Linuxにおけるリアルタイムタスクの扱い方、および、関連技術の動向

松原 克弥
株式会社イーゲル

CELF booth, ET2007
2007.11.14-16

背景

igel

Linuxはリアルタイムではない=組み込みに不向き？？

システム全体のタスクプライオリティ設計をきちんとやったのに、思うようにタスクが動かないなぁ。

RTタスクを使っていると、システムがしばらくフリーズすることがある。なぜだろう…

システム規模が大きくなるにつれて、加速度的に性能が悪くなる（遅延が大きくなる）なぁ…

良く聞くトラブル

今日の目的

igel

Linuxでリアルタイム性が必要なシステム（組み込み系システム）が構築できないと思われている誤解を解きたい。

- なぜそう思われるか？
- それは誤解か？
- どの程度のリアルタイム性に対応できるのか。

Linuxに対して尻込みしてしまうのはなぜか？

igel

その最大要因のひとつは
「スケジューラ」です。

- Linuxには、高機能なスケジューラが備わっている。スケジューリングポリシーに基づき、最適なタスク実行（切り替え）を遂行する。＝**綿密な**タスクプライオリティ設計の**必要がない**。
- でも、アプリケーションから、そのポリシーを「完全に」コントロールすることは**できない**。

リアルタイム性に関して、Linuxに不安を感じている人への提案

igel

1. Linuxスケジューラの特徴をきちんと理解する。
   - 「（組み込み）システム設計者が
     スケジューラについて知っておきたいこと」
2. Linuxにおけるリアルタイム性確保のための選択肢を知る。
   - 「Linux communityが提供する
     リアルタイム性向上の選択肢」

きちん理解をして、開発すれば…

- 前述のようなトラブルを回避・解決できる。
- 効率的な開発、最適な性能を引き出せる。
- Linux上でリアルタイム性が求められるシステムの構築できる（構築の実現可否を判断できる）。

igel

（組み込み）システム設計者が
スケジューラについて知っておきたいこと

スケジューラについて知っておきたいこと

igel

1. スケジューリングの契機
   - いつタスクがプリエンプト（一時停止）される（できる）のかを知る。
2. スケジューリングにおけるパラメータ
   - 何の要因で次のタスクが決定されているのかを知る。
3. スケジューラのオーバヘッド
   - タスクの数がシステム性能に与える影響を知る。
4. スケジューラに関する最新動向
   - 改良された箇所を知る。

1. スケジューリングの契機

igel

スケジューラとは、
現在動作中のタスクを停止（プリエンプト）して、次に実行すべきタスクを決定し、そのタスクにCPUを割り当てる（ディスパッチ）する役割をもつ機能モジュールである。

スケジューリングの契機

- 割り込み処理（tickタイマを含む）からの復帰時
- システムコールからの復帰時
- タスクが自主的に休眠した時

⇒リターン時に別タスクのスタックポインタをセットし、pop

- ただし、2.4 / CONFIG_PREEMPTなしの2.6では、カーネルモードではプリエンプション（タスク切り替え）が起きない（割り込み処理は起きる）。
  - カーネルモード時の割り込みからの復帰は元の場所に戻る（プリエンプション非対応コードのため）

2. スケジューリングにおけるパラメータ

igel

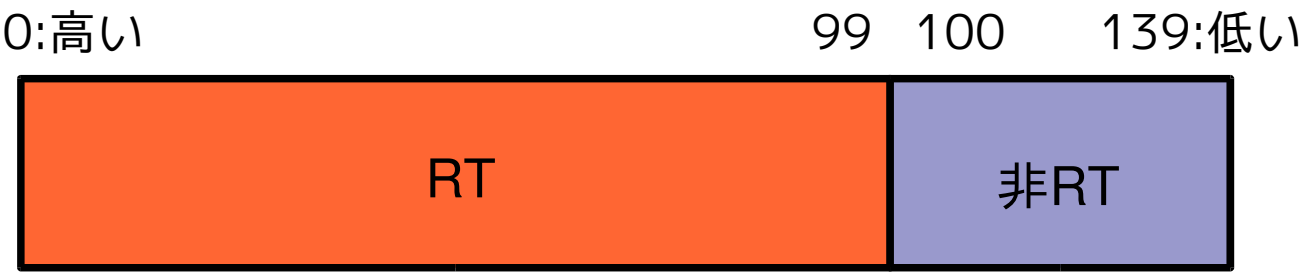
次のタスクを決めるためのパラメータ

- 優先度
- タイムスライス
- タスク種別

優先度:RTとNon-RT

igel

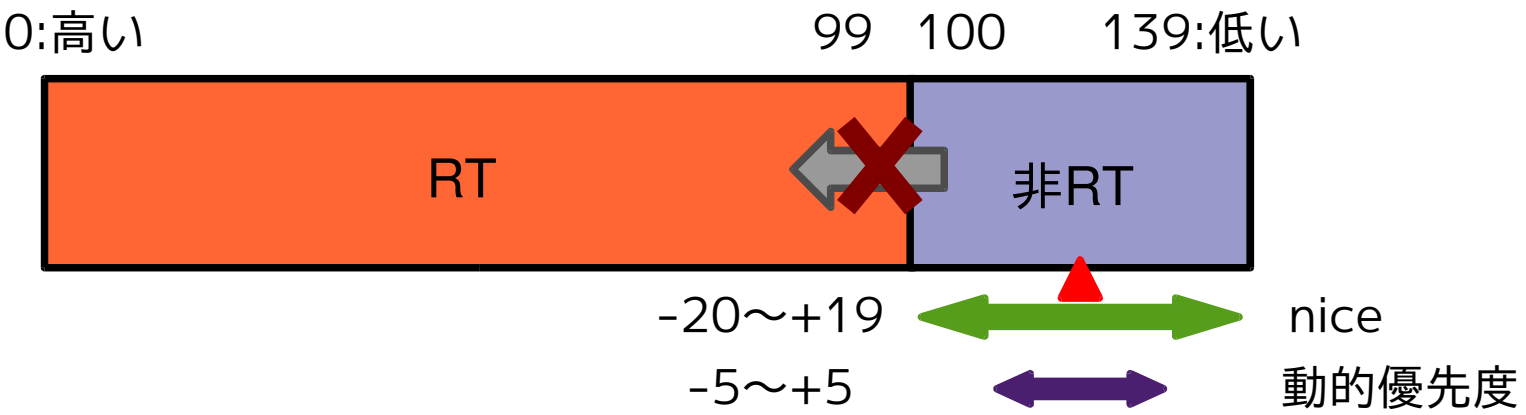
- RTタスクの優先度:0～99
- Non-RTタスクの優先度:100～139
- RTタスクは、いつでもnon-RTタスクをプリエンプト可能
- RTタスクが実行中、他のタスク(non-RTタスク)は動けない。



