

gUSA

***Simple and Efficient User Space Atomicity
Emulation with Little Kernel Modification***

新部 裕

独立行政法人 産業技術総合研究所 (AIST)

2002 年 9 月 19 日

もくじ

- gUSA とは?
- 原理
- 実装 (SuperH, MIPS R5900, ARM)
- 評価
- 考察

gUSA とは

gUSA は単一プロセッサシステムのユーザ空間の排他制御の機構である。

- 単一プロセッサのシステムであっても、プログラミングモデルとしてのスレッドは有用。
- しかし、これまでよい方式がなかった。
- そこで、新しい “g” User Space Atomicity.

General : 排他制御のいろいろなルーチンの実装に対して利用できる, 一般性の高い仕組みである。

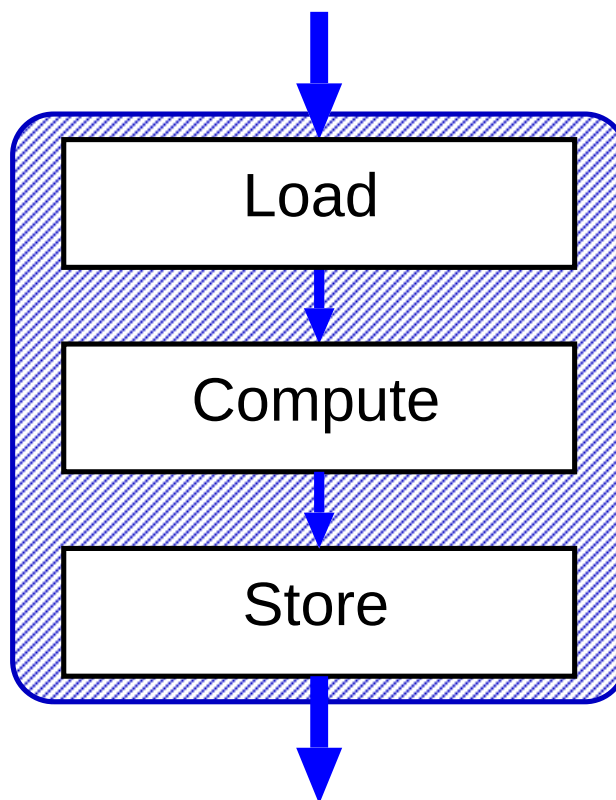
Generic : 特殊な機能を用いず広範囲のプロセッサに対応可能である。

Gentle : カーネルの変更は最小限であり, 適用が容易である。
また, 既存の機能に与える影響が少ない。

gUSA の原理 (1)

排他制御のためには以下の条件を満たせばよい:

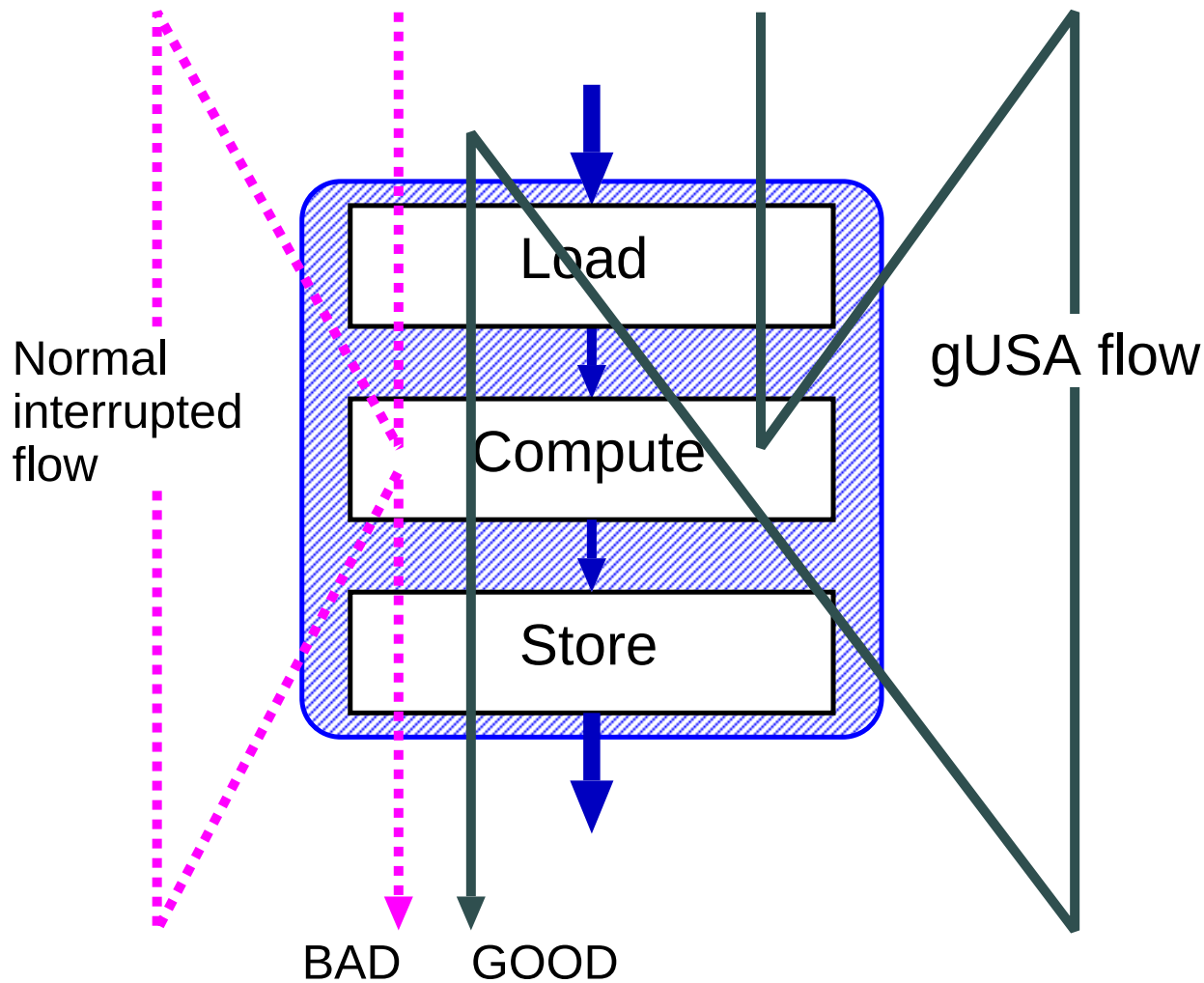
- **Store** で書き込まれるデータは, 分断されることなく **3 段階** で処理されたデータである。



