CAC グループ企業から

UNIX／NT サーバー 高信頼性確保のための方策

株式会社アークシステム　サービス本部 テクニカルマネジャー　北 村 雅 人

１．イントロダクション

インターネットの普及により、企業ネットワークは従来の専用線によるスター型接続から、各種通信サービスを利用したメッシュ型接続に移行してきている。これを実現するためのハブ、ルーター、ネットワーク・サーバーなどが、多くのハードウェア・ベンダーから供給され、また、セキュリティを含む各種通信制御のためのソフトウェアも数多く出荷されている。このような製品群を組み合わせて構築した企業ネットワーク上で、既存の基幹系業務だけでなく、さまざまな企業サービスが提供されるようになってきている。ネットワーク形態の多様化に加え、サービス時間の24時間化により、ネットワーク全体として高い信頼性が必要になってきた。ここでは、ネットワーク・サーバーとして利用される、ハードウェアとソフトウェアについて考察する。

２．UNIX／NT サーバー利用のメリットとデメリット

UNIX／NT サーバーは低価格であること、およびインターネットで培った各種ソフトウェアの豊富さにより、ネットワークの核として利用されることが多い。また、UNIX／NT サーバーを利用することで、短納期でシステムを構築できるという利点もある。しかしながら、メインフレームの場合と比較すると、UNIX／NT サーバーは、ハードウェア、ソフトウェアとも相対的に可用性が低くなる傾向にある。これは、それぞれの設計思想に起因するものであり、早急な改善は望めない。

例えば、リソースの扱い方にその特徴が現れている。メインフレームでは、リソースは処理の要求ごとにシェアするものとして、OS が管理している。リソースの要求は、OS により優先度に応じて処理される。したがって、処理要求の集中によって輻輳（ふくそう）状態が発生した場合でも、要求は優先度順に処理されていく。

一方、UNIX／NT 系 OS においては、リソースは十分に用意されている前提で設計されており、リソースの要求は順番に行われるため、システムの処理であっても優先されることはない。したがって、リソースが不足した場合には、要求はキャンセルされる。輻輳状態が発生した場合には、システムの処理も同様にキャンセルされることがあり、OS の動作も保証されない。

さらに、ハードウェアの設計においても、UNIX／NT サーバーはメインフレームと比較すると、耐障害性の面で劣る部分がある。メインフレームでは、１つの部品の故障が全体の停止に結びつかないように代替の手段を用意している。例えば、エラー修正が多発した場合には、メモリバンクの自動切り離しが行われる。これに対し、UNIX／NT サーバーでは、この点について、まだ十分に考慮されていない。UNIX／NT サーバーにおいては、低コストで開発サイクルを早くするために、耐障害性について、ある程度犠牲にしている傾向が見受けられる。

最近では、可用性を飛躍的に向上させた NT 用のハードウェアも開発されているが、そうなると価格がメインフレームと大差なくなり、保守費などを考慮すると価格面でメリットがない。しかも、OS は Windows NT のままなので、ソフトウェア障害に対してはほとんど無防備である。システムの停止を招く障害原因の割合は、統計によるとソフトウェア40%、ハードウェア10%であり、このようなハードウェア面だけ信頼性が高い NT マシンを採用しても効果

は得られない[*1]。

UNIX／NT サーバーの単体での可用性は、統計によると99.7%である。つまり、単体のマシンで構成するシステムでは、８時間程度のサーバー停止を年３回（年間累積停止26時間相当）ほど許容する必要がある。それ以上の可用性を求める場合には、特に高信頼確保を目指したインフラ設計が必要である。

３．システム構成設計上の方策

本章では可用性を高めるための方策、および実例について、システム構成設計、およびシステム運用管理の側面から解説していく。

3.1　システム構成設計上の方策

前述のように UNIX／NT サーバー・システムは、ある程度のシステム停止を前提としてシステムを構築する必要があるが、ハードウェア、ソフトウェアの構成を工夫することにより、可用性の向上を図ることが可能である。

