

Evaluation of response time with memory access load

Yoshitake Kobayashi and Kouta Okamoto
Advanced Software Technology Group
Corporate Software Engineering Center
TOSHIBA CORPORATION

2010/9/3

Outline

- はじめに
- **Jamboree 33のおさらい**
- **評価環境**
 - ターゲット
 - 周期実行遅延の測定プログラムフロー
 - 測定環境
- **測定手法と測定結果**
 - リアルタイム性能比較
- **まとめ**

はじめに

- 前回のJamboree 33で各種リアルタイム拡張を用いた周期タスクのレイテンシ評価の話をしました
 - http://elinux.org/images/3/33/Verification_of_response_time-20100604.pdf
- コメント・・・
 - 「メモリの負荷を掛けた場合にどうなるの？」
- ということで、やってみました。

おさらい:Linuxでのリアルタイム実現方式

以下のような実現方式がある

■ 標準Linuxカーネル

- カーネルに変更を加えたくない場合
- そこそこのレスポンスが得られる

本日の話題の範囲

■ RTパッチ適用カーネル

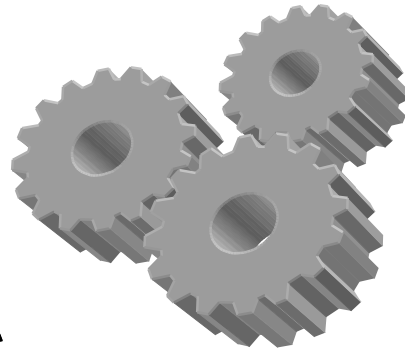
- 安定したレスポンスを必要とする場合
- Linuxの標準APIを利用可能

■ Xenomai

- より短く安定したレスポンスを必要とする場合
- XenomaiのリアルタイムAPIを使用する必要あり

■ 今回は時間の関係でXenomaiの評価は行っておりません

おさらいの続き



知りたいこと

- 最良値
- 最悪値
- レスポンスのばらつき



レスポンスの種類

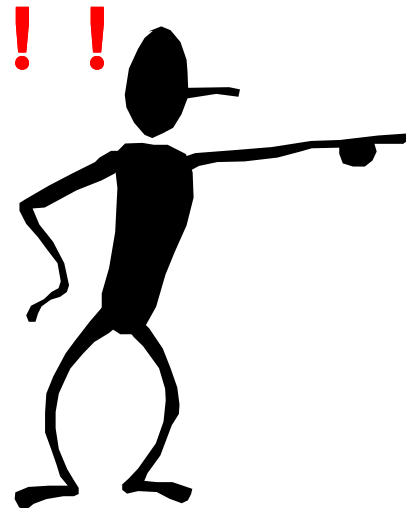
- 割り込みのレスポンス
- プロセス起動のレスポンス
- 周期起動のレスポンス
- etc...

今回も...