優先度:niceと動的優先度

igel

- nice：静的優先度の調整
  - -20から19の範囲
  - タスク起動時のデフォルトは0
  - nice()システムコールで変更
- 動的優先度
  - 加減算することで、I/O型タスクをrewardして、CPU型タスクをpunishするための優先度
  - 加減算する値＝動的優先度：+5から-5まで



タイムスライス

igel

非RTタスク（と一部のRTタスク）に割り当てられる最大CPU占有時間

- タイムスライスを使いきると、タイムスラスが残っている他のタスクがいる限り、スケジュールされない。
- （単純に）静的優先度に比例して計算され、高い静的優先度には大きいタイムスライスが与えられる。

タスク種別

igel

（ユーザが指定するのではなく）スケジューラにより判定される。

- 実行時間とスリープ時間の割合から推測

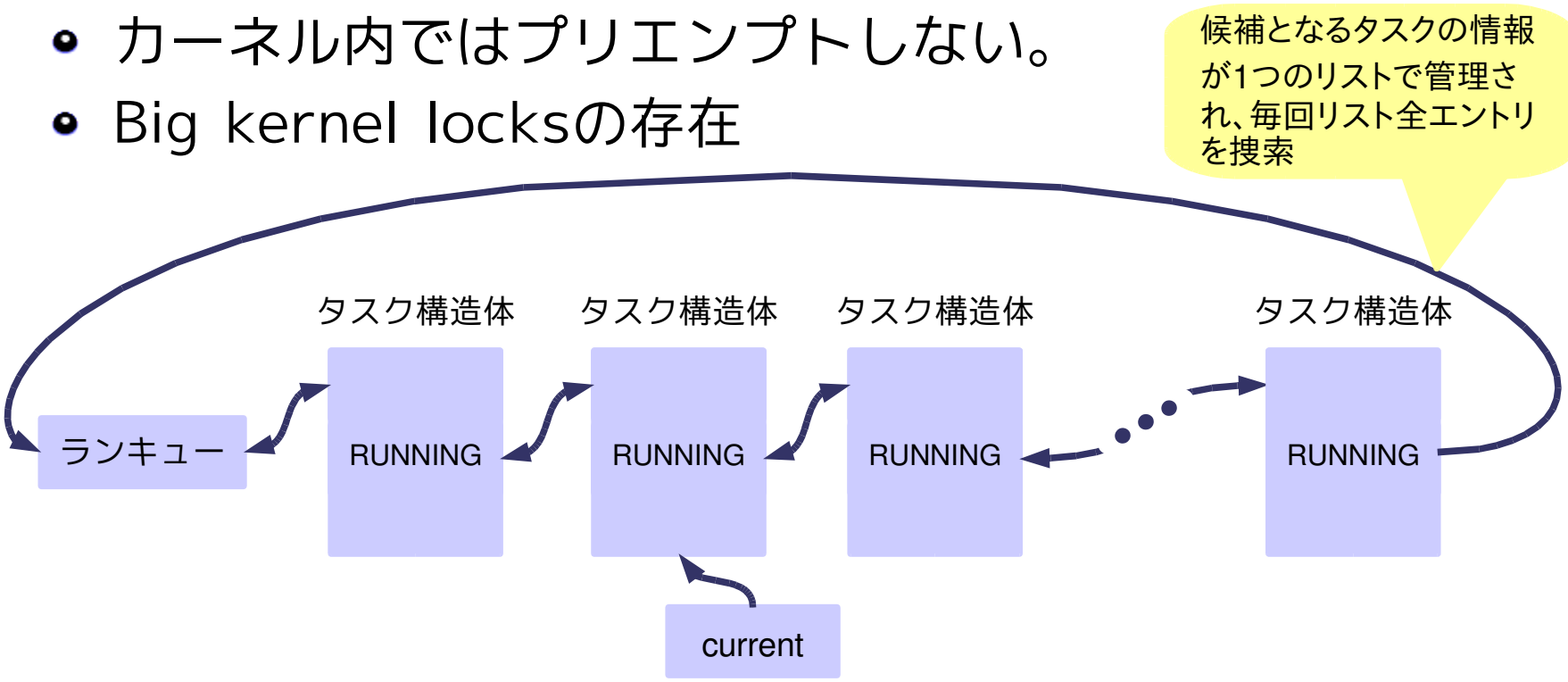
- CPU型
  - タイムスライスや動的優先度によるペナルティ
- I/O型
  - 動的優先度による優遇
- 対話性
  - タイムスライスの大きさや最割り当てによる優遇

3. スケジューラのオーバヘッド

igel

カーネル2.4の場合

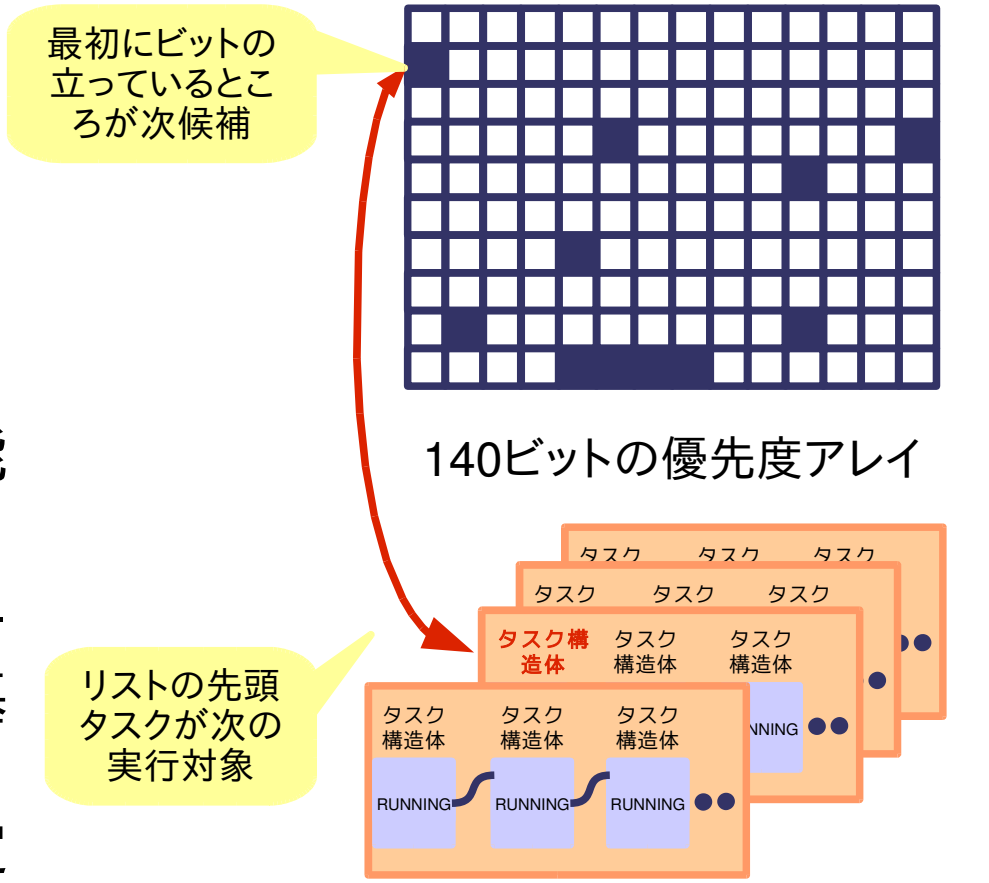
- 構造がシンプル＝オーバヘッドが小さい
- O(n)スケジューラ
- カーネル内ではプリエンプトしない。
- Big kernel locksの存在



