3.24 障害中の運用に関する教訓（T24）

教訓 T24

サービス縮退時の対策を考慮せよ

問題

A社の統合システムは、基幹業務と情報提供業務をメインとして様々な対外システムやインターネットと連携したシステムである。

ある日、統合システムの3台でクラスタリング構成を組んだDBサーバが順次停止し、全DBサーバが停止した。原因が判明しなかったが、1台だけ稼働させたところ（図3.24－1①）、性能的に問題があるもののサービスを続けることができた。そこで、情報提供業務を停止させて基幹業務だけに専念させてサービスを継続させた（図3.24－1②）。

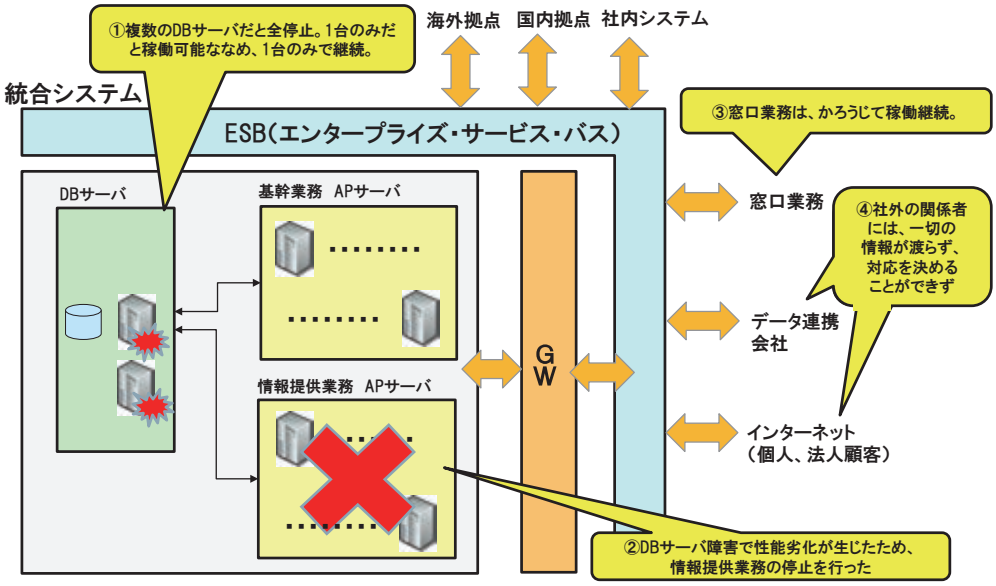

図3.24－1　障害状況

A社の窓口では、システム部門と状況を確認しながら業務を実施することができた（図3.24－1③）。しかし、データ連携会社は、情報提供を受けられなかったため、自社の業務を進めることができなかった。さらに、インターネットからの法人顧客や個人顧客も情報提供を受けられなかったため、直接電話でA社の窓口に状況を確認するしか方法がなかった（図3.24－1④）。そのため、A社の窓口も外部からの対応に追われてしまい、業務の遅れが徐々に大きくなり、混乱は終日続いた。

原因

統合システムの停止に至った直接原因は、DB サーバのソフトウェアのバグであった。

障害の影響が拡大したのは、情報提供業務を稼働させることができなかったためであった。全 DB サーバが停止したが、その後 1 台だけ稼働させ基幹業務を稼働させることができた。しかし、情報提供業務は、性能上の問題から停止せざるを得なかった。これは稼働当初からの懸案で、当時 1 台で性能要件を満たせる高性能のサーバが製品化されていなかった。

当初統合システムは、基幹業務が中心であった。数年後に情報提供業務を追加することになったが、あくまで付属的な意味合いで考えていた。そのため、大幅な業務要件の見直しを考えていなかった。

そのため、アプリケーションサーバは基幹業務アプリケーションサーバと情報提供業務アプリケーションサーバに分割していたが、DB サーバについてはひとつのインスタンス（1 論理 DB）構成としたままであった。

しかし、インターネットやスマートフォンを活用した業務へのニーズが増すにつれて、データ連携会社やインターネット（個人、法人顧客）に対する情報提供業務の全業務に占める割合は大きく拡大し、多くの外部関係者に重要なサービスを提供することが業務の中心になっていた。

今回のシステム障害では、統合システムが障害になりサービス縮退となった場合、対外システムやエンドユーザに「障害情報と復旧見込み」などの情報を発信することができなかったことが、システム障害の影響を大きくしてしまった。

根本原因は、サービス縮退が長期化することによって、問題が大きくなることを見逃していたことである。

対策

直接原因の DB サーバのソフトウェアのバグを改修した。また、業務制限を行わなくても 1 台で十分性能を満たせる DB サーバに入れ替えた。

さらに、根本原因で提示した「サービス縮退が長期化することによって、問題が大きくなること」を未然防止する対策を考えたい。

【対策 1】 エンドユーザの要求の変化を検証する

今回の事例における情報提供業務の追加のように、新システムを検討する場合は、お客様の要求変化を考慮し、基幹系業務とそれ以外の周辺システムを分けるなど、サービスの特徴、役割を考慮した設計を行う。そのために新システムを検討する場合は、要求定義の中で「お客様の要求の変化」を分析する。

