ホットスタンバイシステムにおける共有ディスクの高速切替え方式

市川正也[†] 春名高明[†] 二瀬健太[†]
真矢譲[†] 三瓶英智[††]

本論文では，計算機システムの可用性向上のためのホットスタンバイシステムにおいて，システム切替え時における高速な共有ディスク切替え方式を提案する．本方式では，(1) デバイスドライバへのI/O 要求をトラップし制御ルーチンを呼び出し，ディスクへのアクセス可否を判定してエラー終了またはドライバを呼び出す擬似オフライン方式と，(2) 共有ディスクをグループ化し一括制御する制御グループ方式を組み合わせる．従来方式では共有ディスクの切替え時にI/O 処理が必須であり，この切替え時間はディスク数に比例し十数秒かかる．提案方式を実装した高可用化ミドルウェア Hitachi HA Booster Pack for AIX☆は，ディスク数に関係なく切替え時間を 2〜3 ミリ秒に短縮でき，システム切替え時間を十数秒程度に短縮できる．

Fast Shared Disk Switching Methods on a Hot-Standby System

Masaya Ichikawa,[†] Takaaki Haruna,[†] Kenta Ninose,[†]
Yuzuru Maya[†] and Hideaki Sanpei[††]

We propose two high-availability methods which support shorter switching time of the shared disks in a hot-standby system. The pseudo off-line method traps a device driver call in each I/O request, then executes the control routine which determines whether it calls the device driver or not. The control group method classifies the shared disks into a few groups, and controls the access to all disks in the same group simultaneously. In existing methods, since they are necessary to issue I/O (Input/Output) for switching shared disks, the switching time is proportional to the number of the shared disks. Then it takes ten and several seconds to switch the shared disks and the whole switching time becomes tens of seconds. In the proposed methods, it is not necessary to issue any I/O requests, the switching time does not depend on the number of the shared disks. We implement the proposed methods in Hitachi HA Booster Pack for AIX☆, a middleware which boosts system availability. We evaluate HA Booster, and verify that it takes only 2 or 3 milliseconds constantly to switch the shared disks, and the whole switching time becomes shorter than twenty seconds in our evaluation environment.

1. はじめに

近年，計算機の性能向上にともない，計算機システムを用いたサービスが多様化している．これらのサービスはオンライントランザクション処理が中心となり，24 時間 365 日連続運転が要求されている．

このようなオンラインシステムでは，計算機システムを停止させないこと，あるいは障害が発生してもユーザに障害を意識させない程度の短い時間（十数秒程度）で復旧させる高速回復技術が重要な課題となる．

ところで，従来から高可用化の手法として，多数の計算機から構成されるクラスタ方式，現用の計算機（現用系），待機の計算機（待機系）および共有ディスクから構成されるホットスタンバイ方式が広く実用化されている[1)〜7)]．これらの方式では，現用系で障害が発生すると，待機系は障害発生した現用系の処理をただちに，新たに現用系として稼動開始する．この処理をシステム切替え処理と呼ぶ．このシステム切替え処理では，切替え時間の短縮が重要であるため，障害の早期検出，共有ディスクや LAN（Local Area Network）

† 株式会社日立製作所システム開発研究所
Systems Development Laboratory, Hitachi, Ltd.
†† 株式会社日立製作所ソフトウェア事業部
Software Division, Hitachi, Ltd.

☆ AIX は，米国における米国 International Business Machines Corp. の登録商標です．

などの共有リソースの高速切替え，およびアプリケーション（以下，AP）の処理データの速やかな引継ぎが要求される[8)~10)]．

しかしながら，データベース（以下，DB）などのAP引継ぎ時間は，DBサイズやジャーナル量などに比例して長くなるという課題がある．このため，障害検出処理や共有リソースなどオペレーティングシステム（以下，OS）に関する切替え処理時間はできる限り短縮させ，ほとんど1，2秒程度に抑える必要がある．

本論文では，従来からの高可用化方式としてクラスタ方式とホットスタンバイ方式を比較し，システム切替え時間の短縮が可能なホットスタンバイ方式を選択する．そして，システム切替え時間をさらに短縮させるために，共有ディスクの高速切替え方式を提案する．以下，2章では従来の高可用化方式を，3章では提案方式が対象とするホットスタンバイ構成を，4章では提案方式の概要を説明する．さらに，5章では提案方式の処理手順を示し，6章では提案方式による共有ディスク切替え時間の実測結果を示し，考察を加える．

2. 従来の高可用化方式

従来の高可用化方式は，**図1**のようにシステム構成と待機系の設定方法の違いにより，現用系と待機系の2台の計算機で構成されるホットスタンバイ方式，3台以上の計算機をLAN，共有ディスクあるいは共有サーバ[11),12)]で接続するクラスタ方式に分類される．さらにクラスタ方式は，待機系を固定する共通待機方式と任意のノードを待機とする浮動待機方式に分類される．

システム切替え時間とシステム拡張性の観点から各高可用化方式の特徴を**表1**に示す．以下，システム切替え処理を構成する障害検出処理，共有ディスク切替え処理，およびAP引継ぎ処理の高速化という観点から，各方式を比較する．