UNIX／NT サーバー・システムのハードウェア、ソフトウェアを組み合わせて高信頼システムを構築する場合には、複数のハードウェアを用意して、障害の発生時に処理を他サーバーが引き継ぐような複合系構成を用意する。その形態には、HA（High　Available）クラスタ構成やパラレル構成などがある。

3.1.1　HA クラスタ構成の特徴

HA クラスタ構成は、２台以上のサーバーを用意して１つのグループ（クラスタ）を構成する形態である。クラスタ内のサーバーに異常が発生した場合、他のサーバーが業務を引き継いで続行する。

HA クラスタ構成の特徴を以下に挙げる。

①障害発生の前後において、クラスタ内で共有するディスク上のデータは引き継がれる。

②ディスク障害には耐性がないため、RAID５などの構成により、ディスク自体に障害に対する耐性を持たせる必要がある。

③障害の検知から業務の引き継ぎ完了までは、５分程度の時間が必要。その間、障害の起きたサーバーの業務は停止する。

④接続中だったユーザーは、引き継ぎ完了後に再接続の動作が必要となる。

⑤アプリケーションの正常動作の確認要件として「一定のレスポンスを維持する」ことを加えると、異常なレスポンス低下時に待機系へ切り替えることも可能である。

⑥メモリ上の情報は引き継ぐことができない。

クラスタ構成はデータの一元管理を最優先とする、データベース・サーバーなどで利用される形態である。業務に応じて完全スタンバイ、相互スタンバイなどの構成が選択できる。処理を実行するアプリケーション・サーバーと、データを管理するデータベース・サーバーをそれぞれ用意して、相互スタンバイ構成を組むことにより、クラスタの最小構成となる（図１参照）。この場合、障害時にはある程度の性能劣化を許容する必要がある。

また、前述のとおり、ディスク障害についてはディスク自体の可用性に依存するため、RAID５などの障害耐性を持ったディスク装置を用意する必要がある。SAN(Storage Area Network）を利用するのも一案である。

3.1.2　パラレル構成の特徴

２台以上のサーバーを用意し、各サーバーに同一構成の環境を構築する。ロードバランサーなどを用いて、処理要求を各サーバーに振り分ける。あるサーバーで異常が発生した場合には、これをロードバランサーが検知し、異常の発生したサーバーへの処理要求の割り付けを中止する(図２参照)。

