

分散サーバシステムの自律的な運用方法の検討

A Study on Autonomous Operation for Distributed Server System

寺下 雅人†

西谷 明彦†

大岸 智彦†

Masahito TERASHITA

Akihiko NISHITANI

Tomohiko OGISHI

1. はじめに

近年、クラウドコンピューティングが普及する中、多数のPCサーバを論理的に結合して、巨大な計算環境やストレージなどを提供する分散サーバシステム[1, 2]が注目されている。分散サーバシステムは、システム全体として耐障害性確保が容易なため、比較的故障が起きやすい低価格なPCによる構築が可能である。また、サービスの利用形態やユーザ数の拡大に合わせて、PC数、メモリ容量、HDD容量などのリソースを増強し、システム構成を柔軟に変更可能である。

一方で、複数のサーバで構成される分散サーバシステムにおいて、サーバ故障時の緊急性や増強すべきPC数やリソースを把握するには、サーバシステムの論理的な構成から、システムとしての状態を把握する必要がある。これには、分散サーバシステムの仕組みや構成に関する知識が必要であり、運用者に高いスキルが求められる。

そこで筆者らは、高いスキルを必要としない自律的な運用方法を目指して、個々のサーバをリソース毎に監視し、分散サーバシステムとしての障害時の緊急性やリソース不足に関わる情報を把握できる手法を考案した。本稿では、その概要について述べる。

2. システム設計方針

1章で述べた自律的な運用とは、分散サーバシステム全体の状態を判断して、できる限り運用者の介入無しにシステムの状態を自動的に改善する運用である。これを実現するため、システムの状態を示す指標として健全度を導入することとした。健全度が低い場合は、遠隔から、故障したサーバの代わりに予備サーバを起動するなど健全度を自動的に高めるための制御を行う。自動制御ができない場合は、運用者にアラーム通知を行う。従って、運用者のマニュアル操作は最終手段となり、出来る限り手間を削減することができる。

システム設計の前提条件として、問題を単純化するため、本稿での検討においては全てのサーバが同じ役割を持ち同列に接続し、ハードウェアスペックが同等のサーバ機を使用した分散サーバシステムを想定する。

3. 詳細設計

3.1. 運用システムの構成例

運用システムの構成例を図1に示す。管理サーバは、分散サーバシステムの自動制御を行うとともに、運用者にサーバおよびシステムの健全度の情報を提供する。分散サーバシステム上でサーバ障害が発生した場合、管理サーバは健全度を計算し、それを向上させるため、運用者が予め設置しておいた予備サーバへの切り替えを自動的に行うことで健全度の向上を図る。運用者は、運用システムを通して、システム全体の健全度を確認するとともに、予備サーバの計画的な設置や障害対応の緊急性を知ることができる。

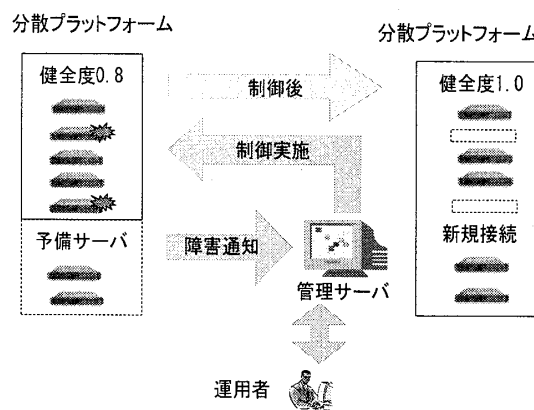


図1 運用システム構成例

3.2. パラメータと収集方法

健全度の算出にあたり使用するパラメータは、サーバ内のハードウェア(CPU、ファン、ネットワークカードなど)、ソフトウェア(サーバプロセスなど)、あるいは、リソースの容量(メモリ使用率など)に関わるパラメータである。これらは、IPMI(Intelligent Platform Management Interface)やSNMP(Simple Network Management Protocol)などの手段で収集する。

3.3. パラメータの正規化と健全度

各パラメータは取り得る値とその意味が異なるため、正規化を目的とし、パラメータ値の範囲ごとに、正常(1)、異常(0.5)、危険(0)の3つの属性(属性値)を定義する。パラメータと属性の例を表1に示す。

健全度は、上記の属性をパラメータ毎およびサーバ毎に集計したものであり、0~1の値を取り、1に近いほど正常であることを示す。

†株式会社 KDDI 研究所, KDDI R&D Laboratories, Inc.