スケジューラのオーバヘッド (contd.)

カーネル2.6の場合

- O(1)スケジューラ
- CONFIG_PREEMPT: カーネル内でのプリエンプトを可能に
- 「実行時間とスリープ時間の割合」に基づく動的優先度とタイムスライスの決定



注）本発表内容は、2.6.8における仕様に基づく

4. スケジューラの最新動向

igel

- linux-2.6.23での変更
  - スケジューリングポリシのモジュール化
  - CFS(Complete Fair Scheduler)
    - タスク種別による微調整をやめ、公平性を追求
    - タスクグループに対するスケジューリング(2.6.24)

igel

Linux communityが提供する
リアルタイム性向上のための選択肢

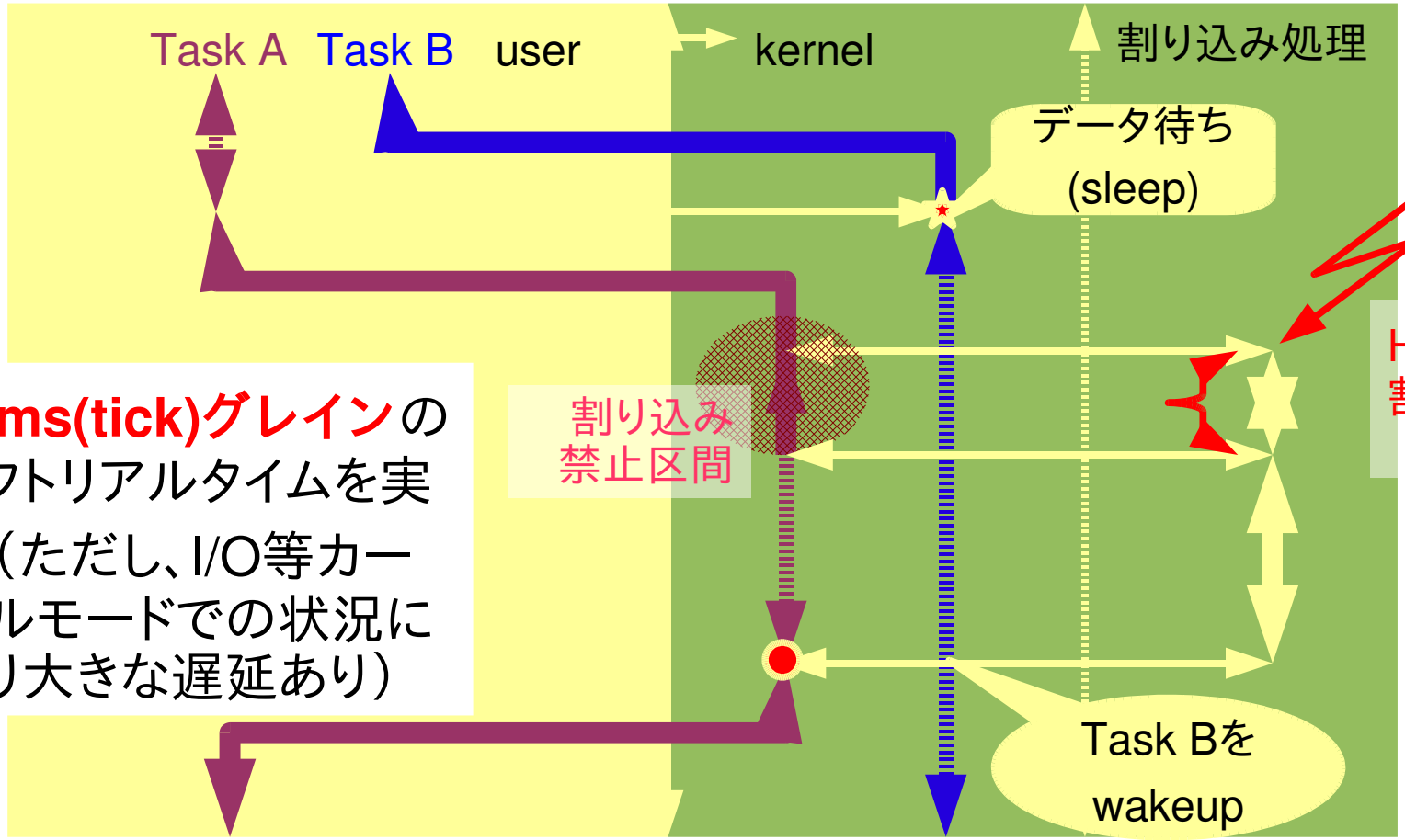
'CONFIG_PREEMPT (CONFIG_PREEMPT_DESKTOP)'

igel

リアルタイム性向上へのアプローチ

- プリエンプションの機会を増やす。
- スケジューラ動作の回数が増える。
- プライオリティに応じてタスク切り替えが起きる確率が増える。
- 動くべきタスクが実際にCPUにディスパッチされるまでの遅延が減る。
- **リアルタイム性が向上する！**