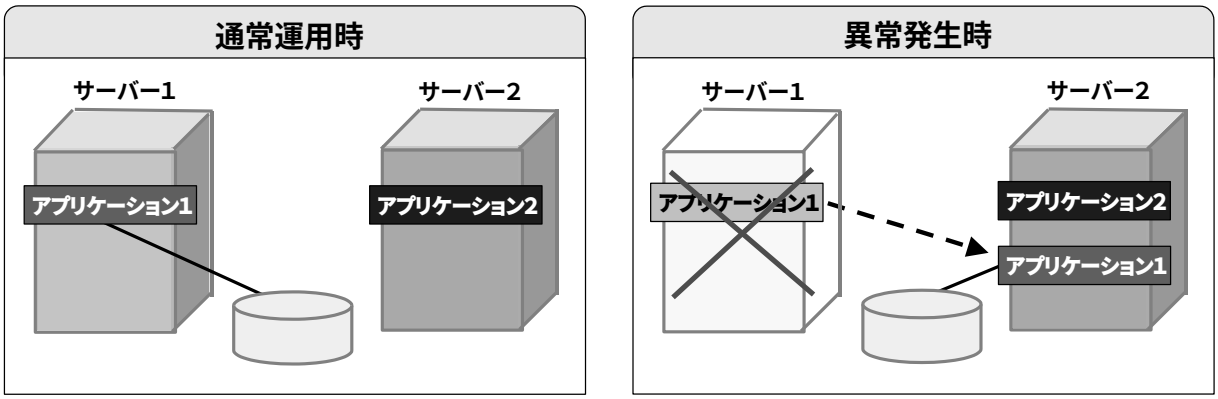

図１　HA クラスタ構成システム

＊１）Windows2000データセンター・サーバーについては、現時点では評価できていない。

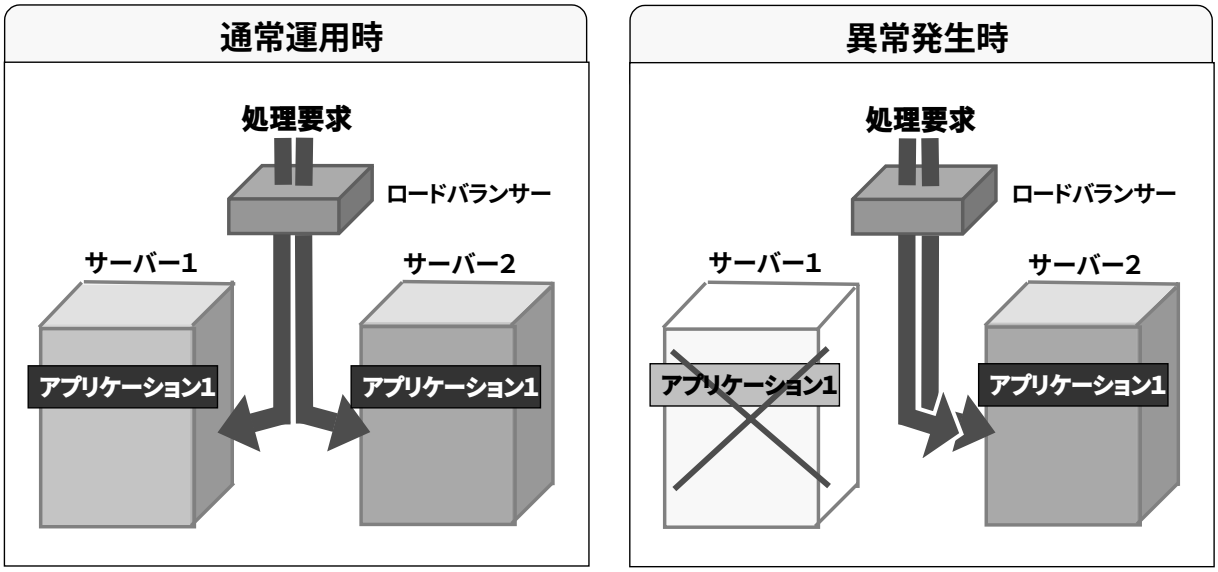

図2　パラレル構成システム

パラレル構成の特徴を以下に挙げる。

①サーバーは独立して処理を実行するため、障害の発生したサーバーの影響を受けない。逆にサーバー間でのデータ引き継ぎはできない。

②複数サーバーのディスク上にデータを保持するため、１つのサーバーでディスク障害が発生してもデータの完全な喪失を防げる。

③障害の検知から振り分け制限まで数十秒で完了し、ユーザーが接続できない時間が短い。

④接続中だったユーザーは、再接続の動作が必要である。

⑤アプリケーションの正常動作の確認要件として「一定のレスポンスを維持する」ことを加えると、異常なレスポンス低下の発生したサーバーを切り離すことも可能である。

⑥対象のサーバーを増設することで、処理能力の向上が可能である。

パラレル構成は、接続不可となる時間を極小化したい参照系の Web サーバーなどに利用される。また、サーバーの増設により、処理能力を容易に向上させることが可能なので、急激なアクセスの増加が予想されるインターネット向けの Web サーバーなどのシステムに向いている。

全体として高い信頼性を維持するためには、ロードバランサーとなるハードウェアに信頼性の高いものを採用することと、複雑な振り分けをさせないことがポイントである。また、サーバーの信頼性向上だけでなく、ロードバランサーをクラスタ化することにより、ロードバランサーの障害による停止を回避することもできる。

3.2　システム構成設計上の留意点

システムの高可用性を実現するための留意点はいくつかあるが、以下に見落としがちな点を挙げる。

⑴　ソフトウェアの最新版の採用を避ける

OS、DB／DC などのソフトウェア最新版は、機能も豊富で一見は便利そうに見える。だが、システムにとって必須の機能が搭載されていないのであれば、最新版の採用は避けるべきである。なぜなら、初期バージョンはソフトウェアのバグ処理が充分ではなく、障害対応に多大な労力と時間がかかる場合がある。また、止むを得なく採用した場合でも、そのバージョンから新規搭載された機能の利用は、不安定な場合が多いのでお薦めできない。可用性を追求するのであれば、二世代ほど前のバージョンから備わっている機能のみでシステムを構築することが望ましい。

