



# フォールトトレラント仮想マシンの実用的システム設計

The Design of a Practical System for Fault-Tolerant Virtual Machines

商用環境における実用的なVMフォールトトレランスの実現

● 性能オーバーヘッド10%未満

● 帯域幅要件20Mbps以下

● vSphere 4.0での実用実装

Daniel J. Scales, Mike Nelson, Ganesh Venkitachalam

VMware, Inc.

論文解説プレゼンテーション (コンピュータ専門家・研究者向け)

2025年7月26日

# 目次

フォールトトレラント  
仮想マシンシステムの  
技術詳細と実装



- 1 研究背景と課題
- 2 システム概要
- 3 技術的核心：決定論的リプレイ技術
- 4 重要技術コンポーネント
- 5 実験結果
- 6 考察と将来展望
- 7 結論と意義

# 研究背景と課題

## ▶ エンタープライズ環境におけるVMの可用性要求

ミッションクリティカルなアプリケーションは、ハードウェア障害発生時にもサービス継続が求められる

## ▶ 従来のHAソリューションの限界

- › VMの再起動を要する手法：数分間のダウンタイムが発生
- › 共有ストレージへの依存：単一障害点のリスク

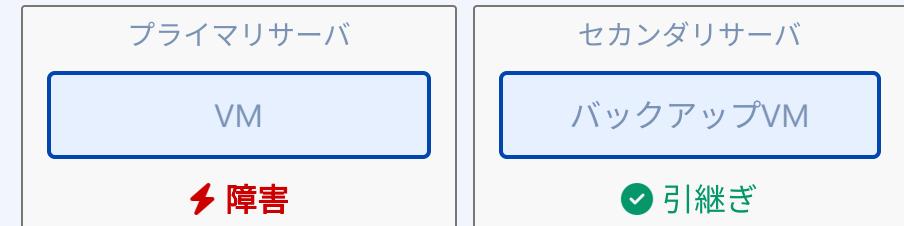
## ▶ 技術的課題

- › 透過的かつ瞬時のフェイルオーバーの実現
- › 非決定的イベントを含む完全な状態複製
- › 実用に耐えるパフォーマンスと帯域要件

## 研究目標

「実用的なシステムでは、性能オーバーヘッド10%未満、必要帯域幅20Mbit/s以下の実現が求められる。また、特殊なハードウェアを必要とせず、通常のx86サーバ上で動作できることが重要。」

## フォールトトレランスの課題



## リアルタイム同期

課題: 同期遅延、決定論的実行、オーバーヘッド

## 従来のアプローチの制約

- ロックステップ実行：特殊HW依存、高コスト
- 共有ディスク型HA：フェイルオーバー時間が長い
- アプリケーションレベル冗長：アプリ修正が必要

# システム概要：VMware FTのアプローチ

## ▶ プライマリ／バックアップアプローチ

プライマリVMの実行をセカンダリサーバ上のバックアップVMで完全に複製し、障害発生時に瞬時に切替

## ▶ VMware vSphere 4.0上での実現

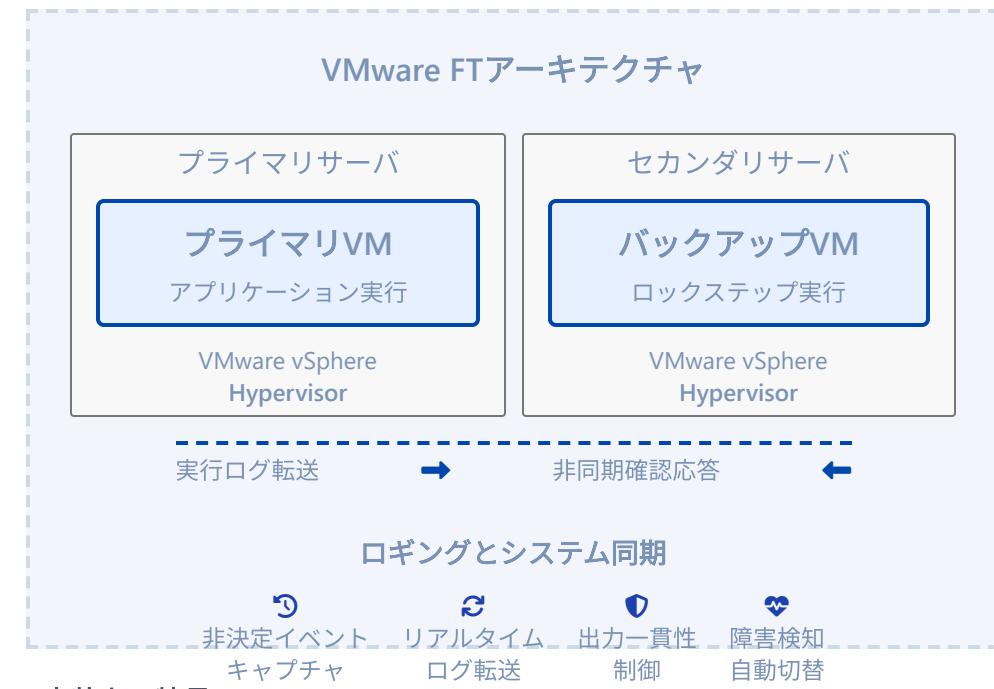
- 既存の仮想化プラットフォーム上に構築された商用実装
- 特別なハードウェア不要、一般的なx86サーバで動作