gUSA の原理 (2)

それには:

- 割り込まれたら最初からやり直し (rewind)。



gUSA の原理 (3)

「最初からやり直し」をカーネルで実現。ユーザ空間に関して:

- 排他領域の弁別
- 戻り先の同定

⇒ スタックポインタを使う。

カーネルの変更は, 2 点だけでよい (UNIX):

- シグナルハンドラに制御が移る場合
- **preemption** で他のスレッドに制御が移る場合

⇒ 最小限の軽い変更

gUSA の実装 (1)

GNU/Linux システムにおいて実装した。以下の 3 種類のプロセッサ。

- SuperH (SH-3, SH-4) $\Rightarrow$ Dreamcast
 - MIPS 5900 $\Rightarrow$ Playstation 2
 - ARM (StrongARM) $\Rightarrow$ Zaurus
-
- スタックポインタが(符号付き整数として)負の値を取ることによって排他領域を示す。
 - 戻り先は 2 種類 (シグナル, preemption)

gUSA の実装 (2)

例: SuperH での実装

- ハードウェアでは TAS (test and set) 命令のみのサポート。

排他領域の定義 (ABI 拡張):

- スタックポインタ **r15** は (符号付き整数として) 負の値を取り, 絶対値が排他領域の大きさを示す。
- レジスタ **r0** は排他領域の出口を指す。
- レジスタ **r1** に元のスタックポインタ **r15** の値が退避される。

gUSA の実装 (3)

例: SuperH での実装

- TAS (test and set) 命令による spin lock より有用。

カーネルの変更:

- arch/sh/entry.S で実行権限を他に渡すところ (reschedule)
- arch/sh/signal.c でシグナルハンドラに制御がいくところ (handle_signal)

gUSA の評価 (1)

カーネルの変更の影響 ⇒ 無し

- コンテキストスイッチ, シグナルハンドリングで影響があると考えられた。
- SH-3, SH-4 のシステム (Debian) で Imbench-2.0-patch2-2 で比較。
- カーネルの性能劣化は見られなかった。
- 重い処理に最小限の追加だけなので, 影響がほとんどない。

gUSA の評価 (2)

ユーザ空間での利用の性能 $\Rightarrow$ 高性能

- 簡潔で明快な実装となるため, 高い性能をあげることが期待できた。
- アトミックな加算を構築する `eea.c` というプログラムで評価。
- オブジェクトの数 O , 一つのオブジェクトを共有するスレッドの数 T がパラメータ。
- gUSA, test and set によるスピンロックを用いたもの (単一, 分散), システムコールの方式を経過時間で比較。
- 全ての場合について gUSA がもっとも高性能である。

考察

- ABI の拡張の影響
- スタックポインタの初期値
- システムセキュリティ (誤用, 悪用)
- スレッドライブラリの実装の種類
- アトミックな Store
- PREEMPTIVE カーネル

おわりに

これから:

- カーネル 2.4 に実装した。メインストリームへ。
- カーネル 2.5 と GNU C library の適用。
- gUSA によりスレッドの利用が開かれる。

謝辞:

- 着想と設計: 小島 一元さん, 田中 哲さん
- 実装: Greg Banks さん, Ulrich Drepper さん, 戸村 哲さん
- 評価: 八重樫 剛史さん, 後藤 正徳さん, 野首 貴嗣さん

Happy Hacking.