⑵　ネットワーク機器の重要性を認識する

システム規模が大きくなると、構成部品や構成機器が多くなるため、ネットワーク全体としての信頼性を確保することが難しくなる。例えば、サーバーは二重化していても、ハブやルーターなどのネットワーク機器が障害に未対策である場合が多く見られる。ネットワーク機器の障害は、即座にシステム全体の停止に繋がる場合があるので、細心の注意を払う必要がある。

⑶　安定した電源を確保する

システム設計／付設の段階から UPS（Uninterruptible Power System：無停電電源装置）を導入し、ケーブルも品質の良いものを採用するなどの配慮が大切である。現在の、コンピュータ設置にあらかじめ配慮したビルディングなどでは、ビル付帯の電源設備として、UPS を付設しているところが増えてきている。ただし、電源設備には「法廷点検」が必要であり、受電設備（電力会社との接続点）

から二重化していないと、点検中に停電（瞬断）が発生する可能性がある。システムの重要度によっては、専用のUPSを用意する必要がある。

４．システム運用管理上の方策

4.1 運用設計の重要性

UNIX／NTサーバー・システムであっても、運用業務はメインフレームと大きく異なるわけではなく、運用設計に工夫を凝らすことにより可用性の向上を期待できる。資源管理、セキュリティ管理、稼動管理（オペレーション、自動運転）、変更／移行管理、問題管理など、メインフレームと同様の管理体系／管理水準を適用していく必要がある。

しかし、UNIX／NTサーバー・システムにおいては、システム構築や構成変更にスピードが要求される場合が多く、それぞれの管理がおざなりにされる傾向がある。また、私見ではあるが、メインフレームの管理者に比べて、UNIX／NTサーバー・システムの管理者は、先進的な技術を導入したがる傾向にあり、信頼性への配慮が薄いように感じる。高信頼性が要求されるシステムであるならば、管理レベルがインフラの種類によって異なってしまうのはおかしいことだ。

また、UNIX／NTサーバー・システムは、インフラが比較的低いコストで構築できるため、運用費用や担当者数までも低く見積もられてしまう傾向にある。その結果、運用フェーズに入ってから、運用費用や担当者工数が不足する場合が多い。しかし、サービスレベルを同等に維持しようとするならば、システム運用に関するランニング費用は、メインフレームと大差ないはずである。

さらに、UNIX／NTサーバー・システムは、メインフレームよりも運用業務の自由度が高く、オペレーターの裁量に頼る範囲が広くなっている。すなわち、オペレーションミスによる、システム停止の危険性が高くなるということである。UNIX／NTサーバー・システムは、メインフレームの場合よりも運用設計が重要なポイントとなることを認識しておきたい。

4.2 システム運用管理の留意点

システム運用を実行していく上で見落としがちな留意点を以下に挙げる。

高可用性システムの運用を考える上での参考にしていただきたい。

⑴ 定期的にリブートする

前述のような、高信頼性確保のために工夫を凝らしたシステムであっても、完全な無停止で運用することは難しい。私の経験したシステムでは、UNIXサーバーで６カ月程度、NTサーバー（4.0　SP５）については１カ月程度でリブート運用を実施している（ただし、FireWallに利用しているUNIXサーバーは除く）。

⑵ ディスク残量の減少に注意する

UNIX／NTサーバーでは、ファイルサイズを制限せずに運用することが多い。このため、気付かずにディスク容量を圧迫してしまい、最終的には残量が無くなってサービスが停止してしまうことがある。これは、運用で回避すべき問題であり、また運用で避ける以外に方法がない。システム管理の担当者はディスク残量を定期的に確認し、極端な減少傾向が見られたら、ただちに原因を究明することが望ましい。例えば、UNIXサーバーはリブート時に／tmpフォルダ内のファイルを削除するように設定されている。しかし、無停止運転を行う場合にはリブートする機会がないために、ディスク容量を恒常的に圧迫し続けてしまう。そこで、定期的に／tmpフォルダ内を意識的に削除する運用が必要となる。