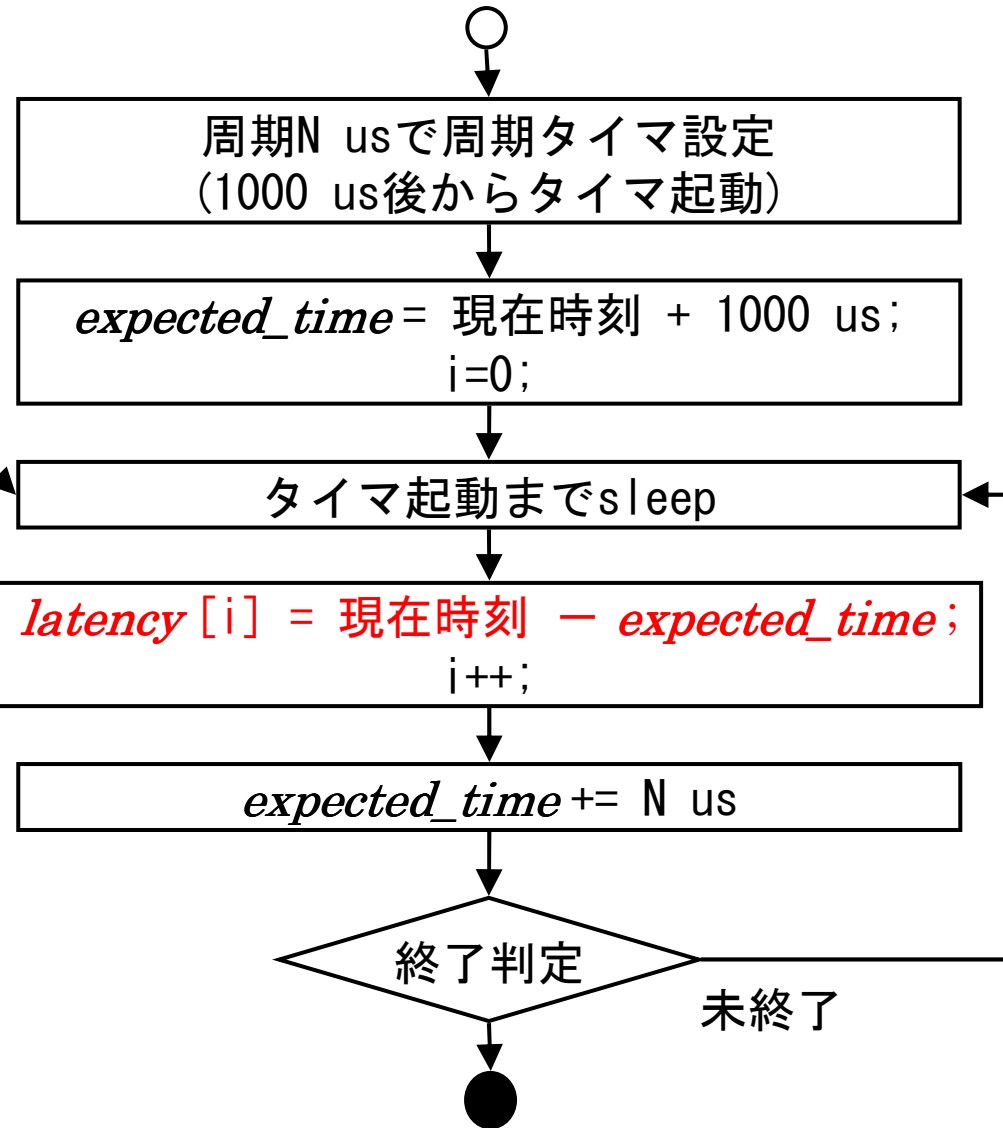
周期実行における
レスポンス(レイテンシ)を検証

測定プログラム情報

周期起動LATENCY 測定フロー



タイマ



測定環境

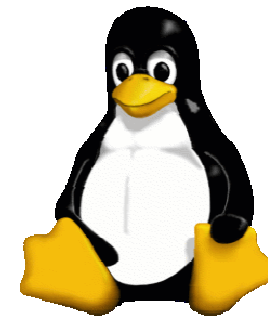
■ ハードウェア

- CPU: Pentium4 2.66 GHz (L1キャッシュ:512KB)
- メモリ: 512MB



■ ソフトウェア

- カーネル
 1. Vanillaカーネル (kernel-2.6.31.12)
 - カーネル内プリエンプションを設定
 2. RTカーネル (kernel-2.6.31.12-rt21)
 - RTパッチ適用+リアルタイム処理向けのチューニング



※: レイテンシの測定結果はマイクロ秒で整理

基礎評価(CPU負荷のみ)

■ CPU負荷プログラム

1. 演算を永遠に繰り返す

■ 測定パラメータ

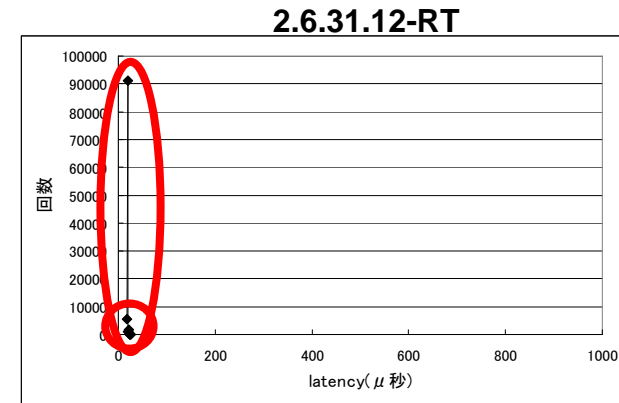
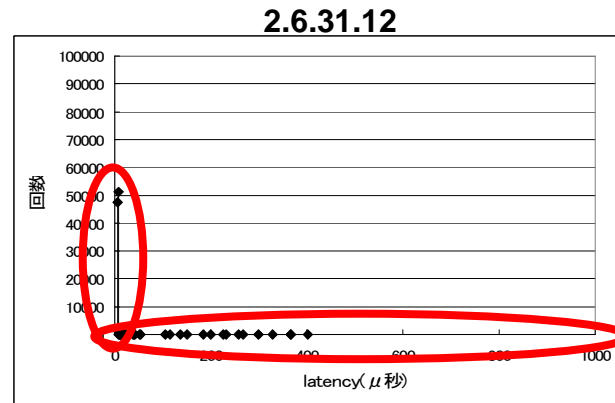
- サンプル数
 - 100000
- 周期
 - 300us, 500us, 1000us
- 負荷プログラムによるCPU負荷
 - (負荷なし), 50%, 100%