(1) 共通待機方式

共通待機方式は3台以上の計算機から構成され，そのうち1台をつねに待機として稼動させる．待機系はすべての現用系のバックアップ処理を行うため，待機系のAPは事前にファイルオープン処理などを実行できない．また待機系は，共有ディスクを擬似的にオンライン化する高速な共有ディスクの切替え方式も適用できない．

共通待機系はすべての現用系のバックアップ処理を行うため，すべての現用系とInput/Output（以下，I/O）を共有しなければならない．この結果，現用系の数が増加すると待機系の共有ディスクなどのI/Oポート数がネックとなり，拡張性は多くても3，4台程度である[13)]．

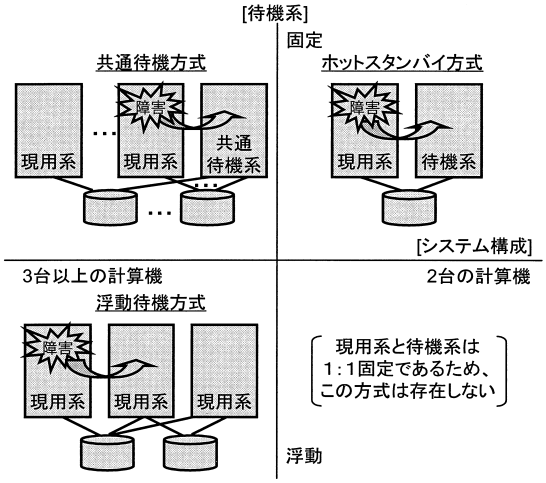

図1 従来の高可用化方式
Fig. 1 Existing high-availability methods.

表1 高可用化方式の特徴
Table 1 Characteristics of high-availability methods.

方式名		ホットスタンバイ方式	クラスタ方式	
			共通待機方式	浮動待機方式
システム切替え時間	障害検出時間	△	△	△
	AP引継ぎ時間	○	×	×
	共有ディスク切替え時間	○	×	×
システム拡張性		×	△	○

(2) 浮動待機方式

浮動待機方式は3台以上の計算機から構成され，通常稼動時には待機系がなく，現用系のみで稼動する．障害発生時には障害の箇所に応じて，引継ぎ処理を行う計算機を決める方式であり，プロセスペア方式[14),15)]やプロセス二重化方式[16)]が提案されている．これらの方式はプロセス単位の切替え処理を行う．このため，本方式は容易にシステムを拡張できるが，APの事前処理や共有ディスクの高速切替えは実行できない．

(3) ホットスタンバイ方式

ホットスタンバイ方式では，現用系，そのバックアップ処理を専用に行う待機系，共有ディスクやLANなどの共有リソースから構成される．現用系は，障害が発生しても，待機系が定期的にチェックポイントデータを待機系に転送している[17)]．

従来よりシステム切替え時間を短縮させる方法として，現用系はI/O発行時にチェックポイントデータを待機系に転送し，切替え処理ではジャーナルによる回復を不要とするI/O同期方式[18)]，および現用系と待

機系は同時に端末からの電文を受信し，現用系で障害が発生しても，リンクの切断を防止する端末無中断方式[19] が提案されている．

現用系と待機系は 1 対 1 に対応しているため，待機系はシステム立ち上げ時に現用系と同一の AP がファイルを事前にオープンさせ，AP 引継ぎ時間を短縮させている[20]．

同様に，待機系はシステム立ち上げ時に，時間のかかる共有ディスクのオンライン化処理を行うことにより，共有ディスクの高速切替え処理が可能になる．しかしながら，現用系と待機系はセットにして拡張しなければならず，容易にシステムは拡張できない．

そこで本研究では，システム拡張性よりシステム切替え時間の短縮を重視するため，ホットスタンバイ方式をベースに共有ディスクの高速切替え方式を検討することにした．なお，障害検出処理は，各方式とも現用系の OS 障害発生を契機に待機系に即時通知する検出方式を適用することにより高速化できる．

3. ホットスタンバイ方式

本章では，ホットスタンバイ構成とシステム切替え時間，および従来の共有ディスク切替え方式の問題点を説明する．

3.1 システム構成

ホットスタンバイ構成を**図 2** に示す．ハードウェア構成とソフトウェア構成に分けて説明する．

(1) **ハードウェア構成**

ハードウェアは，現用系とこれをバックアップする待機系，および共有ディスクから構成される．現用系と待機系はそれぞれローカルに CPU とメモリを持ち，それぞれの系は 1 つの共有ディスクに接続されている．共有ディスクは，Physical Volume（PV），Volume Group（VG），Logical Volume（LV）という 3 つの概念から構成される．PV は物理的なディスク装置，VG は 1 つ以上の PV から構成されるディスク装置の集合，そして LV は VG の領域に確保された論理的なディスク装置を示す．OS や AP は VG または LV を対象として I/O を発行する．これにより，柔軟なディスク管理を実現している．