## ▶ 状態マシンアプローチの核心技術

- VMを決定論的なステートマシンとして扱い、入力と非決定イベントを制御
- Deterministic Replay技術を応用した実行ログの転送と再現
- Output Rule実装により一貫性を保った出力制御

### VMware FTの設計目標

「使いやすさ、低コスト、高パフォーマンスの三要素を両立する実用的なフォールトトレランスシステム。一般サーバ上で10%未満の性能低下と20Mbit/s以下の帯域でのロックステップ実行を実現。」



### 実装上の特長

- VMクラスタ管理と統合された自動化されたFT
- FT VMotionによる無停止メンテナンス対応
- 障害発生時の新バックアップVM自動生成
- 運用管理の容易さを重視した設計

# 技術的核心：決定論的リプレイ技術

## ▶ 決定論的リプレイの基本原理

仮想マシンを決定論的状態マシンとして扱い、同一の入力に対して常に同一の出力を生成することを保証

## ▶ VMの非決定性イベントの捕捉と複製

- › 入力：ネットワークパケット、ディスクリード、ユーザー入力
- › タイミング：仮想割り込み、デバイス制御タイミング
- › ハードウェア要因：CPU内部カウンター、DMA操作

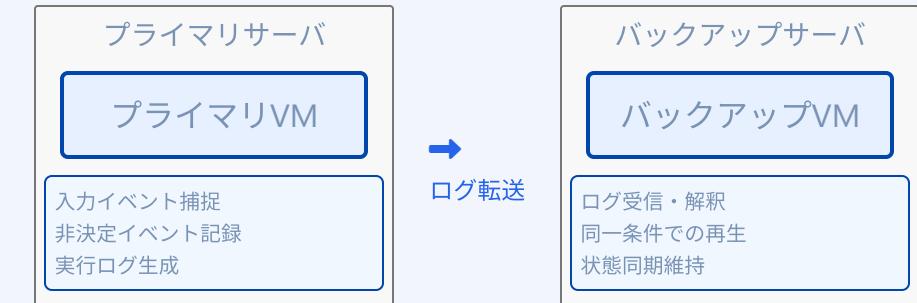
## ▶ ログベースの実装方式

- › プライマリVMでの実行ログ生成（非同期マルチスレッド）
- › UDP経由でのログ送信（低遅延、高効率）
- › バックアップVMでのリアルタイムログ再生

### VMware FTでの技術的革新

「既存の決定論的リプレイ技術をデバッグツールからフォールトトレランス用途に拡張。リアルタイム処理、低遅延同期、効率的ログ転送を実現し、冗長システムの商用実装に成功。」

## 決定論的リプレイの動作モデル



1. 非決定的イベント発生（割り込み等）

2. プライマリがイベント詳細をログ化

3. リアルタイムでバックアップへ転送

- ・ 帯域の最小化：特定イベントのみ記録、共有メモリ最適化
- ・ バッファの高効率転送：受信確認の最適バッチ処理
- ・ CPU効率：ハイパーバイザ処理の軽量化、マルチスレッド処理

# 重要コンポーネント①：ロギングとOutput Rule

## ▶ 決定論的リプレイのためのロギングメカニズム

- › 非決定的イベント（割り込み、タイマー、I/O）を全て記録
- › プライマリVMの実行ログをTCPで高速転送
- › バックアップVMでログを再生し同一状態を維持