その上で、設計時にリスク分析を行い、システム障害発生時のリスク低減策を講ずる。

【対策 2】ディペンダビリティ[13]を追求したシステム構成

【対策 1】の検証に基づき、今回のような事例においては、お客様の観点でシステム障害発生時の情報を提供できるように、業務の疎結合を考慮し DB サーバを基幹業務と情報提供業務に分離した DB サーバ構成とする。

その際、基幹業務サーバが障害のときに情報提供業務サーバで障害情報を提供するのはもちろんのこと、情報提供業務サーバが障害の場合でも、外部へ情報提供できる代替策を検討する（図 3.24－2）。

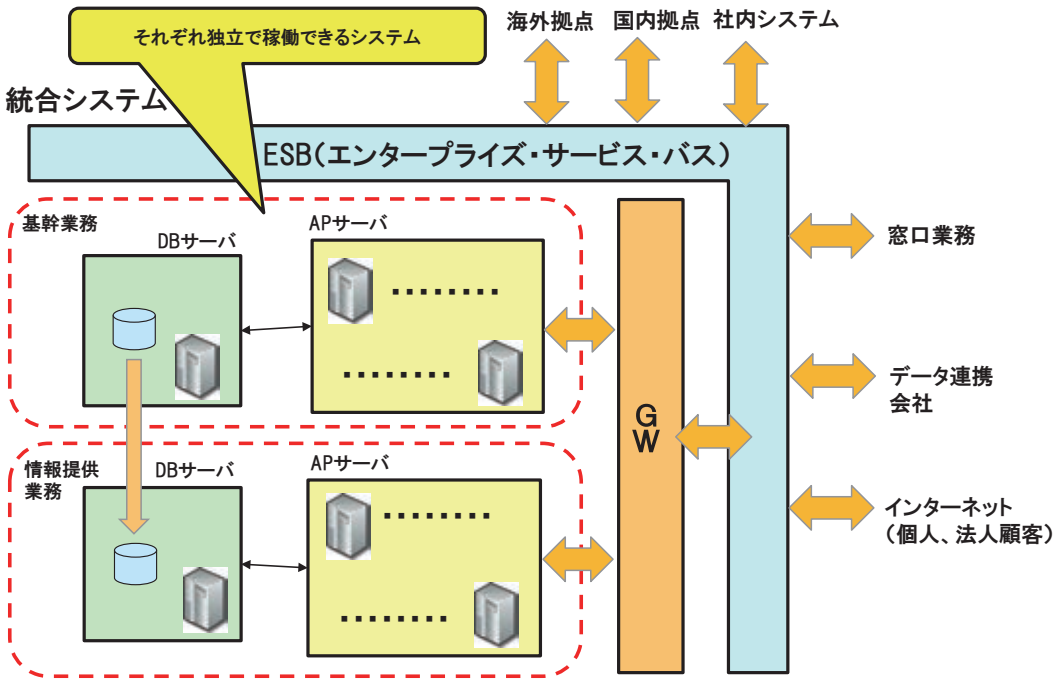

図 3.24－2　【対策 2】の一例

この事例のように途中で新たな要件が加わり、長年使用し続けているシステムのエンドユーザのニーズが変化することは、往々にして起こり得る。「システムを利用する顧客のどの要件に照準を当ててシステムを見直すか」は、重要なポイントである。

新たなシステムの要件を検討する場合は、例えば変化したステークホルダとの要件確認（あるいは合意）プロセスで、要件定義を再検討することも必要であろう。

[13] ディペンダビリティ（dependability）とは、例えば、システムの一部が壊れても残りの部分で補いながら、または機能を縮小させながら、稼働状態を保つといった自立的、自己修復的な動作を示す概念である。

【対策 1】と【対策 2】の対策をまとめると、「【対策 1】エンドユーザの要求の変化を検証する」ことにより、「【対策 2】ディペンダビリティを追求したシステム構成」のあり方が、方向づけられることになる。

効果

サービスの拡大にともない今まで付属的なシステムと思われていたものが、重要度を増すような変化がしばしば見られる中において、この事例のような「エンドユーザが要求する機能」の優先度も変化していくことがある。そのため、今までサービス縮退で停止しても問題にならなかった機能が、長時間停止することで問題になったりする。

前述した対策は、そのようなサービス縮退時の観点を見失わず、対策を検討するのに役立つ。

教訓

この事例では、設計時にリスク分析を行いディペンダビリティの確保に対応する最適なサービス縮退を考慮することにより、最適対策を取ることができる。

サービス縮退のあり方から、「システムの集中」が良いか、「システムの分散」が良いかなど、システム障害のリスクを軽減する構成を決めることも検討できる。

この教訓は、「サービス縮退時の対策を考慮せよ」とした。