(2) **ソフトウェア構成**

ソフトウェアは，系監視プログラム，OS のカーネル，本論文で提案する共有ディスク高速切替え方式を実装した高可用ミドルウェア Hitachi HA Booster Pack for AIX（以下，HA Booster），デバイスドライバ，および業務を行う AP（DB など）から構成される．現用系と待機系の系監視プログラムは，自系の障害監視や障害時のシステム切替え処理を行う．また，現用系と待機系が同一の領域を更新してデータを破壊しないよう，共有ディスクへのアクセス権を排他制御し，待機系からのアクセスを禁止する．HA Booster は，本論文で提案する共有ディスク高速切替え方式を実装しており，系監視プログラムと連携して共有ディスクへのアクセス権の排他制御を行う．

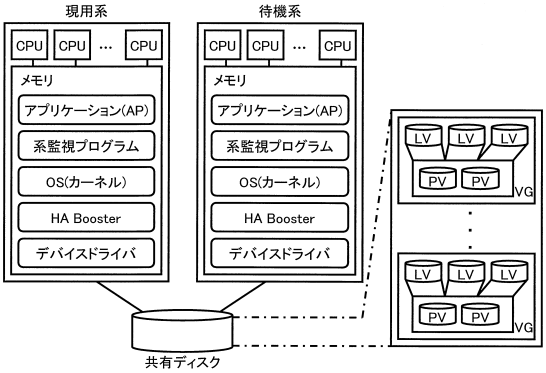

図 2 ホットスタンバイシステムの構成
Fig. 2 Configuration of a hot-standby system.

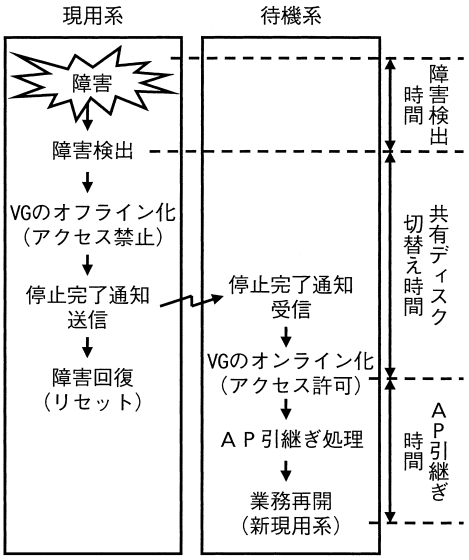

図 3 システム切替え時間
Fig. 3 System switching time.

3.2 システム切替え時間

システム切替え処理は**図 3** に示す障害検出処理，共有ディスク切替え処理，および AP 引継ぎ処理からなり，これらの処理時間の和をシステム切替え時間とする．

(1) **障害検出処理**

現用系の系監視プログラムは，自系の障害を検出すると，以下のシステム切替え処理を開始する．

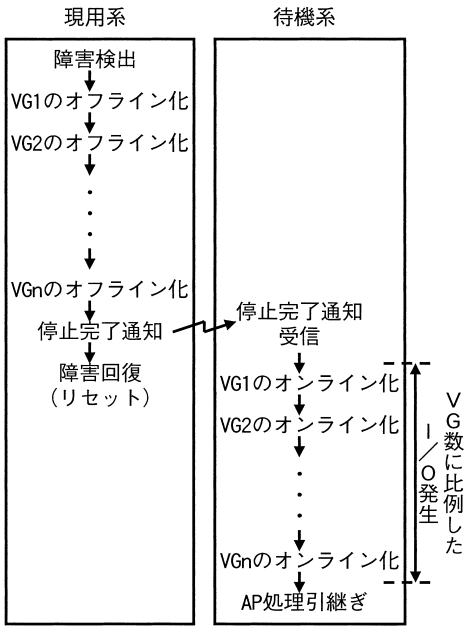

図 4 従来方式による共有ディスク切替え処理
Fig. 4 Shared disk switching operation of the existing method.

(2) **共有ディスク切替え処理**

現用系の系監視プログラムは，障害によるデータ破壊を抑止するため，現用系からの共有ディスクへのアクセスを禁止する．続いて，現用系の系監視プログラムは，待機系に停止完了通知を送信する．待機系の系監視プログラムは，この通知を受けて待機系からの共有ディスクへのアクセスを許可する．

(3) **AP 引継ぎ処理**

待機系は，DB の回復処理などの AP 引継ぎ処理を行う．

3.3 従来の共有ディスク切替え方式

従来方式による共有ディスク切替え処理を**図 4** に示す．この方式はデバイスドライバが有する VG のオンライン化処理，およびオフライン化処理の機能を用いて共有ディスクを切り替えていた．VG のオンライン化とは，デバイスドライバが VG の制御情報を PV 内の特定領域から読み込む処理である．これにより，OS や AP から VG や LV へのアクセスが可能となる．そして，VG のオフライン化とは，デバイスドライバが保持していた VG の制御情報を破棄し，OS や AP から VG や LV へのアクセスを不可能にする処理である．本論文では，オンライン化された VG の状態をオンライン状態，オフライン化された VG の状態をオフライン状態と呼ぶ．

通常稼動時は，待機系における AP の誤動作や利用者の誤操作などによるデータ破壊を防ぐため，現用系のみ VG をオンライン化し，待機系は VG をオフライン化する．これにより，現用系からのみ VG の内容の読み込みや更新が可能となる．