## ▶ Output Rule：出力の一意性保証

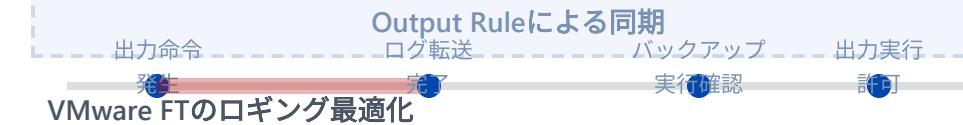
- › プライマリVMの出力（ディスク書き込み・ネットワーク送信等）はバックアップVMでの再現確認後に実行
- › ACKベースの同期：出力命令の実行前にログ転送完了を確認

## ▶ ログ帯域幅の最適化手法

- › ロギング量削減：本当に必要な非決定イベントのみを記録
- › ディスク読み取りの最適化：バックアップVMでも直接読み取り
- › 圧縮技術：類似イベントのバッチ処理と圧縮送信

## 技術的成果

「一般的なエンタープライズアプリケーションにおいて、ロギング帯域幅は20Mbit/s以下に抑えられた。これにより地理的に離れた環境（WAN経由）でも冗長構成が実現可能になった。」



- パフォーマンス評価：Oracle Swingbenchでは12Mbit/s、MS-SQL DVD Storeでは18Mbit/s程度のロギング帯域で動作
- バックアップ側でディスク読み取り実施時：Oracle Swingbenchでは12→3Mbit/sにロギング帯域が減少

# 重要コンポーネント②：障害検知と自動復旧

## ▶ 障害検知メカニズム

- › UDP/TCPベースのハートビート監視
- › ロギングネットワークトラフィックの監視
- › 管理ネットワークによる二重検証

## ▶ 瞬時フェイルオーバープロセス

- › 障害検知後、即座にバックアップVMがプライマリに昇格
- › IPアドレス・MACアドレスの自動引継ぎ
- › クライアントから透過的なフェイルオーバー（無停止）

## ▶ 冗長性の自動回復

- › vCenter Serverによる自動バックアップVM再生成
- › FT VMotionによるシームレスな冗長性再構築

## スプリットブレイン防止

「障害発生時のネットワーク分断状況でも、両方のVMが同時にプライマリとして動作する事態（スプリットブレイン）を防止する仕組みを実装。複数のネットワークパスを使った検証とロックメカニズムによる排他制御を実現。」

## VMware FTのフェイルオーバー処理

- 1 障害検知  
プライマリVMからのハートビート・ロギングデータが途絶
- 2 バックアップVMのアクティブ化  
バックアップVMがプライマリに昇格し、すべての通信を処理
- 3 ネットワーク接続の引継ぎ  
IP/MACアドレスの引継ぎ、ARPアナウンス、クライアント接続維持
- 4 新バックアップVMの自動生成  
vCenterが新しいホスト上で新バックアップVMを作成・同期



# 実験結果：性能評価

## ▶ 実験環境と評価対象

一般的なx86サーバで、各種ワークロードにおけるFT機能の性能への影響を測定

## ▶ 測定項目

- › FT有効時と無効時のスループット比較
- › ロギング帯域幅の測定

## ▶ 最適化手法の効果

- › バックアップ側でのディスクリード実行
- › ロギングのバッチ処理と最適化

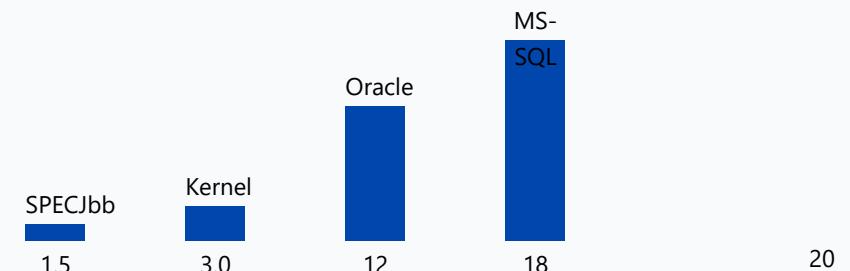
### 主要な発見

「全てのワークロードで性能低下は10%未満に抑えられ、ロギング帯域も20Mbit/s以下で実現。これにより、一般的な広域ネットワークでも地理的冗長構成が実用的に。」