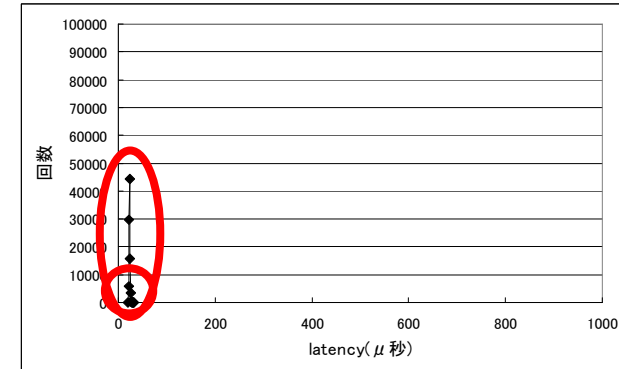
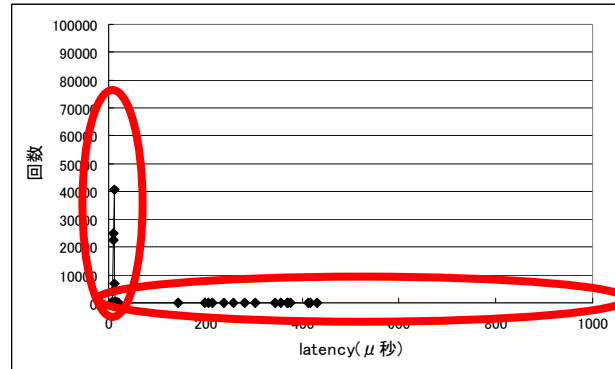
測定結果(周期300us/CPU負荷)

CPU負荷

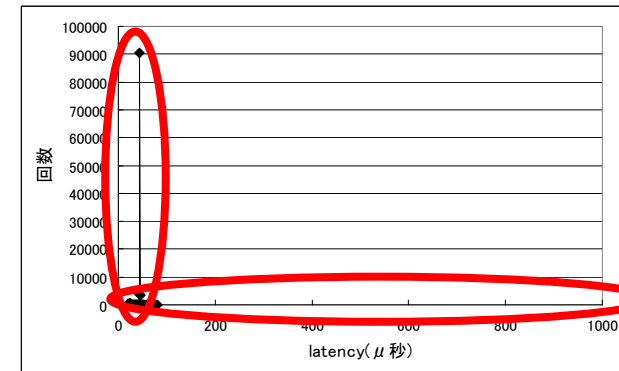
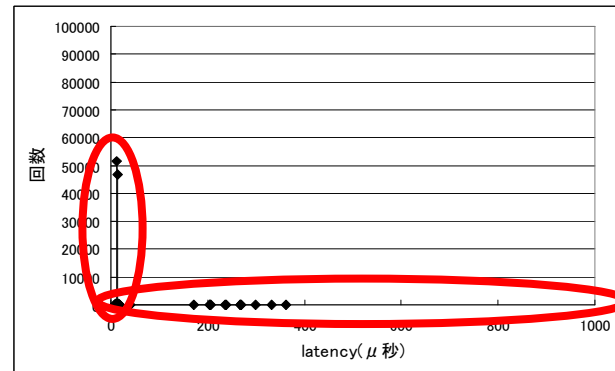
0 %



50 %



100 %



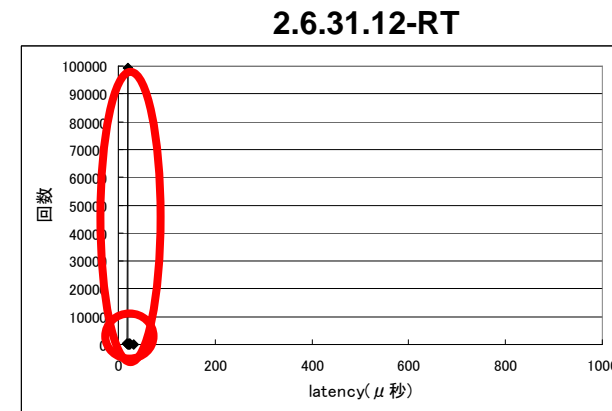
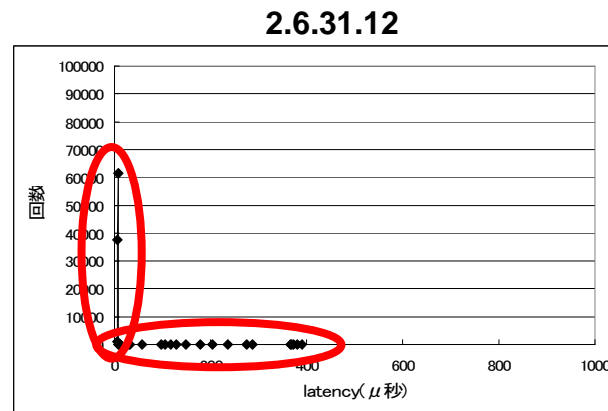
測定結果(周期300us/CPU負荷)

カーネル	CPU負荷率	平均レイテンシ	最大レイテンシ	標準偏差	周期超過回数
2.6.31.12	0%	7.036	402.131	3.167	4
	50%	10.768	430.681	4.054	9
	100%	13.043	361.986	3.131	2
2.6.31.12-RT	0%	19.162	25.629	0.422	0
	50%	23.340	32.353	0.979	0
	100%	44.624	28282.187	12.87	1