障害発生時は，現用系は VG のオフライン化処理を実施し，続いて待機系で VG のオンライン化処理を行うことで，共有ディスクの切替えを実現している．このとき，待機系でのオンライン化処理において VG の構成情報を取得するための I/O が VG ごとに発生するため，共有ディスク切替え時間は VG 数に比例して増加するという問題があった[21]．

4. 提案方式

本章では前節の問題点を解決するために，共有ディスク高速切替え方式として，I/O 要求時のデバイスドライバ呼び出しをトラップしてアクセス制御ルーチンを呼び出す擬似オフライン方式と，複数の VG を一括してアクセス制御を行う制御グループ方式を提案する．この 2 方式を組み合わせることにより，従来方式の問題点であったオンライン化時の I/O 処理を抑止できる．

4.1 擬似オフライン方式

本方式の概要を**図 5** に示す．本方式は，VG への I/O 要求時に発生するデバイスドライバ呼び出しをトラップして，VG へのアクセス可否を判定するアクセス制御判定ルーチン（以下，判定ルーチン）を呼び出させることで，デバイスドライバ呼び出しの直前にアクセス可否を判定する方式である．

HA Booster は，内部に判定ルーチンとアクセス制御状態フラグ（以下，状態フラグ）を設ける．判定ルーチンは，デバイスドライバの処理ルーチンと同一のインタフェースを持ち，当該処理ルーチンへの I/O 要求を受け付けることができる．この判定ルーチンは，後述する状態フラグが示す値に応じて，デバイスドライバの処理ルーチンを呼び出すか，当該呼び出しをエラー終了させるかのいずれかを行う．状態フラグとは，判定ルーチンでの挙動を規定する値を格納し，実体はメモリ上に確保された領域である．この値は「VG へのアクセス許可」あるいは「VG へのアクセス禁止」を意味する．そして，状態フラグの値は系監視プログラムからの要求により書き換える．

現用系と待機系は，システム立上げ時に以下の処理を行うことにより，I/O 要求時のトラップ処理を実現する．

(1) システム起動時に，VG をオンライン化する．そして，HA Booster はカーネルが所有するデバイス

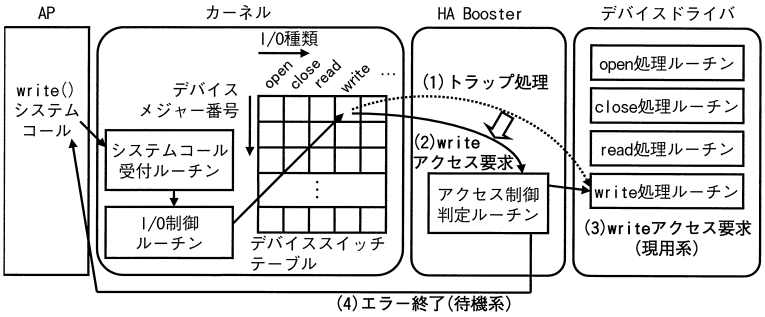

図 5 擬似オフライン方式
Fig. 5 Pseudo offline method.

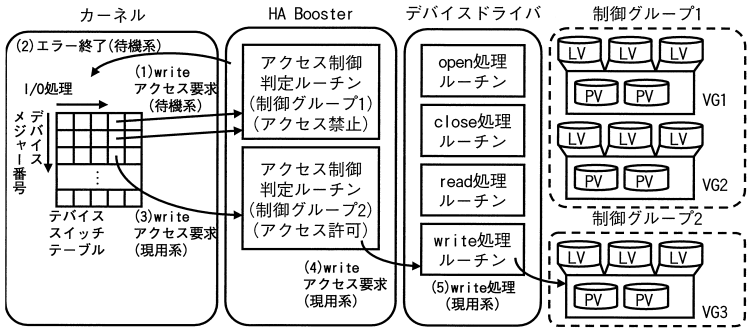

図 6 制御グループ方式
Fig. 6 Control group method.

スイッチテーブルを書き換え，判定ルーチンを登録する．デバイススイッチテーブルとは，カーネルが I/O 要求に応じてデバイスドライバを呼び出す際に用いるテーブルであり，各デバイスに対しカーネルが割り当てたデバイスメジャー番号と，このデバイスに対する各種 I/O（open，close，read，write など）を実施する処理ルーチンのアドレスを対応付けている．I/O 種類ごとに処理ルーチンを登録するためのエントリが設けられており，HA Booster は制御対象の VG の write 処理に対応するエントリを書き換え，判定ルーチンのアドレスを登録する．ここで，read 処理に対するアクセス制御を行う場合は，read 処理に対応するエントリを書き換える．

(2) カーネルや AP が VG に対する I/O を要求すると，カーネルは HA Booster の判定ルーチンを呼び出す．

(3) 判定ルーチンは VG が属する制御グループの状態フラグを参照し，これが「VG へのアクセス許可」を示す場合（現用系の場合），デバイスドライバの処理ルーチンを呼び出す．

(4) 状態フラグが「VG へのアクセス禁止」を示す場合（待機系の場合），デバイスドライバを呼び出さず，AP にエラー終了を通知する．

このように，現用系と待機系はメモリの値を書き換えるだけで状態フラグを変更できる．この結果，現用系と待機系は VG をオンライン状態に保ったまま VG へのアクセス制御が可能となるため，従来方式で発生していたオンライン化処理時の I/O を抑止できる．