## 代表的なワークロードでのパフォーマンス

ワークロード	性能比 (FT有効/無効)	ロギング帯域
SPECJbb2005	0.98 (2%減)	1.5 Mbit/s
Kernel Compile	0.95 (5%減)	3.0 Mbit/s
Oracle Swingbench	0.99 (1%減)	12 Mbit/s
MS-SQL DVD Store	0.94 (6%減)	18 Mbit/s

## ロギング帯域幅 (Mbit/s)



## 最適化の効果 (Oracle Swingbench)

標準設定：

バックアップでディスクリード：

12 Mbit/s  
3 Mbit/s (75%減)  
10

出典：The Design of a Practical System for Fault-Tolerant Virtual Machines, Table 1, Section 5.1

# 考察：技術的利点・制約・今後の課題

## ✚ 技術的利点

- ✓ 一般的なx86サーバ上での実装 (特殊ハードウェア不要)
- ✓ 低オーバーヘッド：性能低下10%未満
- ✓ 低帯域要件：通常20Mbit/s以下 (地理的冗長可能)
- ✓ アプリケーション透過性 (修正不要)

## ▲ 現在の制約

- › シングルプロセッサVMに限定されている
- › マルチプロセッサVM対応には決定論的実行の課題
- › ディスク書き込み性能への影響 (特にスループット要求高アプリ)

## ● 将来の研究課題

- › マルチプロセッサVMでの決定論的リプレイの効率化
- › 部分的なハードウェア障害 (メモリエラーなど)への対応
- › 低レイテンシシナリオでのさらなるオーバーヘッド最小化

## 今後の展望

「VMware FTの基本アーキテクチャと低帯域要件により、地理的に分散したデータセンター間での可用性確保や災害対策としての活用も十分に現実的である。」

参考資料・引用：The Design of a Practical System for Fault-Tolerant Virtual Machines (Scales, et al. VMware, Inc.)

## VMware FTと他のHA技術の比較

技術アプローチ	フェイルオーバー時間	アプリ透過性	ハードウェア要件
VMware FT	ゼロ (即時)	完全透過	標準x86
従来型VM HA	数分	透過	標準x86
ハードウェアFT	ゼロ	透過	特殊FTサーバ
アプリレベル冗長	数秒～分	非透過	標準x86

## 将来の展望：マルチプロセッサFT

### VMware FTの学術 産業的意義

- ・仮想化と耐障害技術の融合による新しいアプローチ
- ・決定論的リプレイ技術の実用化
- ・クラウド時代のミッションクリティカルシステムの新たな保護手法



地理的分散構成

20Mbit/s以下の低帯域で実現可能

# 結論と意義

## ▶ 主要な技術的貢献

VMware FTは、決定論的リプレイ技術に基づいた実用的な商用フォールトトレラントVMシステムを実現

- › 性能低下10%未満、帯域要件20Mbit/s以下を達成
- › 特殊ハードウェアを必要とせず、一般的なx86サーバで実現

## ▶ 学術的意義

- › ハイパーバイザレベルでの透過的なフォールトトレランスの実証
- › 非決定的イベント処理を含む完全な状態複製の実現手法
- › リアルタイムロギングとOutput Ruleによる整合性担保技術

## ▶ 産業的インパクト

- › エンタープライズ環境における無停止運用の実現
- › 低帯域での実現により地理的冗長化が可能に
- › クラウド時代のレジリエンスモデルへの応用

## 今後の展望

「マルチプロセッサVMへの拡張、部分的障害への対応強化、および地理的分散環境での最適化が次の研究ステップとなる。これらの技術がクラウド環境の可用性基盤として発展することが期待される。」

参考資料: The Design of a Practical System for Fault-Tolerant Virtual Machines (Scales, et al. VMware, Inc.) — 実用的なフォールトトレラント仮想マシンシステムの先駆的研究

## VMware FTの主要成果

- ✓ 実用的なオーバーヘッド  
10%未満の性能低下で運用可能

- 器 低帯域要件  
多くの用途で20Mbit/s以下

- 一般ハードウェア対応  
特殊機器不要、標準x86サーバで運用

学術的貢献  
技術の歴史的位置づけ

過去  
特殊HW依存  
高コスト

VMware FT  
汎用HW  
実用的性能

将来  
マルチプロセッサ  
地理分散