令和 元年度 卒業論文概要			
所　　属	機械情報工学科	指導教員	光來 健一
学生番号	16237027	学生氏名	木村 健人
論文題目	GPU からのメモリ書き換えによるプロセスレベル障害からの復旧		

1　はじめに

近年，コンピュータシステムの複雑化に伴い，システム障害を避けるのが難しくなっている．システム障害が発生するとシステム上で動作しているサービスの品質が低下したり，完全に停止したりするため，サービスの利用者や提供者は大きな損失を被る．そのため，障害発生時には迅速かつ正確に障害を検知し，システムの復旧を行う必要がある．信頼性の高い障害検知を行うためのシステムとして，監視対象ホストの GPU 上で監視を行う GPUSentinel[1] が提案されている．しかし，GPUSentinel は障害を検知した後の復旧には対応していない．そのため，障害の発生したシステムが応答しなくなっている場合，ハードウェアリセットを行うことで復旧するしかない．強制的にシステムのリセットを行うとデータが失われる可能性があり，障害発生直前の状態に復元するためにコストがかかる．

本研究では，できる限りシステムのリセットを避けるため，GPU からメインメモリ上の OS データを書き換えることでシステム障害からの復旧を行うシステム GPUfas を提案する．

2　システム障害の検知と復旧

コンピュータシステムの障害件数は年々，増加する傾向にある．これは，システムが複雑化・多様化・大規模化しているため，システム障害を引き起こしうる要因を事前に発見して対策を行うことが困難になってきているためである．システム障害が発生するとシステム上のすべてのサービスの品質が低下したり完全に停止したりするため，サービスの利用者と提供者の双方にとって経済的に大きな損失になる．

そのため，障害発生時には迅速かつ正確に障害を検知して復旧する必要がある．信頼性の高い障害検知システムとして GPU を用いた GPUSentinel[1] が提案されている．図 1 に GPUSentinel のシステム構成を示す．GPUSentinel では GPU 内で OS 監視システムが自律的に動作し，メインメモリ上の OS データを分析することで監視対象システムの障害を検知する．GPU を用いる利点として高信頼・高性能・汎用性

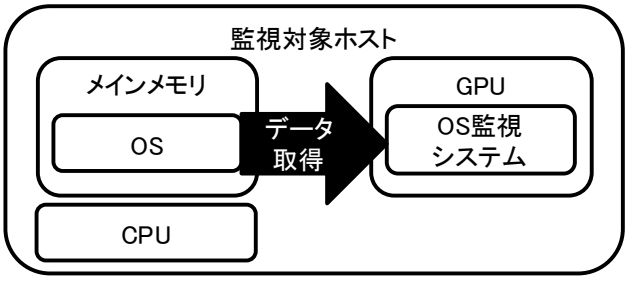

図 1　GPUSentinel のシステム構成

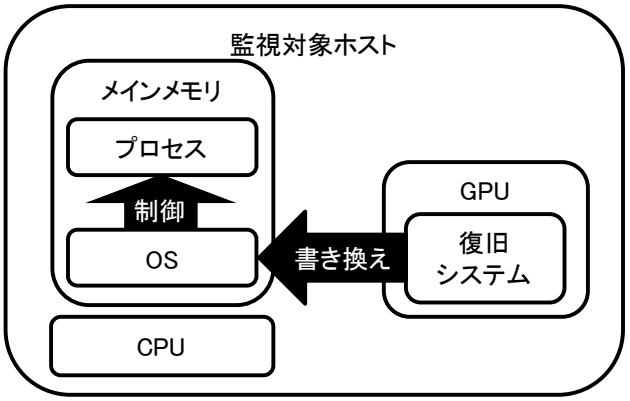

図 2　GPUfas による障害からの復旧

が挙げられる．GPU は OS が動作する CPU やメインメモリから物理的に隔離されており，ソフトウェア障害による影響を受けにくいため，障害検知の信頼性は高い．また，GPU は多数の演算コアを有しており，それらを用いることで複数の障害検知を並列に実行したり，それぞれの検知処理を並列化したりすることができる．加えて，GPU は多くの計算機に標準的に搭載されている汎用ハードウェアであり，調達が容易である．

しかし，GPUSentinel は障害を検知した後の復旧には対応していない．そのため，障害が発生したシステムの復旧を遠隔から行うには，OS の機能を使って対象システムにリモートログインして復旧作業を行う必要がある．その際に，システムが応答しなくなっている場合にはハードウェアリセットを行うしかないことが多い．例えば，IPMI 等のリモートコンソールを用いたり，リモート電源管理装置を用いたりしてシステムのリセットを行う．このように強制的にシステムをリセットするとデータが失われる可能性があり，障害発生前の状態に戻すにはコストがかかる．

3　GPUfas

本研究では，障害検知時にできる限りシステムリセットを避けるために，GPU からメインメモリ上の OS データを書き換えることでシステム障害からの復旧を行うシステム GPUfas を提案する．GPUfas は GPU において障害を検知した後で障害原因を特定し，可能であればそれを自動的に取り除くことで復旧を行う．さらに，GRASS[2] を用いることで，障害情報を受け取った管理者がインタラクティブに復旧を行うことも可能である．GRASS はリモートホストが OS を介さずに GPU と直接通信を行うことを可能にするシステムである．

GPUfas はプロセスレベルの障害からの復旧を行うために，図 2 のように，障害の原因となったプロセスを一時停止させたり，終了させたりする手法を用いる．また，プロセスが利用