4.2 制御グループ方式

制御グループ方式を**図 6** に示す．本方式は，1 つまたは複数の VG を「制御グループ」というグループに分類し，VG へのアクセス可否の設定を制御グループ単位に一括して実施する方式である．

HA Booster は制御グループごとに判定ルーチンと状態フラグを対応付ける．システム起動時に行うデバイススイッチテーブルの書換え処理で，VG が属する制御グループの判定ルーチンを登録する．

これにより，同一の制御グループに属する VG への I/O 要求が発生すると，同一の判定ルーチンが呼び出される．判定ルーチンは当該判定ルーチンと対になった状態フラグを参照し，アクセス制御を行う．したがって，ある制御グループに対応した状態フラグを変更すると，当該制御グループに属する全 VG に対し同一のアクセス制御を実施できる．

図 6 では，VG1，VG2 という 2 つの VG を制御グループ 1 とし，VG3 を制御グループ 2 とする．そして，VG に対する write 処理について判定ルーチンを登録し，制御グループ 1 に対応する状態フラグを「VG へのアクセス禁止」，制御グループ 2 に対応する状態フラグを「VG へのアクセス許可」に設定する．各制御グループに属する VG や LV に対して write アクセス要求が発生したときの処理手順を以下に示す．

(1) VG1 および VG2 への write アクセス要求が発生すると，カーネルはデバイススイッチテーブルの内容に従い，制御グループ 1 の判定ルーチンを呼び出す．

(2) 制御グループ 1 の状態フラグは「VG へのアクセス禁止」のため，判定ルーチンは当該要求をエラー終了させる．

(3) VG3 への write アクセス要求が発生すると，カーネルは制御グループ 2 に対応した判定ルーチンを呼び出す．

(4) 制御グループ 2 に対応した状態フラグは「VG へのアクセス許可」のため，判定ルーチンはデバイスドライバの write 処理ルーチンを呼び出す．

(5) デバイスドライバは，要求に応じディスクへの write 処理を行う．

以上により，アクセス制御の対象とする VG を 1 つの制御グループとして定義すれば，1 回のフラグ操作により制御グループに属する VG を一括して制御できる．なお，write 処理以外のアクセス要求に対しても，同様の処理を実施することにより制御できる．

4.3 方式比較

本節では，提案方式の用途を明確にするため，提案方式間の効果と欠点を示し，その後提案方式と従来方式を比較する．

4.3.1 提案方式の比較

提案方式の概要，効果，欠点，およびコスト（I/O 処理の増加オーバヘッド）を**表 2** に示す．両方式ともにソフトウェアで実現しており，特殊なハードウェアは不要である．擬似オフライン方式におけるコストは，判定ルーチンでのフラグ確認だけであり，10 命令程度と非常に小さい．また，制御グループ方式におけるコストは，システム立上げ時のデバイススイッチテーブル書換え処理の際に同一の判定ルーチンを登録するだけであり，I/O 処理時のコストはゼロである．

4.3.2 提案方式と従来方式の比較

提案方式と従来方式の特徴を**表 3** に示す．従来方式では，待機系での VG のオンライン化処理時に VG の構成情報を取得するために，VG に比例した I/O 時間を要する．一方，提案方式では VG をオンライン状態に保ったままアクセス制御が可能となるため，VG のオンライン化処理で発生していた I/O 処理が不要となる．また，制御グループの状態フラグの書換え処理のみで，制御グループに属する VG のアクセス制御を一括して実施できる．このため，提案方式は従来方式に比べシステム切替え時間を大幅に短縮できる．

表 2 提案方式の比較
Table 2 Comparison of the proposed methods.

方式名		擬似オフライン方式	制御グループ方式
概要		I/O 要求時のドライバ呼び出しをトラップして判定ルーチンを呼び出し	複数 VG に対するアクセス制御を同一の状態フラグと判定ルーチンで実施
効果		切替え時の I/O を抑止	複数 VG を一括制御
欠点		処理時間が VG 数に比例	グループ内の VG に対する個別のアクセス制御状態設定が不可
コスト	追加ハード	無	無
	オーバヘッド	10 命令程度	0

表 3 提案方式と従来方式の比較
Table 3 Comparison of the existing method and the proposed methods.

方式名		従来方式	提案方式
I/O 処理	現用（障害）系の VG オフライン化処理	要	不要
	待機（新現用）系の VG オンライン化処理		
制御グループ単位の一括制御		無	有

5. 処理手順

提案方式によるホットスタンバイの処理手順をシステム起動時とシステム切替え処理時に分けて**図 7** に示す．

5.1 システム起動時

現用系と待機系のシステム起動手順を示す．

(1) **制御グループの定義**

現用系と待機系は HA Booster 起動後に，AP が利用する VG を制御グループとして設定する．つまり，制御対象の VG に対して，擬似オフライン方式によりデバイススイッチテーブルを書き換え，HA Booster によるアクセス制御を有効にする．制御対象の VG が複数存在する場合は，制御グループ方式により同一の制御グループへ登録することで，一括したアクセス制御を有効にする（図 7 (a)）．