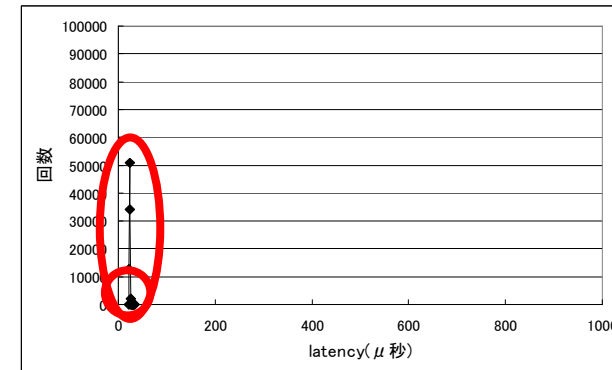
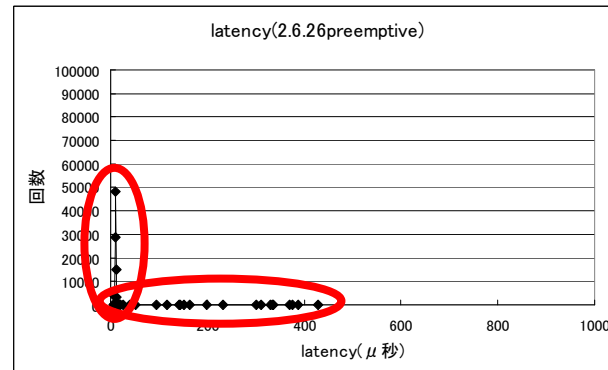
測定結果(周期500us/CPU負荷)

CPU負荷

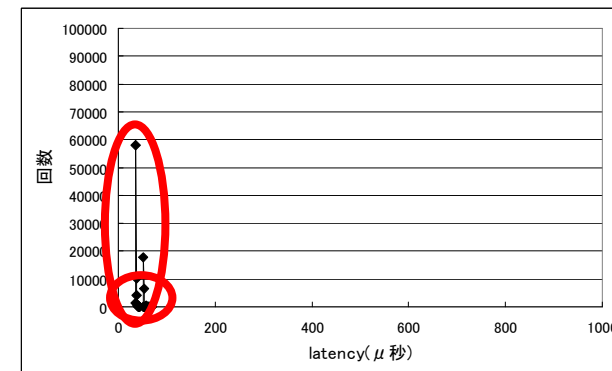
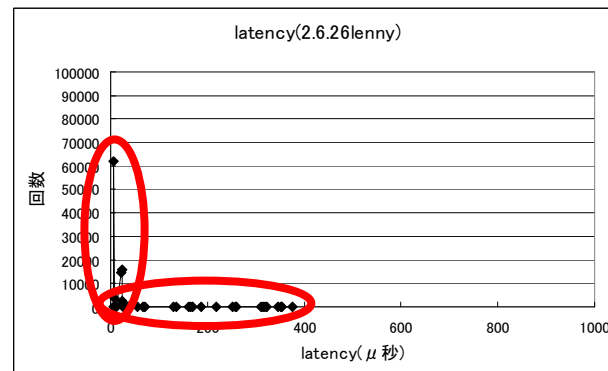
0 %



50 %



100 %



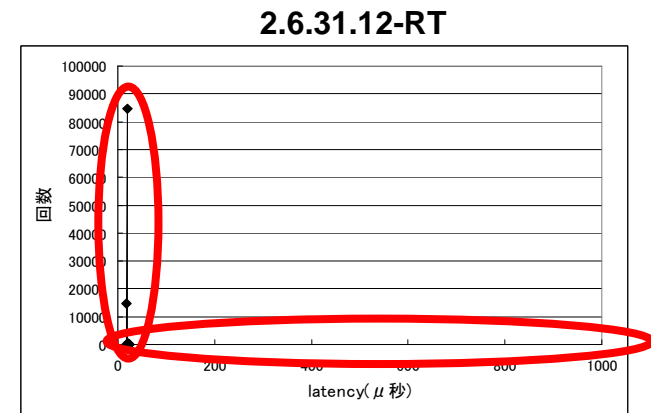
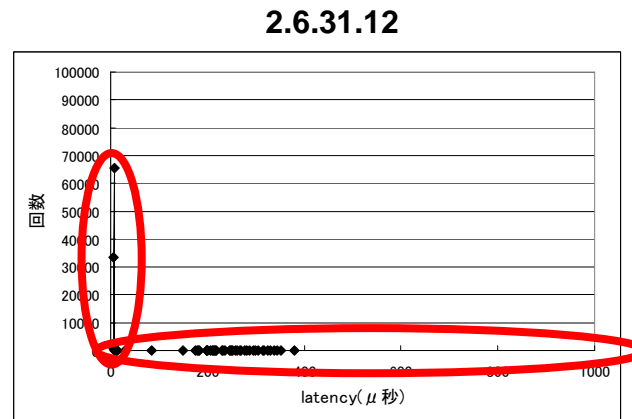
測定結果(周期500us/CPU負荷)

カーネル	CPU負荷率	平均レイテンシ	最大レイテンシ	標準偏差	周期超過回数
2.6.31.12	0%	7.068	391.005	3.259	0
	50%	10.420	428.748	4.107	0
	100%	12.370	375.133	8.661	0
2.6.31.12-RT	0%	19.269	32.615	0.246	0
	50%	23.763	33.263	0.802	0
	100%	40.802	69.411	7.02	0