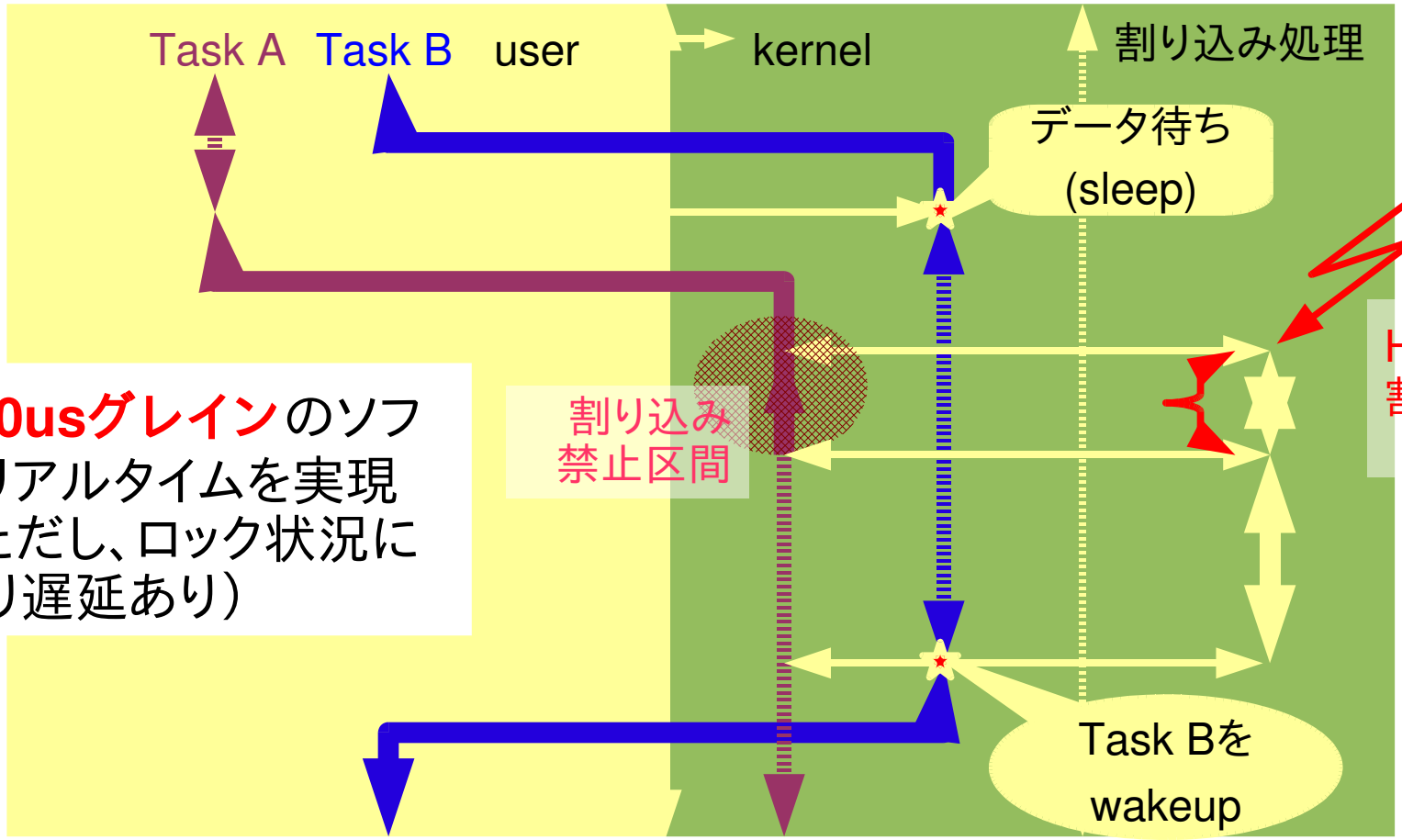
CONFIG_PREEMPT not set



10ms(tick)グレインの
ソフトリアルタイムを実現（ただし、I/O等カーネルモードでの状況により大きな遅延あり）

CONFIG_PREEMPT=y

igel



100usグレインのソフトリアルタイムを実現（ただし、ロック状況により遅延あり）

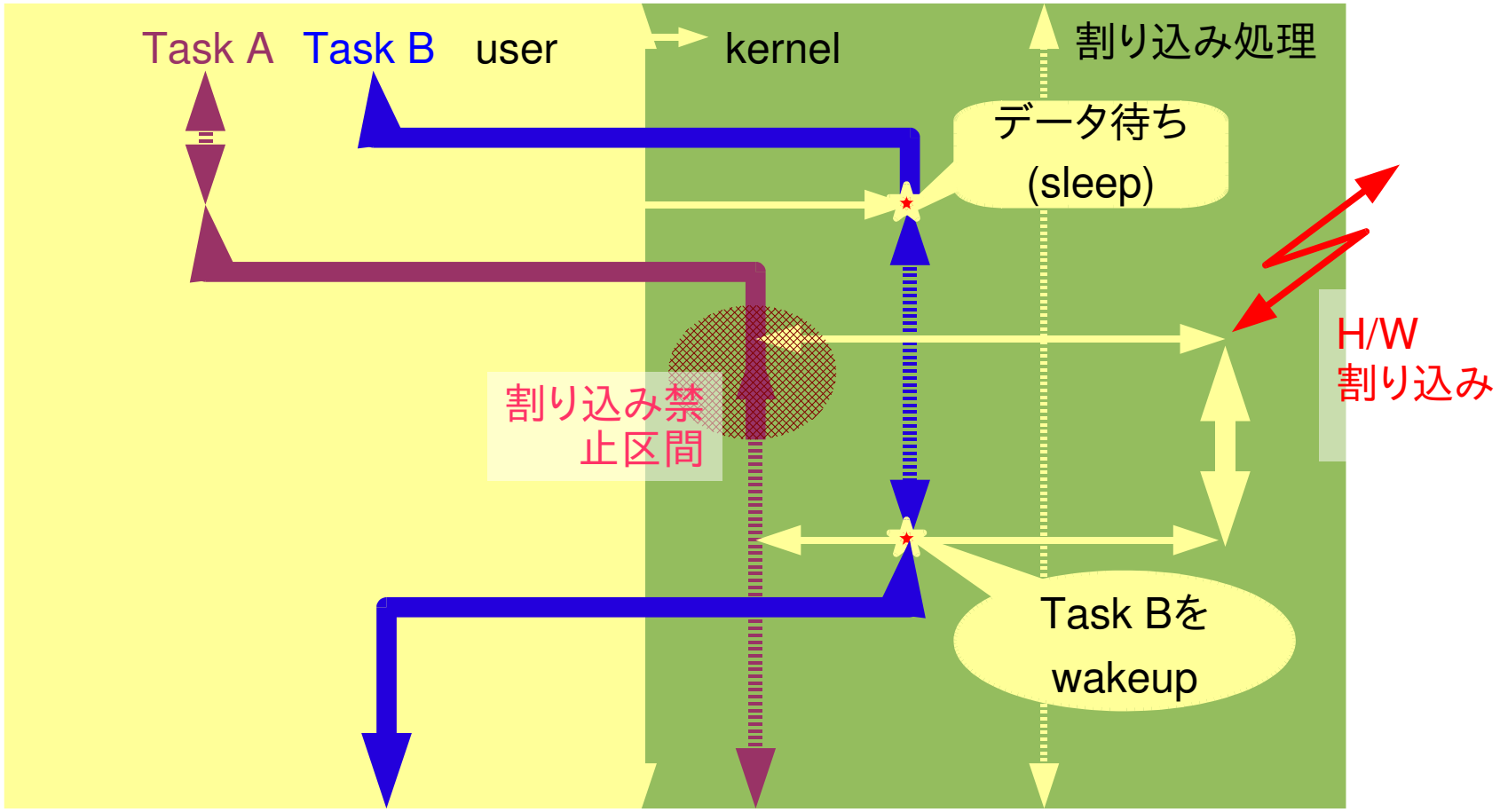
CONFIG_PREEMPT_RT

igel

- Ingo MolnarらのLinux realtime preemptionパッチ
- プリエンプションの機会を、割込み処理中やクリティカル領域内にも増やすことで遅延を軽減し、タスク起床および割り込み処理において、20～30usのソフトリアルタイムを実現
