Reversibility for Efficient Computing pp249-255

可逆性は熱の削減以外に何に役立つのか

・エラーの検出

実行前に初期状態を保存しておき，実行後に逆実行を行って初期状態と比較する．その比較結果が異なる場合は，実行結果は信頼できないものと判断できる．プロセッサがハードウェアレベルで可逆性をもって設計されている場合はハードウェアエラーとなり，そうでない場合はハードウェアエラーかソフトウェアエラーとなる．

一時的なハードウェアエラーなら，初期状態と一致するまで計算を行うことで高い信頼性のある計算を行うことができる．ただ，高い信頼性の計算はすでに別の方法（同じ計算を複数実行して比較する）がある．

永続的なハードウェアエラーも検出可能である．理由は，永続的なハードウェアエラーはおそらく同じ個所が誤りになるためである．ただし，可逆性を保ってしまうようなハードウェアエラーは検出不能である．また，状態を比較する計算や，メインの計算に影響を与えるようなハードウェアエラーは必ずしも検出可能なわけではない．

ソフトウェアエラーなら，対応する逆実行に一致しない順実行のエラーを検出可能である．ただし，それ以外のソフトウェアエラーは検出不能である．

・情報の保護

クラッカーが干渉不能なレベルで可逆性をもつ場合，可逆性の性質から情報の完全な破壊ができない．また，可逆性によりクラッカーの操作も残るため，クラッカーの操作を逆実行することで情報を復元することができる．

ただし，システムへの攻撃そのものを防げるわけではない．クラッカーが行った操作を逆実行するまで，システムは異常なままである．

別の方法に，初期状態とすべての操作ログを取る方法があり，可逆性を利用する方法はそれに比べて優れていると言えるわけではないため，これは可逆性の有用性を示しているとはあまり言えない．

・プログラムデバッグ

可逆性をもつアプリケーションは，双方向デバッガを簡単に作成できる．双方向デバッガは，バグが起こった部分から実行をさかのぼって，本当にバグが起きた瞬間を探せるため，有用性がある．

しかし，双方向デバッガの実装には，厳密には可逆計算は必要ない．可逆計算を用いない双方向デバッガの簡単な手法としては，チェックポイントを保存しておき，実行を逆に辿る場合は，チェックポイントからの順実行で目的の場所に辿り着く手法がある．そのうえ，場合によっては，プログラミング環境を再インストールした方が簡単なこともある．

・トランザクション処理とデータベースのロールバック

データベースのロールバックは，可逆コンピューターの相互通信を行う可逆プロセスの，より一般的なフレームワークの上に実装可能である．

ただし，データベースのロールバックは現在十分実用的な手法がある．

・マルチプロセッサでの投機的実行

マルチプロセッサでの投機的実行は楽観的に計算を行うため，情報の不整合が検出された場合は情報が正しかった場所から計算を行わなければならない．可逆計算を用いれば，逆実行により実行をさかのぼることができる．

・物理シミュレーションの数値安定性

可逆性は，現実の世界でも維持される法則であるため，シミュレーションでも維持される必要がある．実際に非可逆なシミュレーションでは，誤りがどんどん大きくなっていくことがあり，可逆なシミュレーションでは誤りは一定の大きさに保たれた．

この利点を得るために，物理層などの低レベルな部分が可逆である必要はない．