(2) **制御グループへのアクセス可否の設定**

現用系と待機系の HA Booster はそれぞれ，現用系では「VG へのアクセス許可」，待機系では「VG へのアクセス禁止」に状態フラグの値を設定する．これに

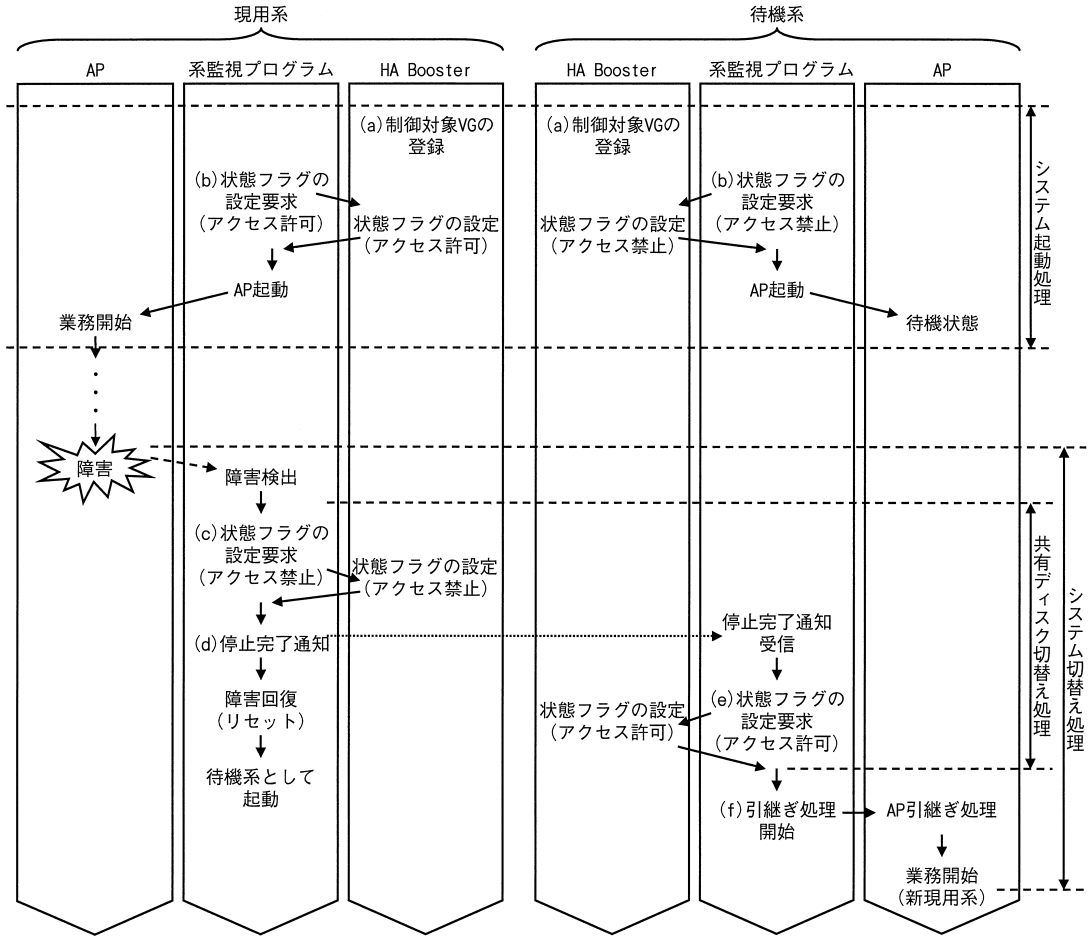

図 7 提案方式によるホットスタンバイの処理手順
Fig. 7 Hot-standby operation of the proposed methods.

より，現用系のみ VG への I/O が可能となり，待機系からは I/O が不可となる（図 7 (b)）．

5.2 システム切替え処理

現用系で障害が発生すると，現用系および待機系は以下の手順でシステム切替えを行う．

5.2.1 現 用 系

(1) 現用系の系監視プログラムは，HA Booster に制御グループのアクセス禁止を要求する．HA Booster は，制御グループの状態フラグを「VG へのアクセス禁止」に変更する．これにより，これ以後 I/O 要求が発生してもすべて実行されずエラーとなる（図 7 (c)）．

(2) 現用系は待機系に停止完了を通知する（図 7 (d)）．

5.2.2 待 機 系

(1) 待機系は現用系から共有ディスクへのアクセス許可を受信すると，待機系の系監視プログラムは HA Booster に対し VG へのアクセス許可を要求し，HA Booster は制御グループの状態フラグを「VG へのアクセス許可」に変更する（図 7 (e)）．

(2) 待機系の系監視プログラムは，AP 引継ぎ処理を行う．これにより，待機系は新現用系として業務を再開する（図 7 (f)）．

6. 評 価

前章で提案した擬似オフライン方式と制御グループ方式について，VG 数に対する共有ディスク切替え時間の変化を実測し，従来方式と比較する．さらに，障害検出および AP 引継ぎ処理を含めたシステム切替え時間を考察する．

6.1 共有ディスク切替え時間

6.1.1 測 定 環 境

測定環境を**表 4** に示す．この環境において，VG 数に対する共有ディスク切替え処理時間を 3 回測定し，平均処理時間を算出した．測定結果は**図 8** に示すとおりである．

表 4 測定環境
Table 4 Measurement environment.