- 優先度逆転を防ぐため、優先度継承ロックを提供
- 他コード部分のリファクタリング・最適化

CONFIG_PREEMPT_RT=y

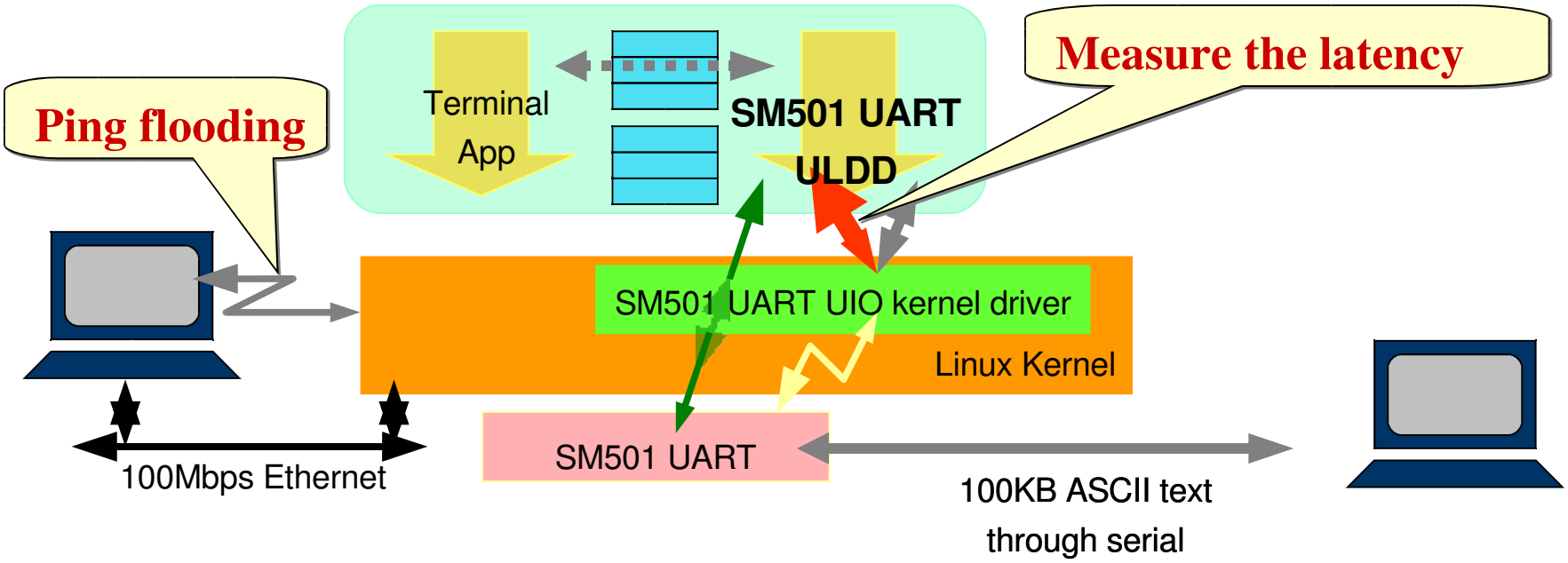
igel



PREEMPT_RTの評価：タスク起床遅延の測定

igel

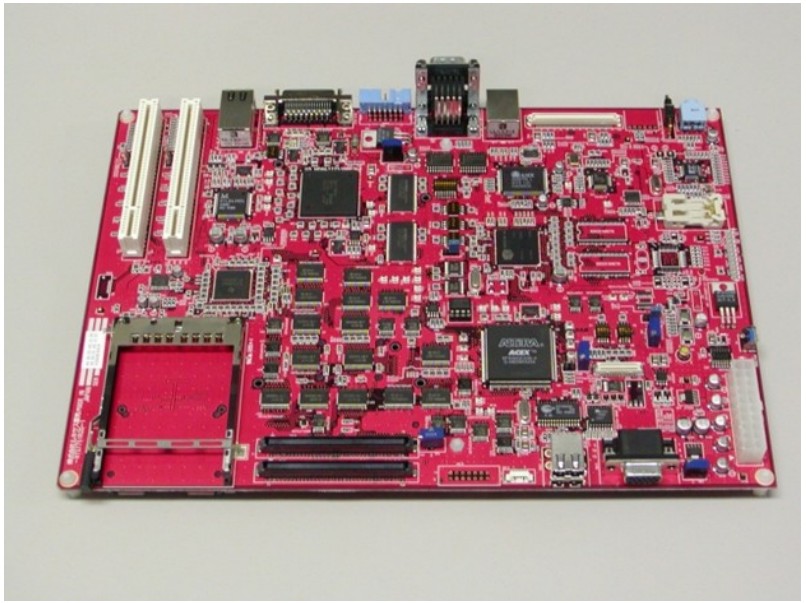
- Implement a SM501 UART driver with UIO
- Measure the schedule latency (when kernel handler invokes user handler)