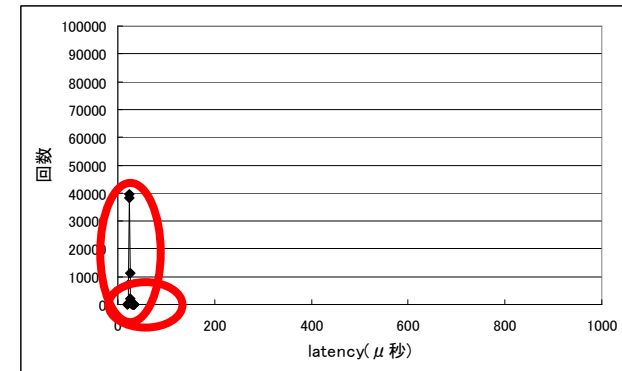
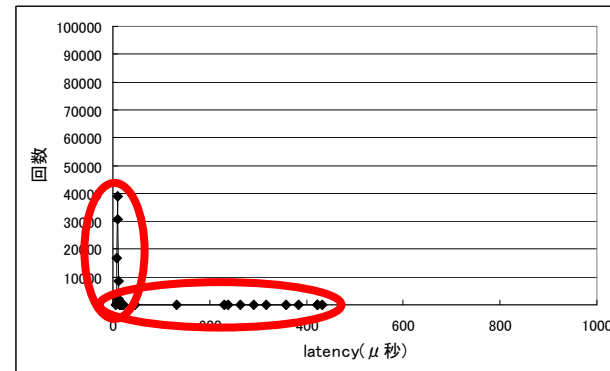
測定結果(周期1000us/CPU負荷)

CPU負荷

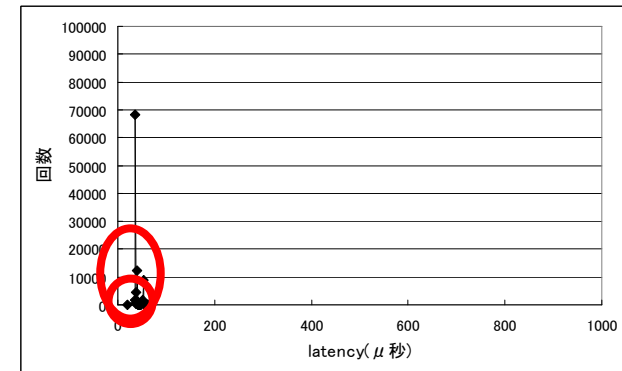
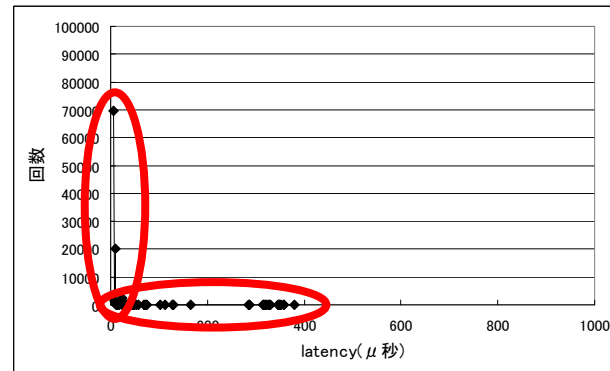
0 %



50 %



100 %



測定結果(周期1000us/CPU負荷)

カーネル	CPU負荷率	平均レイテンシ	最大レイテンシ	標準偏差	周期超過回数
2.6.31.12	0%	7.455	27395.703	13.629	1
	50%	9.881	433.542	3.534	0
	100%	8.122	380.988	6.216	0
2.6.31.12-RT	0%	19.361	27207.563	12.593	1
	50%	24.162	34.459	1.041	0
	100%	39.129	70.020	5.369	0

メモリの負荷をどうかけるか？

- そもそもメモリの負荷とは？

- 今回はメモリの負荷を以下のように定義
 1. メモリ割当てと開放処理が頻繁にある状態
 - ページフォルトが発生
 - メモリアクセス違反処理が発生
 2. メモリアクセスが頻繁にある状態

注：スワップアウトはしない

メモリ確保・開放負荷について

■ ページ確保・開放負荷プログラム

1. mmapを利用して領域を確保(読み書き可能)
2. 確保した領域の各ページに何か書く
3. CPU負荷がかかり過ぎないようにsleepで調整
4. unmapして開放
5. 1~4を永遠に繰り返す