使用マシン（EP8000 690）	CPU	PowerPC 1.3 GHz
	CPU 数	8 WAY
	二次キャッシュ	720 KB
	メモリ	16 GB
VG 構成	PV サイズ	18 GB
	PV 数	1 PV/VG
	VG 数	1〜72
OS		AIX 5L Version 5.2

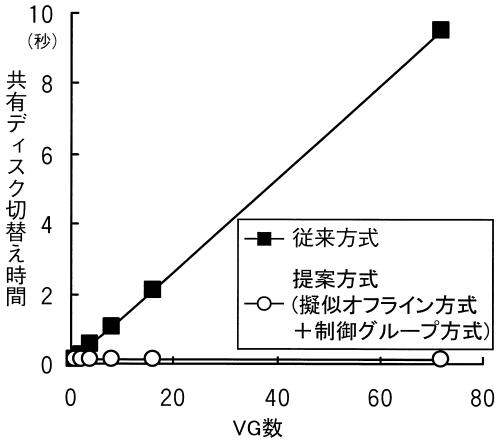

図 8 共有ディスク切替え時間
Fig. 8 Switching time of the shared disk.

6.1.2 考　　察

(1) **提案方式**

提案方式では，擬似オフライン方式と制御グループ方式を提案した．擬似オフライン方式によるシステム立ち上げ時の VG のオンライン化処理によりディスク切替え時の VG オンライン化処理が不要となる．また，制御グループ方式による VG 一括切替えにより切替え時間が VG 数に依存しない．この切替え時間は VG 数が増加しても，2〜3 ミリ秒程度と一定である．

(2) **従来方式**

従来方式では，VG ごとに現用系でのオフライン化処理と待機系でのオンライン化処理を行う．このオンライン化処理は，VG を構成する PV から VG の構成に関する情報を読み出すために I/O を発行している．このため切替え時間は VG 数に比例し，VG あたり約 0.13 秒の割合で増加する．

6.2 システム切替え時間

ユーザ環境でシステム切替え時間が，ユーザニーズに満足できるものかどうか検証するために，障害検出時間と AP 引継ぎ処理時間を含めたシステム切替え時間を考察する．

障害検出時間は，従来方式と提案方式ともに，障害をただちに待機系に通知する方式をとっており，1 秒程度で検出可能である．

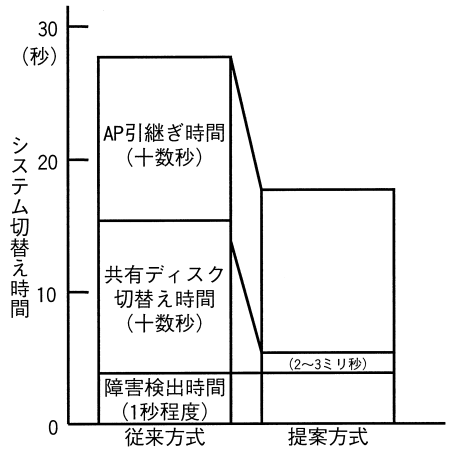

図 9 システム切替え時間
Fig. 9 Evaluation of the system switching time.

AP 引継ぎ時間を算出するにあたり，AP として HiRDB（リレーショナル DB）を用いた．DB の引継ぎ処理は，データサイズやジャーナル量に依存し，引継ぎ時点での未完了トランザクションの量により変動する．システム起動時に HiRDB を事前に起動させておくことにより，合わせて十数秒程度に抑止できる．

この結果，従来方式のシステム切替え時間は**図 9** に示すように 20〜30 秒であったが，提案方式によりこれを約半分の十数秒に短縮できる見通しを得た．

7. おわりに

本論文ではホットスタンバイシステムにおける共有ディスクの高速切替え方式として，擬似オフライン方式と制御グループ方式を提案した．擬似オフライン方式は，デバイスドライバへの I/O 要求をフックしてアクセス制御ルーチンを呼び出し，デバイスドライバを呼び出すかまたはエラー終了させる．これにより，VG をオンライン状態に保ったままアクセス制御が行える．また，制御グループ方式では複数の VG に対するアクセス制御を同一の判定ルーチンで一括して実施するため，VG 数に比例せず一定の処理時間で実施できる．この 2 方式を組み合わせることで，従来方式のオンライン化処理における VG 数に比例した I/O が発生する問題を解決し，VG 数に依存せず共有ディスク切替え時間を 2〜3 ミリ秒に抑止できる．