実験プラットフォーム

igel

- *Renesas RTS7751R2D board*
  - *SH7751R(SH-4 architecture)*
  - *Peripherals*
    - *Network (RTL-8139DL)*
    - *Serials (SCIF, SM501 UART)*
    - *USB (provided by SM501)*
    - *2D graphics (SM501)*
    - *PCI buses*
    - *CF/PCMCIA slots*
- *Base software*
  - *Linux-2.6.21-rc5*
  - *patch-2.6.21-rc5-rt12*



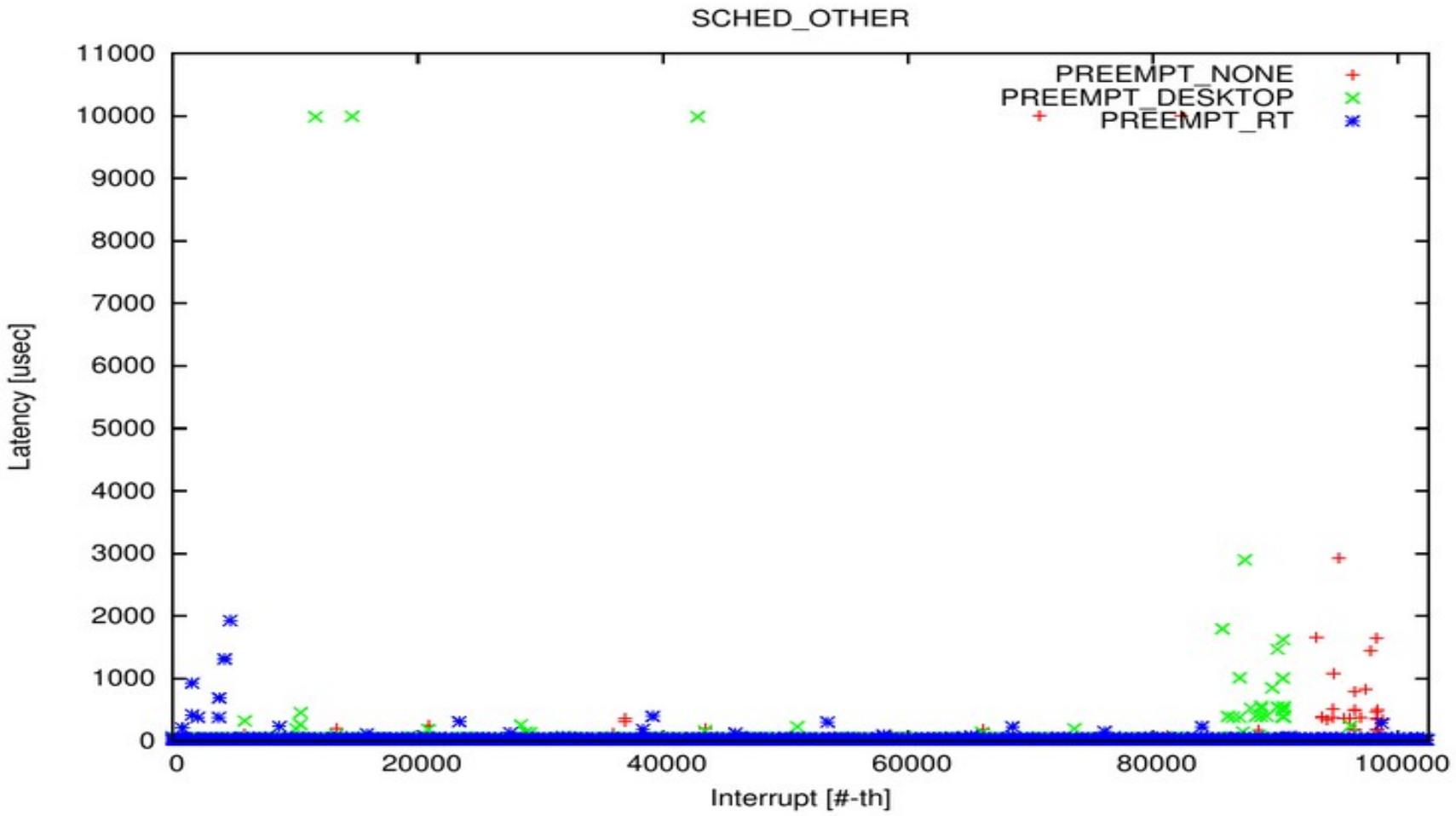
実験条件

igel

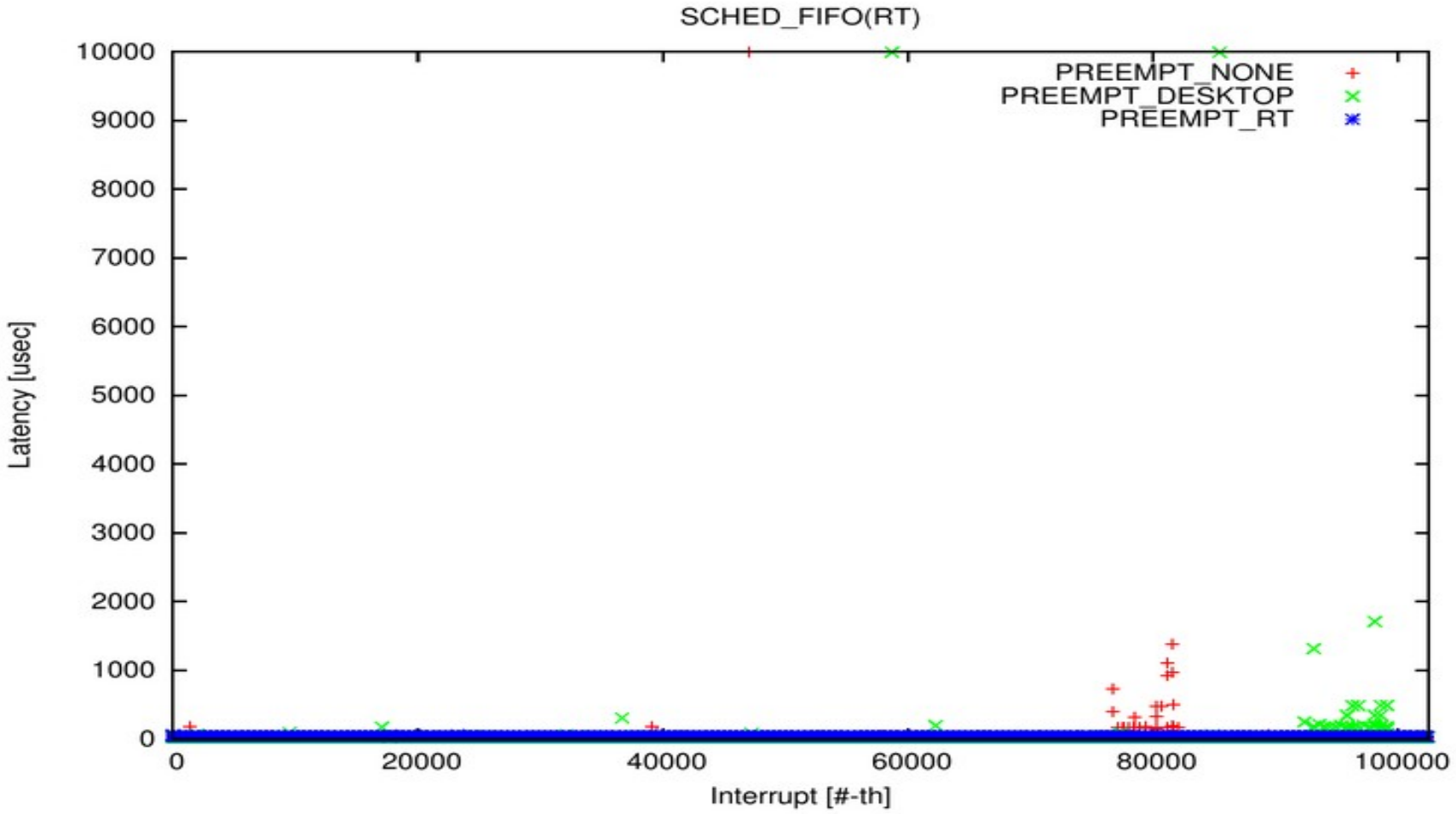
LOAD	SCHED	PREEMPT
none （負荷なし）	_OTHER （非RTタスク）	NONE(CONFIG_PREEMPT not set)
		DESKTOP(CONFIG_PREEMPT=y)
		RT(CONFIG_PREEMPT_RT=y)
	_FIFO （Rtタスク）	NONE(CONFIG_PREEMPT not set)
		DESKTOP(CONFIG_PREEMPT=y)
		RT(CONFIG_PREEMPT_RT=y)
ping -f （ネットワークを 使った高負荷状態）	_OTHER （非RTタスク）	NONE(CONFIG_PREEMPT not set)
		DESKTOP(CONFIG_PREEMPT=y)
		RT(CONFIG_PREEMPT_RT=y)
	_FIFO （Rtタスク）	NONE(CONFIG_PREEMPT not set)
		DESKTOP(CONFIG_PREEMPT=y)
		RT(CONFIG_PREEMPT_RT=y)