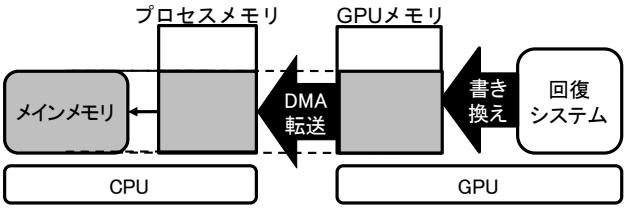

図 3　GPU からメインメモリのデータ書き換え

できる CPU やメモリなどの上限を制限したり，プロセススケジューリングを変更したりすることでも復旧を行うことができると考えられる．現在のところ，プロセスを停止・終了することにより，CPU 高負荷とメモリ不足による障害から復旧が可能になっている．

3.1　シグナルの疑似送信

多くの OS ではシグナル機構を用いることでプロセスの制御を行うことができる．例として，CPU 負荷が異常に高いプロセスを一時停止することでシステム全体の CPU 負荷を下げて，システム全体が応答する状態にすることができる．また，メモリを大量に消費しているプロセスを終了することでメモリを解放させ，メモリ不足によるサービスの性能低下や異常終了を防ぐことができる．しかし，シグナルは OS の機能を用いてプロセスに送信することしかできず，GPU からプロセスに直接シグナルを送信することはできない．

そのため，GPUfas はメインメモリ上にあるプロセス情報を書き換えることで，GPU からプロセスへのシグナルの疑似送信を実現する．疑似送信はプロセスの状態をシグナルが送られた後の状態に変更する手法である．まず，シグナルビットマップに対して送信するシグナルの種類に対応するビットをセットし，さらに未処理フラグもセットする．これにより，OS の機能を用いてシグナルを送信した場合と同様に，プロセスがカーネルモードからユーザモードへ遷移するタイミングでシグナルの処理を行わせることができる．

3.2　GPU からの OS データ書き換え

GPUfas は GPU から OS データの書き換えを行うために，CUDA のマップトメモリ機構を用いる．まず，メインメモリ全体をプロセスメモリにマッピングし，プロセスメモリを介して GPU メモリにマッピングする．GPU メモリにマッピングされた OS データの書き換えを行う際には，まず，GPU がデータを自動的にメインメモリから GPU メモリに DMA 転送する．そして，GPU 上の回復システムがそのデータを書き換えると，図 3 のように GPU が自動的にそのデータをメインメモリに DMA 転送し，メインメモリ上の OS データに反映する．

OS データの書き換えを行う際には，その仮想アドレスをメインメモリの物理アドレスに変換してから GPU メモリのアドレスに変換する必要がある．そこで，LLView[1] を拡張することでこのアドレス変換を自動化した．LLView を用いて復旧システムのコンパイルを行う際に，メモリ書き込み命令である store 命令の前にアドレス変換処理を挿入する．

4　実験

GPUfas を用いてプロセスレベルの障害発生後に迅速に復旧できることを調べる実験を行なった．比較として，監視対象

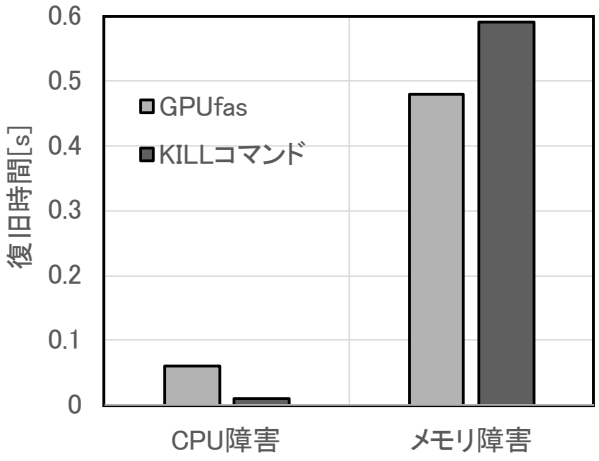

図 4　復旧時間

システム内で KILL コマンドを用いてプロセスにシグナルを送信した．実験には，Core i7-8700 の CPU，16GB のメモリ，NVIDIA GeForce GTX 960 の GPU を搭載した PC を用いた．OS として Linux 4.18.0 を動作させ，CUDA 10.0.130 および NVIDIA GPU ドライバ 430.40 を用いた．

まず，CPU を高負荷にする障害を発生させるために CPU 数と同じ数の yes コマンドを実行した．復旧にかかった時間を図 4 左に示す．GPUfas では 0.06 秒で yes コマンドを実行しているすべてのプロセスを停止させることができた．KILL コマンドを用いた場合の復旧時間は 0.01 秒であった．KILL コマンドの方が復旧時間が短い理由は，シグナルを送信した時にプロセスが即座にスケジューリングされてシグナルが処理されたためである．GPUfas ではプロセスが次にスケジューリングされるまでシグナルの処理が待たされた．

次に，メモリ不足の障害を発生させるために，大量のメモリを確保するプログラムを実行した．図 4 右に示すように，GPUfas は 0.48 秒でプロセスを終了させることができた．KILL コマンドを用いた場合の復旧時間は 0.59 秒であった．GPUfas のほうが復旧時間が短い理由は，メモリ不足の対象システム内で KILL コマンドを実行するのに時間がかかったためと考えられる．

5　まとめ

本研究では，GPU を用いてシステム障害の検知後にメインメモリ上の OS データを書き換えることで復旧を行うシステム GPUfas を提案した．GPUfas は OS にシグナルを疑似的に送信することでプロセスレベルの障害からの復旧を可能にする．今後の課題は，GPU からプロセスのスケジューリングを変更することによる復旧，およびカーネルレベルの障害からの復旧である．

参考文献

[1] Y. Ozaki, S. Kanamoto, H. Yamamoto, and K. Kourai. Detecting System Failures with GPUs and LLVM. APSys 2019.

[2] 金本颯将, 光来健一. GPUDirect RDMA を用いた高信頼な障害検知機構. ComSys 2019.