■ 測定パラメータ

■ 周期

- 300us, 500us, 1000us

■ 負荷プログラムによるCPU負荷

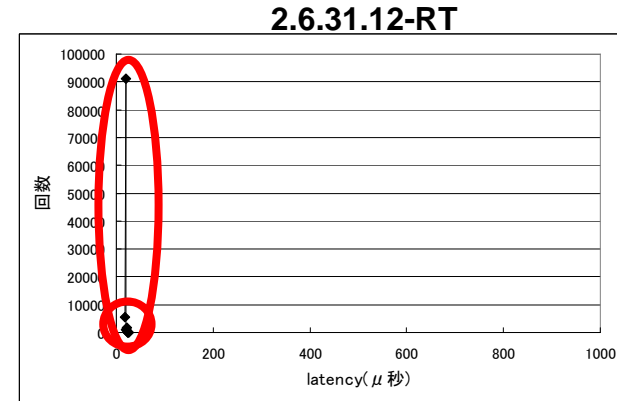
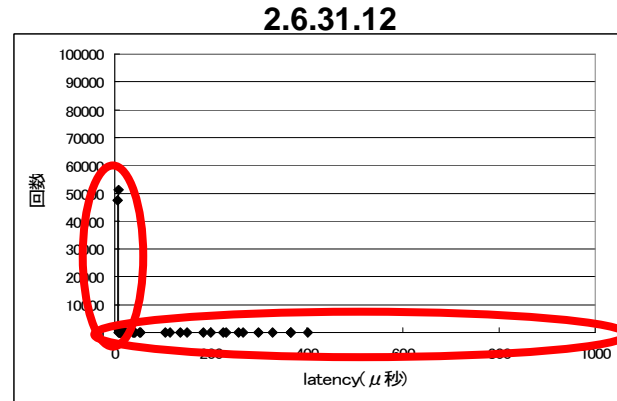
- (負荷なし), 50%, 100%



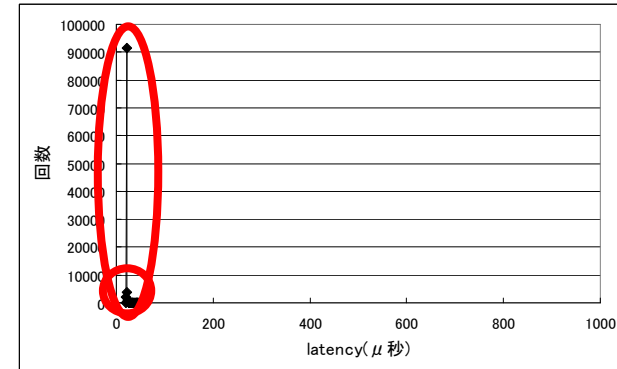
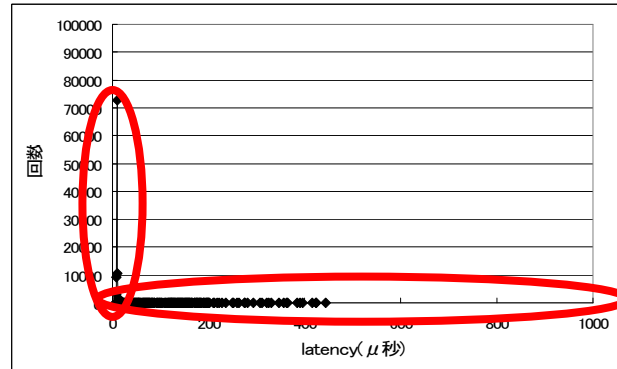
測定結果(周期300us/メモリ確保・開放負荷)

CPU負荷

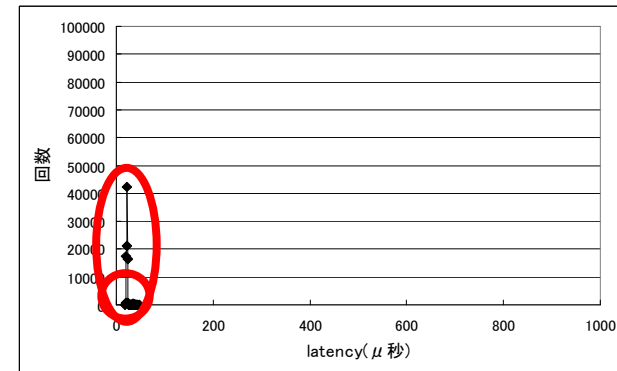
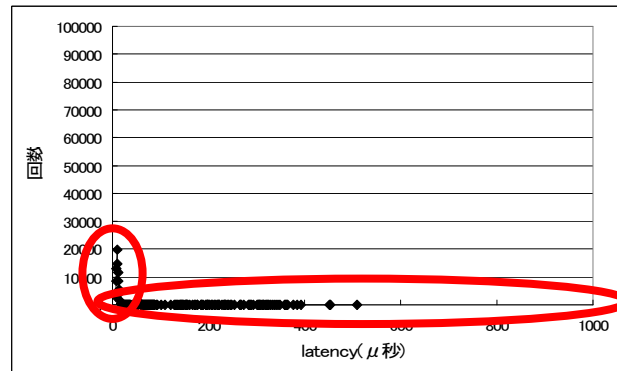
0%



50%



100%



測定結果(周期300us/メモリ確保・開放負荷)

カーネル	CPU負荷率	平均レイテンシ	最大レイテンシ	標準偏差	周期超過回数
2.6.31.12	0%	7.036	402.131	3.167	4
	50%	10.513	443.732	10.725	16
	100%	12.547	508.996	12.108	35
2.6.31.12-RT	0%	19.162	25.629	0.422	0
	50%	21.464	45.758	0.806	0
	100%	22.213	47.250	1.551	0