さらに，想定した環境においてシステム切替え時間を考察した結果，従来方式では 20〜30 秒程度要するのに対し，提案方式により十数秒程度に短縮できた．

最後に，本論文で提案した制御グループ方式と擬似

オフライン方式を組み合わせて適用したホットスタンバイシステムは，現在多くのオンラインシステムに導入され有効に稼動している．

参 考 文 献

1) Siewiorek, D.P.: Architecture of Fault-Tolerant Computers, *IEEE Computer*, Vol.17, No.8, pp.9–18 (1984).
2) Gray, J. and Siewiorek, D.P.: High-Availability Computer Systems, *IEEE Computer*, Vol.24, No.9, pp.39–48 (1991).
3) 当麻：コンピュータシステムの高信頼化技術入門，日本規格協会 (1988).
4) 当麻：フォールトトレラントコンピュータ，電子情報通信学会誌，Vol.70, No.1, pp.72–82 (1987).
5) 内藤ほか：高信頼 UNIX システム，マグロウヒル社 (1994).
6) 島田ほか：ホットスタンバイ UNIX システムの高信頼化，情報処理学会論文誌，Vol.34, No.5, pp.1010–1018 (1993).
7) 島田ほか：UNIX の高信頼化の一手法，電子情報通信学会論文誌（D-1），Vol.J76-D1, No.1, pp.31–35 (1993).
8) Chandra, T.D. and Toueg, S.: Unreliable Failure Detectors for Reliable Distributed Systems, *J. ACM*, Vol.43, No.2, pp.225–267 (1996).
9) Bratsberg, Humborstad: Online Scaling in a Highly Available Database, *Proc. 27th VLDB Conference*, Rome, Italy, pp.451–460 (2001).
10) Pedone, F. and Frolund, S.: Pronto: A Fast Failover Protocol for Off-the Shelf Commercial Databases, *Proc. 19th IEEE Symposium on Reliable Distributes Systems*, Nurnberg, Germany, pp.176–185 (2000).
11) Maya, Y. and Ohtsuji, A.: High-Availability Scheme Using Data Partitioning for Cluster Systems, *IEICE Trans. Inf. Syst.*, Vol.E82-D, No.11 pp.1457–1465 (1999).
12) Maya, Y., Isono, S. and Ohtsuji, A.: High-Availability Scheme for Shared Servers of Cluster Systems Using Commands Transfer, *IEICE Trans. Inf. Syst.*, Vol.E83-D, No.5 (2000).
13) 鶴保ほか：大規模機能分散型システムにおける高信頼化方式，電子情報通信学会論文誌（D-I），Vol.J73.D-I, No.2, pp.235–244 (1990).
14) J. グレイ，渡辺：フォールト・トレラント・システム，マグロウヒル社 (1986).
15) 渡辺ほか：トランザクション処理システム，平成 2 年電気・情報関連学会連合大会プログラム (1991).
16) 岸本ほか：プロセスの 2 重化による OS の高信頼化手法，情報処理学会論文誌，Vol.38, No.11, pp.2251–2261 (1997).
17) 山田ほか：エンドユーザーコンピューティングを実現するハードウェアの新ラインアップ，日立評論，Vol.75, No.9, pp.49–54 (1993).
18) 真矢ほか：分散システムにおける高速回復方式の提案，電子情報通信学会論文誌（D-I），Vol.J74.D-I, No.10, pp.729–738 (1991).
19) 真矢ほか：分散システムにおける端末無中断方式の提案，電子情報通信学会論文誌（D-I），Vol.J77-D-I, No.1, pp.86–93 (1994).
20) 原ほか：ネットビジネスを支えるミッションクリティカルシステムに対応したデータベース—HiRDB Version6，日立評論，Vol.84, No.9, pp.571–574 (2002).
21) 日本アイ・ビー・エム株式会社：AIX システム管理の基礎，アスキー社 (2002).

(平成 15 年 9 月 2 日受付)
(平成 16 年 9 月 3 日採録)

市川　正也（正会員）

1997 年九州大学工学部情報工学科卒業．1999 年九州大学大学院システム情報科学研究科情報工学専攻修士課程修了．同年（株）日立製作所入社．システム開発研究所に勤務．メインフレーム，UNIX サーバの研究開発に従事．

春名　高明

1988 年慶應義塾大学理工学部管理工学科卒業．1990 年同大学大学院理工学研究科管理工学専攻修士課程修了．1993 年同大学院理工学研究科計算機科学専攻博士課程単位取得退学．同年（株）日立製作所入社．現在，システム開発研究所に勤務．メインフレーム，UNIX サーバの研究開発に従事．

二瀬　健太（正会員）

1991 年電気通信大学電気通信学部計算機科学科卒業．1993 年同大学院電気通信学研究科情報工学専攻博士前期課程修了．同年（株）日立製作所入社．システム開発研究所に勤務．大形計算機，大形ディスク装置に関する研究開発に従事．

真矢　譲（正会員）

1980 年愛媛大学工学部電子工学科卒業．同年（株）日立製作所に入社．同社戸塚工場，1982 年よりシステム開発研究所，1999 年より情報コンピュータグループ，2001 年よりシステム開発研究所に勤務，現在に至る．電子交換機，フォールトトレラントコンピュータ，大形計算機，UNIX サーバ，NAS の研究開発に従事．博士（工学）．電子情報通信学会会員．

三瓶　英智

1991 年日本大学文理学部応用物理学科卒業．1993 年日本大学大学院理工学研究科物理学専攻修士課程修了．同年（株）日立製作所入社．ソフトウェア事業部に勤務．超並列計算機，UNIX サーバの研究開発に従事．