結果：非RTタスク／負荷なし

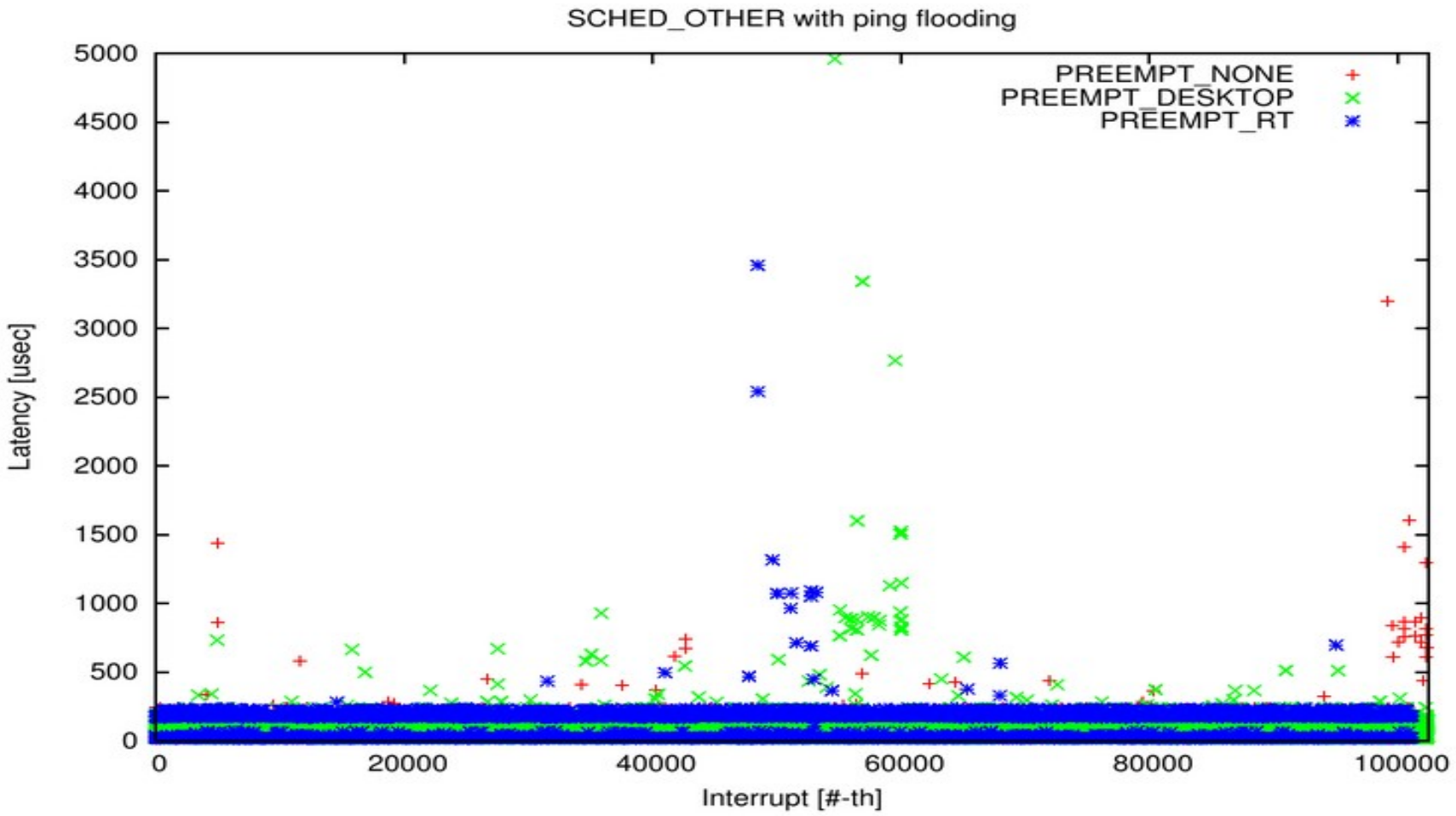
igel



結果:RTタスク／負荷なし

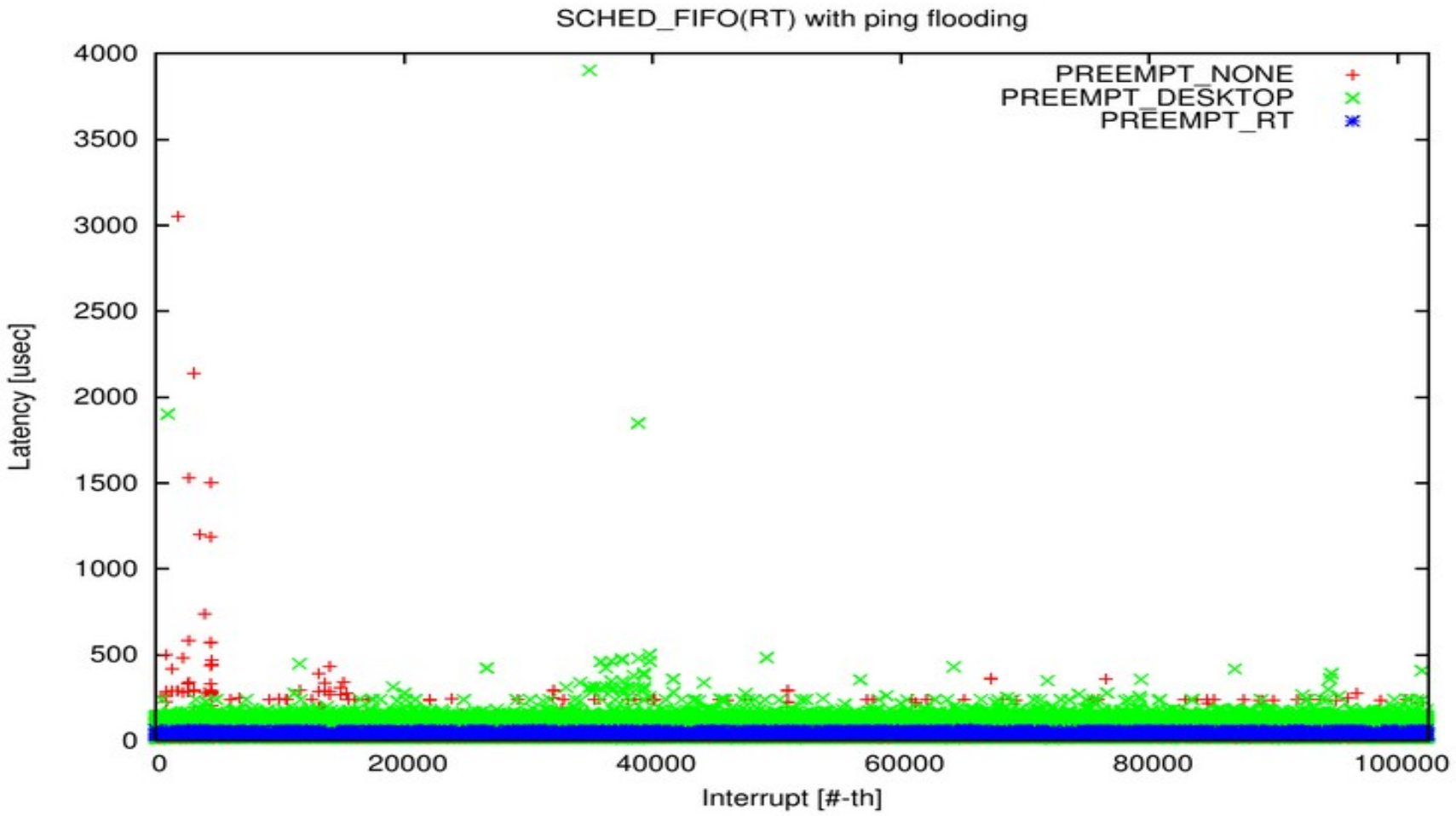


結果：非RTタスク／負荷あり



結果:RTタスク／負荷あり

igel



結果まとめ

igel

LOAD	SCHED	PREEMPT	MIN (usec)	MAX (usec)	AVG (usec)
none	_OTHER	_NONE	10.13	10001.73	13.61
		_DESKTOP	11.80	9994.60	20.04
		_RT	15.40	1925.27	17.06
	_FIFO	_NONE	10.33	9997.73	11.82
		_DESKTOP	9.00	9992.13	16.62
		_RT	29.13	60.47	34.89
ping -f	_OTHER	_NONE	9.93	3196.20	53.59
		_DESKTOP	7.53	4960.20	73.02
		_RT	14.07	3459.00	56.23
	_FIFO	_NONE	10.80	3052.27	50.30
		_DESKTOP	8.93	3901.60	64.17
		_RT	29.07	68.40	41.28

PREEMPT_RTの最新動向

igel

- メインライン化の状況
  - 済：優先度継承つきロック
  - 済：Sleepableスピンロック