表1 パラメータと属性の例

パラメータ名	正常(1)	異常(0.5)	危険(0)	重み
メモリ使用率	$80 < X$	$80 \leq X < 100$	$X = 100$	2
HDD使用率	$70 < X$	$70 \leq X < 90$	$X \leq 90$	1
ロードアベレージ	$3 < X$	$3 \leq X < 5$	$X \leq 5$	3

3.4. パラメータの健全度

あるパラメータの健全度の算出方法を以下に示す。今回のシステムでは全て同等のスペックのサーバを仮定しているため、各サーバのパラメータの属性値の平均をシステム全体におけるそのパラメータの健全度としている。例えば、図2のようにサーバ10台の分散サーバシステムを仮定する。ここでホスト5~9は全てのパラメータの属性が正常とする。この場合、ロードアベレージに関しては、ホスト2,3が異常、であるため、システム全体でのメモリ使用率に関する健全度は0.9となる。また、この例では、メモリ使用率の健全度が一番低いことが分かる。本情報を用いた運用として、システム全体としてメモリ使用率を向上させる制御が最も効果的であると考えられる。

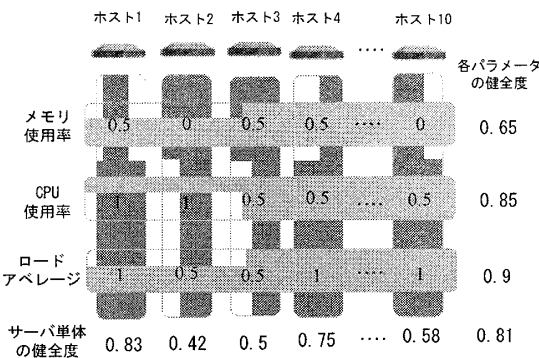


図2 パラメータ毎・サーバ毎の健全度の例

3.5. サーバ単体の健全度

サーバ単体の健全度の算出方法を以下に示す。表1の例に示すように、パラメータ毎の重みづけを行い、健全度は、 $(\sum(\text{パラメータ1の重み} \times \text{属性値})) / \text{全パラメータの重みの合計}$ により算出する。図2の例では、サーバ単体の健全度は下部に表示されている。例えば、ホスト2は、メモリ使用率が危険であり、ロードアベレージが異常であるため、健全度が0.42という低い数値となる。本情報を用いた運用として、健全度が低いサーバを再起動するといった運用が考えられる。

3.6. 時間軸での健全度

ここまでは、パラメータやサーバの健全度の算出方法について説明をした。パラメータは、3.2節の方法に従い定期的に収集され、これに伴い、健全度が定期的に計算される。本節では、健全度の時間軸での分析

方法について述べる。

あるサーバやあるパラメータの健全度が低い場合、その健全度が一時的な値か継続的な値かが分散サーバシステムの運用にとって重要になる。

そこで、まず表2に示すように、健全度についても正常、異常、危険の3つの属性に分類することとした。システム運用において、異常は経過観察が必要な状況、危険は緊急措置が必要な状況を仮定している。異常・危険状況の運用者への通知については、本属性の継続性を考慮して判断する。例えば、図3(a)のように危険が複数回連続する場合や、(b)のように単位時間内に危険が複数回発生する場合は、危険と判断する。一方(c)正常が複数回連続しており稀に危険が発生する場合は、正常と判断し、(d)のように異常の発生頻度が多いながらも正常や危険が発生する場合は、異常と判断する。

表2 健全度と属性の対応例

属性	正常(1)	異常(0.5)	危険(0)
健全度	$0.8 \leq X$	$0.5 \leq X < 0.8$	$0.5 < X$

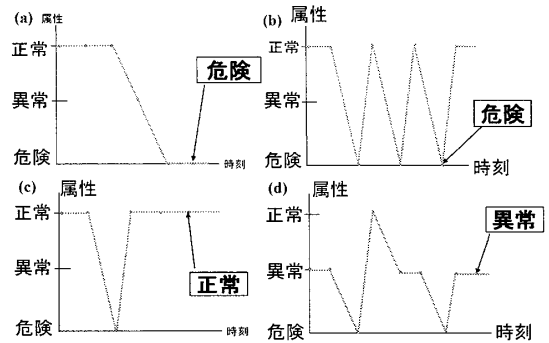


図3 運用者への通知判断

4. まとめ

本論文では、運用者に分散サーバシステムの状態を分かりやすく提供するための状態判断手法について述べた。本手法は、パラメータ、サーバ単体、時間軸において健全度という指標を用いた正規化を行うことを特徴とする。今後の課題として、異なる役割を持つサーバが存在する場合や、異なるスペックのサーバが存在する場合の状態判断手法の改良などが考えられる。

謝辞

日頃よりご指導頂く、株式会社 KDDI 研究所秋葉所長、中島副所長、及び鈴木取締役に感謝致します。

参考文献

[1] Tom White "Hadoop: The Definitive Guide," O'Reilly Media, May 2009.
 [2] 建部 修見, 曾田 哲之, "広域分散ファイルシステム Gfarm v2 の実装と評価," 情報処理学会研究報告, 2007-HPC-113, 2007.