⑶ ログファイルを定期的に整理する

システムの稼動状況を把握するためにはログが重要であるが、いつまでも保存しておくと、ディスク容量を圧迫してしまう。IBMのメインフレームであれば、JES２[*2]などのスプール機能を利用して、システム全体でのログ容量を管理すればよい。しかし、UNIX／NTサーバーでは、ログがファイルに出力される上に、ログの出力先が機能ごとに異なることが多いため、ファイルの管理を実施する必要がある。システム設計時に、管理が必要なログファイルを洗い出すことは当然であるが、運用開始後の３カ月程度はディスク容量の圧迫について、特に注意を払う必要がある。

５．UNIX/NTサーバー高信頼性確保の事例 ～A社 戦略情報システムサーバー～

5.1 システム概要

当社が実際に、UNIX／NTサーバーの高信頼性確保に携わった事例として、A社の事業戦略立案の中核となる情報データベース・サーバーについて紹介する。DB総容量は約２TBである。

システム構成は、２台のUNIXサーバーで、データ・ウェアハウス（DWH）とデータマート（DM）を構成する。システムの利用形態は、データ取り込み／加工を行うバッチ処理と、ユーザー要求により必要データを表示／出力するオンライン処理の２つからなる。

データの一元管理が必要なため、２台のサーバーはHAクラスタの構成を組んでいる。OSの障害発生時には、10

＊２）JES２：IBMのメインフレームのOSであるMVSで動作するジョブ管理機能

分程度のサービス停止と、片系運行時の性能劣化を50%程度とする許容条件でシステム設計を行った。障害発生時に処理中だったトランザクションは、自動的にフォールバックを行うが、オンライン中のユーザーは再接続の処理が必要である。また、実行中のバッチ処理は自動的に再実行する方式を採用しているため、前後JOBと関係なく、独立して実行できる単位でJOBの構成を組んでいる。

5.2 システム構成

(1) 使用ソフトウェア

・OS：HP－UX11.0
・HAソフトウェア：MC／Service Guard11.05
・ジョブ・コントロール・ソフトウェア：UNICENTER TNG2.0（9912a）
・DBソフトウェア：Oracle8.0.5

(2) HAクラスタ構成図

障害発生時には、10分以内のサービス再開を実現する目的で、クラスタ構成を組んでいる（図3参照）。

MC／Service Guardが全体の監視を担い、異常を検知した場合に、ルールにしたがって回復処置を実施する。例えば、Oracle DWHが監視コマンドに応答しない場合、Oracle DWHをDWHサーバー内で停止し、必要なディスク資源とユーザーが接続するIPアドレスをDMサーバーに引き継いで起動する。

5.3 障害回復の動作〔DWHサーバーのOS障害の場合〕

(1) システムの回復の流れ

①DMサーバーのMC／Service Guardは、DWHサーバーから応答が無くなった時点でサーバーダウンと判断する。（このタイミングで、DMサーバーが共有ディスク上にロックディスク[*3]を獲得して主導権を握る。これにより、DWHサーバーはOSのシャットダウンを強制的に実行される。）
②UNICENTER ManagerをDMサーバーで起動するため、UNICENTER Agentを停止する。
③UNICENTER Manager、およびOracle DWHで使用するIPアドレスを、DMサーバーのインタフェースに設定する。
④UNICENTER Manager、およびOracle DWHで使用するディスクを、DMサーバーにマウントする。
⑤Oracle DWHをDMサーバーで起動する。
⑥UNICENTER ManagerをDMサーバーで起動する。
⑦実行中ジョブを再起動する。