カーネル	CPU負荷率	平均レイテンシ	最大レイテンシ	標準偏差	周期超過回数
2.6.31.12	0%	7.036	402.131	3.167	4
	50%	10.768	430.681	4.054	9
	100%	13.043	361.986	3.131	2
2.6.31.12-RT	0%	19.162	25.629	0.422	0
	50%	23.340	32.353	0.979	0
	100%	44.624	28282.187	12.87	1

CPU負荷のみ

測定結果(周期300us/メモリ確保・開放負荷)

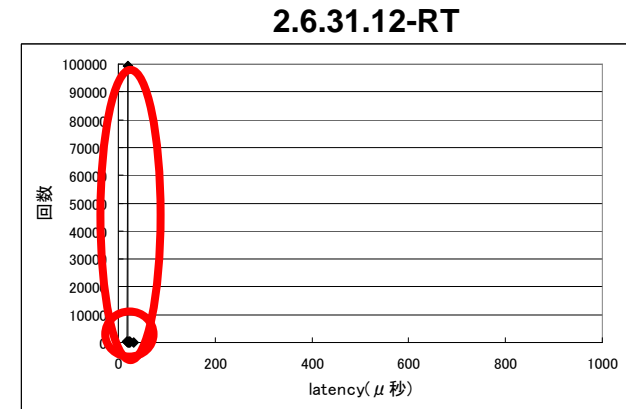
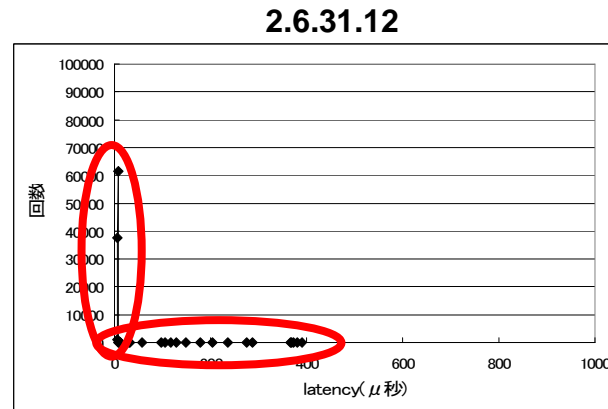
カーネル	CPU負荷率	平均レイテンシ	最大レイテンシ	標準偏差	周期超過回数
2.6.31.12	0%	7.036	402.131	3.167	4
	50%	10.513	443.732	10.725	16
	100%	12.547	508.996	12.108	35
2.6.31.12-RT	0%	19.162	25.629	0.422	0
	50%	21.464	45.758	0.806	0
	100%	22.213	47.250	1.551	0

- VanillaカーネルではCPU負荷のみと比べ、周期超過回数が多くなっている
- RTカーネルでは安定して動作しているように思われる

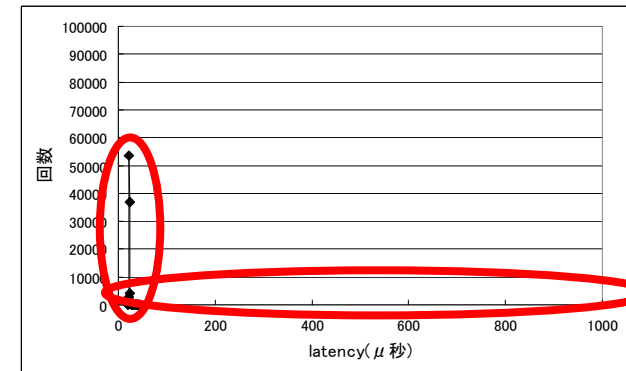
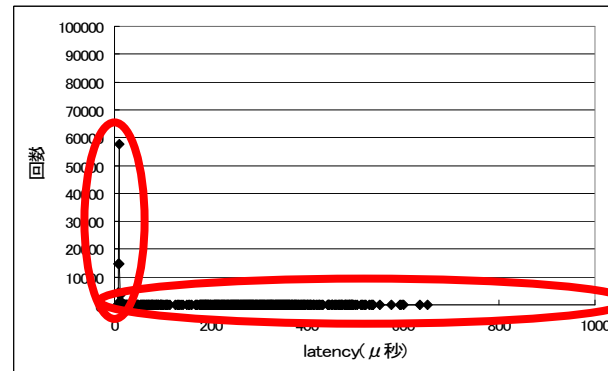
測定結果(周期500us/メモリ確保・開放負荷)

