高信頼UNIX「風雅」のデータ安定格納機構*

6M－4

大橋　勝之†
(株)富士通研究所‡
ohashi@flab.fujitsu.co.jp

1　はじめに

我々はChorusマイクロカーネルとそのUNIXサブシステムをベースとした高信頼UNIX「風雅」の開発を行っている[1]．マイクロカーネル上のUNIXサブシステムは機能別にサーバとしてモジュール分割されている．風雅では，サブシステムを構成する各サーバを現用系と待機系から成るプロセスペアとして構成し，高い信頼性を実現している．

データ安定格納機構（Stable Area (SA)）は，プロセスペアの現用系と待機系の間で処理の引き継ぎに必要なデータを受け渡すための機構である．現用系は、待機系が再生できない最低限の引き継ぎデータだけをSAに保存し（エッセンス引き継ぎ），現用系が故障すると，待機系がSAからデータを読み出して処理を引き継ぐ（遅延引き継ぎ）．

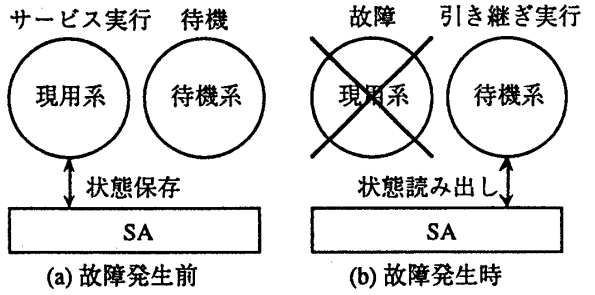

図1　SAの概要

本稿では、SAの機能要件とモデルについて論じ，実際に風雅で開発したSAの概要について報告する．

2　機能要件

SAの機能要件として以下の問題がある．

「データ引き継ぎの保証」

現用系がSAに保存したデータを待機系が確実に引き継げることを保証しなければならない．待機系が存在しない状態で保存されたデータも，待機系生成後に引き継ぎ可能となることが必要である．

「データ保存操作のアトミック性保証」

現用系がSAにデータを保存している最中に発生する故障に対処するために，SAのデータ保存操作のアトミック性を保証する必要がある．すなわち、SAへのデータ保存中に故障が発生すると，指定した全てのデータの保存が完了しているか，もしくは指定したデータは全くSAに保存されていないかのいずれかにならなければいけない．

「性能保証」

SAの使用によるサーバの性能低下が少ないことが必要である．実用的な処理性能を保証するために，エッセンス引き継ぎや遅延引き継ぎ等のSA使用方法の工夫だけでなく，メモリモデルに応じたSAの実装を行う．ただし，メモリモデルに依存しない統一的なSA I/Fを提供する必要がある．

3　メモリモデルによるSAの実装

SAの構成はNORMA/NUMAといったメモリアーキテクチャや，SA領域として使用するメモリの種別（揮発性/不揮発性）によって適切な実装法が異なる．風雅はノード故障に対応するため，マルチノードシステムを前提としており，UMAは検討の対象外とする．

3.1　NORMA型

現用系と待機系はそれぞれ個別にSA用メモリ領域を管理し，現用系が待機系へデータを転送することによって両者が管理するSAデータの整合を保つ．

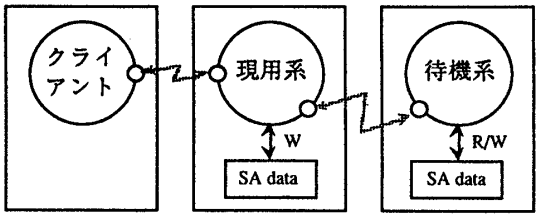

図2・NORMA型SA

3.2　NUMA/揮発性メモリ型

NUMAでは現用系と待機系でのメモリ領域の共有が可能だが，ハードウエア故障が発生すると揮発性メモリ上のSAデータにアクセス不可能となる恐れが

---

*The stable storage mechanism of the highly available UNIX "FUGA".
†Katsuyuki Ohashi
‡Fujitsu Laboratories Ltd.