(2) バッチ処理の回復

DBの回復には、Oracleの自動回復機能を活用する。システムの回復処置により、実行中ジョブが再起動されるため、バッチ処理固有の回復処置はない。

(3) オンライン処理の回復

バッチ処理と同様に、DBの回復はOracleの自動回復機能を活用する。実行中の処理はロールバックされるため、障害発生時のユーザー処理は再実行が必要である。

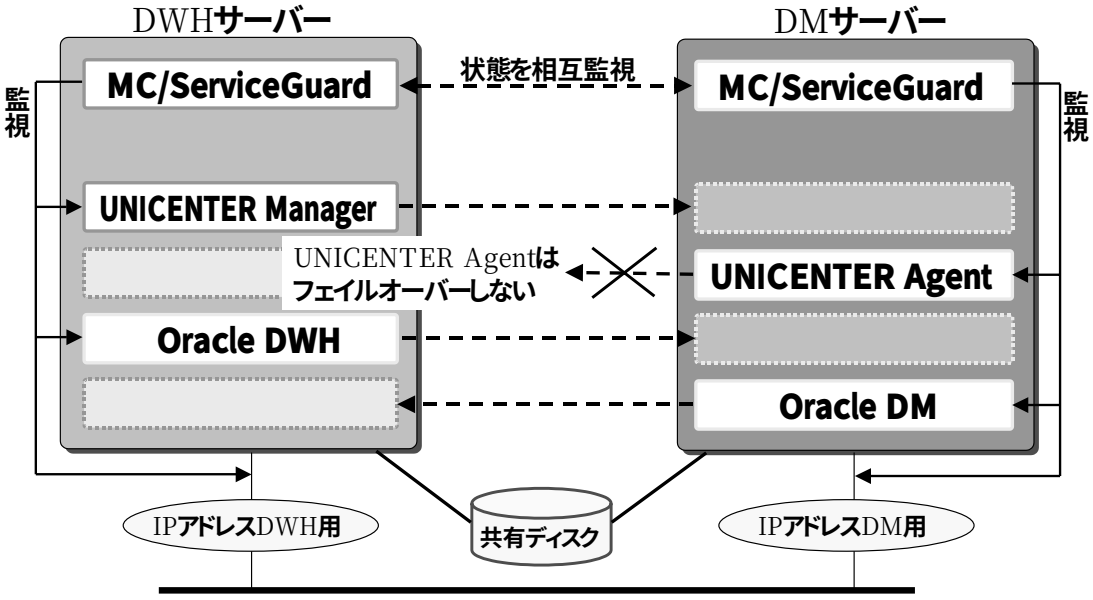

図3　A社システム構成

＊3）システム間の排他制御に使われているファイルを収容しているディスク。（これに対して強制的に排他ロックをかけると、ロックをかけたシステム以外は動作できなくなる。）

5.4 HA クラスタ構成設計上の着目点

HA クラスタを構成する場合には、監視方法に留意する必要がある。例えば、UNIX のプロセス（処理単位）の動作を監視する場合、プロセスの動作状況を監視するだけでは不十分である。プロセスは存在しても処理要求を受け付けない場合が多くあるためであり、可能な限りユーザーの利用形態に近い形での動作確認を行うべきである。

本システムの監視方法としては、Oracle に接続してテーブル参照できることを検分しており、プロセスの死活確認は補助的に実施している。プロセスの死活のみを監視に採用すると、HA クラスタが障害を検知できず、回復処理が実施されない場合がある。

5.5 課題と反省点

A 社戦略情報システムにおいては、障害発生時から10分程度でサービス再開を実現できたが、システム単体の信頼性についての課題が残った。システム導入において、ユーザー要件により、高性能なシステム処理の実現が最優先であったため、ハードウェア、OS、DB について、それぞれ最新のものを採用した。そのため、導入初期には新技術の採用には不可避の検証不足に起因した、システムダウンを含む障害発生による縮退運転が何度か発生してしまった。

それから問題点の解消まで半年ほどかかったが、以降は安定稼動させることができた。しかし、導入時に信頼性を重視して「枯れた技術」によるシステム設計をしていれば、運用開始の当初から安定したシステムを提供できたと思われる。

〈アークシステムの URL〉

http : //www.arksystems.co.jp/