したサーバで動作していたアプリケーションを他の場所で立ちあげる組合せは $\prod(m(k)-1)$(ただし k= 障害発生サーバで動作しているアプリケーション) となり、初期化時より少ない組合せで起動するサーバを選択できる。

この方法では、障害の発生しているアプリケーションがロックしているレコードにアクセスできないが、他のレコードにアクセスするクライアントは動作でき、みかけの平均修復時間を短縮できる。

3.4 検討のまとめ

このように固定的な構成情報では柔軟な構成情報よりアプリケーションの配置に決定する時間が必要ないため構成するのは早いが障害発生時には柔軟に負荷の低いサーバを選んで対処することはできない。一方、柔軟な構成情報は、初期化時に多量な組合せの中から適した配置を決定するには時間が必要だが障害が発生した時、複数の組合せから選択できる。そこで両者の長所をとり初期化時には固定的な構成情報を使用し、障害発生時には柔軟な構成情報を使用することにした。図2に両者の利点をもつ柔軟な構成情報に示す。この構成情報はアプリケーションに対して初期化時に起動するマシンと障害時に起動できるマシン名から構成される。たとえば業務A appは初期化時にサーバAで起動され、障害時にはサーバBまたはサーバCで起動される。

4 システムの動作

次にシステムの動作について説明する。マネージャは固定的な初期構成情報と障害時のための柔軟な構成情報をもつ。初期化時は初期構成情報に基づき、アプリケーションを起動する。マネージャは各サーバを監視する。

マネージャが監視しているサーバの障害を検知すると、マネージャは初期構成情報より障害のあるサーバで動作していたアプリケーションを検索し、検索したアプリケーション名をキーとし障害時のための構成情報からそれを動作できるサーバを検索する。マネージャは検索したサーバのエージェントから負荷情報をえる。獲得した負荷情報より、検索したアプリケーションの割当を決定し、各サーバは割り当てられたアプリケーションを起動する。復旧次第マネージャは復旧したサーバで初期構成情報に基づきアプリケーションを起動し、他のサーバの構成を初期の構成にするため、アプリケーションを割り当てられた各サーバの該当するアプリケーションを停止する。

5 おわりに

本稿では、固定的な構成情報と柔軟な構成情報の比較を行なった。そして、初期化時には固定的な構成情報を用い、障害時には柔軟な構成情報を使用する方法を提案した。今後の課題は、複数のアプリケーションの効率的な配置の検討である。