あるため，両者が個別にSA用メモリ領域を保持する．現用系が待機系のSA用メモリ領域に直接，データを書き込むことによって両者の整合を保つ．

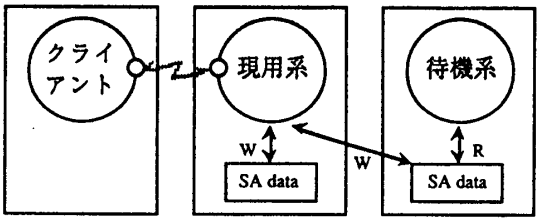

図3　NUMA/揮発性メモリ型SA

3.3 NUMA/不揮発性メモリ型

現用系と待機系間でSA用メモリ領域を共有する．不揮発性メモリでは，現用系あるいは待機系のハードウエアが故障しても，SAデータへのアクセスは保証されている．

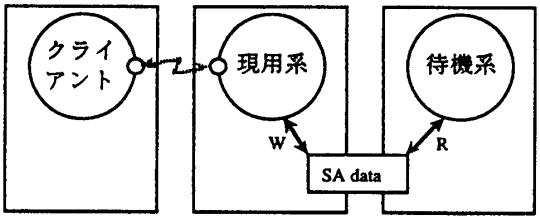

図4　NUMA/不揮発性メモリ型SA

4　データ同期処理

前節で述べたように，NORMA型の場合，現用系から待機系にデータ転送が発生する．現用系と待機系間の結合が低速な場合には，SAデータ更新の度にデータ同期を行う方法では処理のオーバーヘッドが非常に大きくなり，サーバの性能を大きく低下させる．そこで，両者間のデータ同期の実行を明に要求するためのI/F（sync）を提供し，サーバのコードが直接，データ同期の実行タイミングを制御することによってデータの転送回数を削減する．なお、データ転送を行わない場合にも同じI/Fを提供するが，サーバのリクエスト処理コードがsync を要求してもSAは何も処理を行わない．

5　インプリメント

我々はイーサネットによって接続されたPCクラスタ上で動作するNORMA型SA（風雅SA）を開発し，実際にファイル管理サブシステムをプロセスペア化して評価を行った[1]．

5.1　構成

風雅ＳＡはＳＡのペア管理を行うＳＡマネージャ（SAM）と，SAを利用するサーバへのライブラリであるSAライブラリ（SALIB）から成る．SALIBは現用系と待機系の両方にリンクされ，SAデータを保存するSAデータ領域と，SAデータ保存のアトミック性保証に使用するためのログ領域をメモリ上に持ち，SA機構を利用するためのI/Fを提供する．

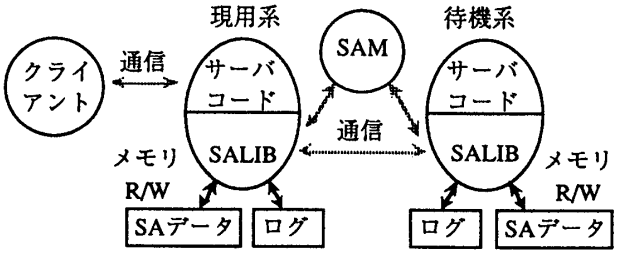

図5　風雅SAの構成

5.2　アトミックライトとデータ同期の処理

現用系のサーバコードはSALIBに対し，SAに設定するデータ集合を指定して，明にアトミックライトを要求する．SALIBは指定されたデータ集合をログ領域に記録した後，ログ領域のデータをSAデータ領域に展開する．このような処理によって，ログ領域への記録完了前の故障発生ではSA領域へのデータ保存は全く行われず，ログ領域への記録完了後の故障発生では指定されたデータ集合の全てがSA領域に保存される．

アトミック性を必要としないデータ保存（ノンアトミックライト）の場合には，SAは指定されたデータを直接，SAデータ領域に書き込むことでSAへのデータ保存処理を高速化する（ログ領域にはデータへのポインタを置く）．

前節で述べたデータ同期は，現用系のログ領域を待機系のログ領域に複写することによって実現する．なお，ノンアトミックライトされたデータはログ領域中のポインタではなく，SAデータ領域中のデータを複写する．

5.3　引き継ぎ処理

待機系はFault Manager (FTM)から現用系故障の通知を受けると，未展開ログをSAデータ領域へ展開した後，SAデータ領域から引き継ぎデータを読み出して引き継ぎ処理を行う．

参考文献

[1] 岸本他,"高信頼UNIX「風雅」",第52回情報処理学会全国大会論文集6M1-3,5-6, 1996