  - 未：プロセスコンテキストによる割り込み処理
- ベンチマーク・評価
  - cyclictest by Thomas Gleixner @ Linutronix
  - IBM Test Cases (rt-test) by Darren Hart @ IBM
  - 'Realtime Testting Best Practices' at elinux-wiki, summarized by CELF Realtime WG
- SMP対応
  - RTタスクのCPU間ロードバランシング

おわりに：背景再び…

igel

LinuxはリアルタイムOSではない=組み込みに不向き？？

システム全体のタスクプライオリティ設計をきちんとやったのに、思うようにタスクが動かないなぁ。

RTタスクを使っていると、システムがしばらくフリーズすることがある。なぜだろう…

システム規模が大きくなるにつれて、加速度的に性能が悪くなる（遅延が大きくなる）なぁ…

良く聞くトラブル

おわりに:スケジューラを理解すれば…

igel

リアルタイム要求に応じた適切な設計が可能に！！

RTタスクは、非RTタスクよりも常に優先されるので、占有時間を短くしないと。

処理時間が長くなりすぎて、CPU型と判断されてしまったために、なかなかディスパッチされないんだな。

今回のシステムは、タスク数が多くなりそうなので、2.6カーネルを使うのがいいな。

良く聞くトラブル

igel

Thanks for your kind attention.