CPU負荷

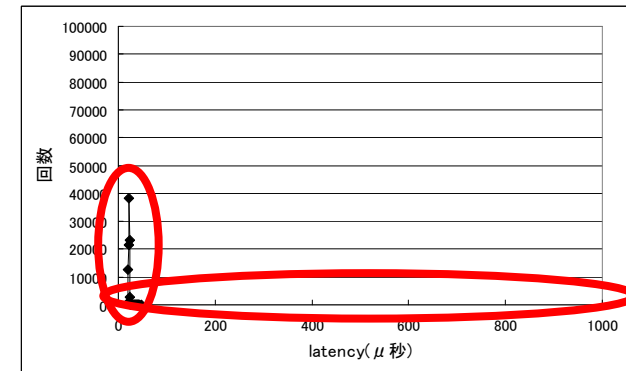
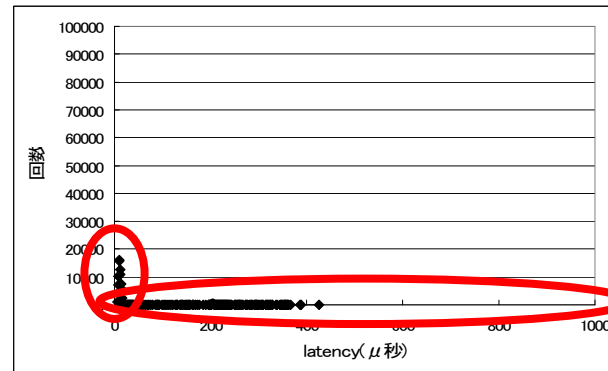
0 %



50 %



100 %



測定結果(周期500us/メモリ確保・開放負荷)

カーネル	CPU負荷率	平均レイテンシ	最大レイテンシ	標準偏差	周期超過回数
2.6.31.12	0%	7.068	391.005	3.259	0
	50%	12.196	27975.413	28.714	19
	100%	17.040	27852.210	35.182	1
2.6.31.12-RT	0%	19.269	32.615	0.246	0
	50%	23.199	27943.415	12.616	1
	100%	22.701	27969.027	12.688	1

カーネル	CPU負荷率	平均レイテンシ	最大レイテンシ	標準偏差	周期超過回数
2.6.31.12	0%	7.068	391.005	3.259	0
	50%	10.420	428.748	4.107	0
	100%	12.370	375.133	8.661	0
2.6.31.12-RT	0%	19.269	32.615	0.246	0
	50%	23.763	33.263	0.802	0
	100%	40.802	69.411	7.02	0

CPU負荷のみ

測定結果(周期500us/メモリ確保・開放負荷)

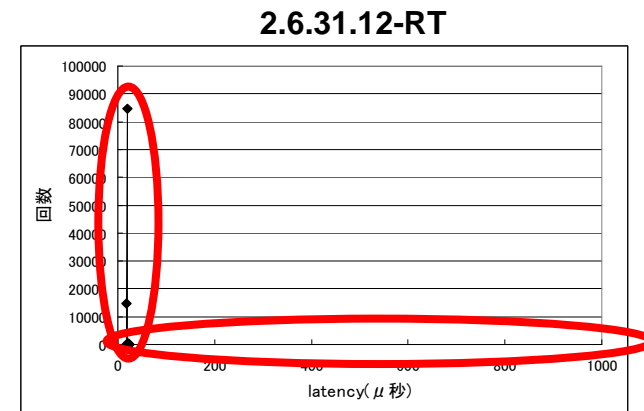
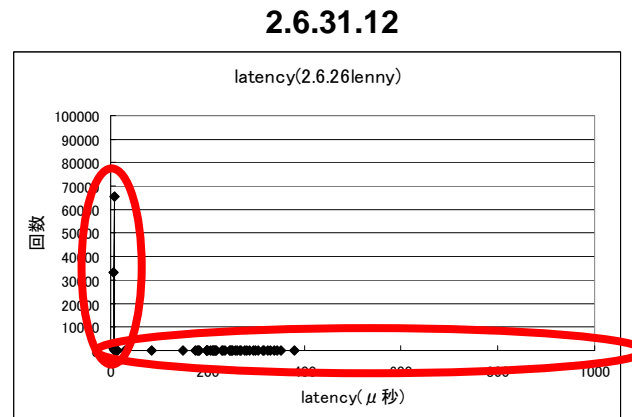
カーネル	CPU負荷率	平均レイテンシ	最大レイテンシ	標準偏差	周期超過回数
2.6.31.12	0%	7.068	391.005	3.259	0
	50%	12.196	27975.413	28.714	19
	100%	17.040	27852.210	35.182	1
2.6.31.12-RT	0%	19.269	32.615	0.246	0
	50%	23.199	27943.415	12.616	1
	100%	22.701	27969.027	12.688	1

- VanillaカーネルではCPU負荷のみと比べ、周期超過回数が多くなっている
- RTカーネルは、300us周期では安定していたように見えたが、500us周期ではデッドラインミスが発生している

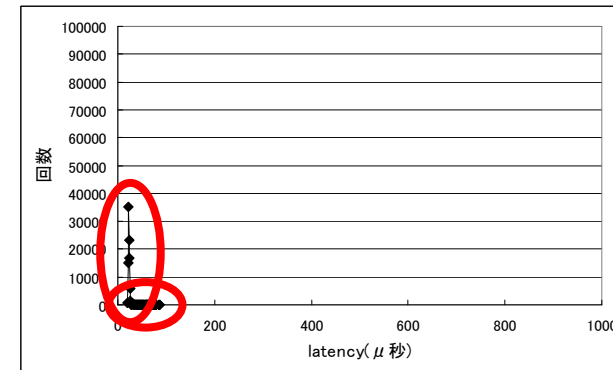
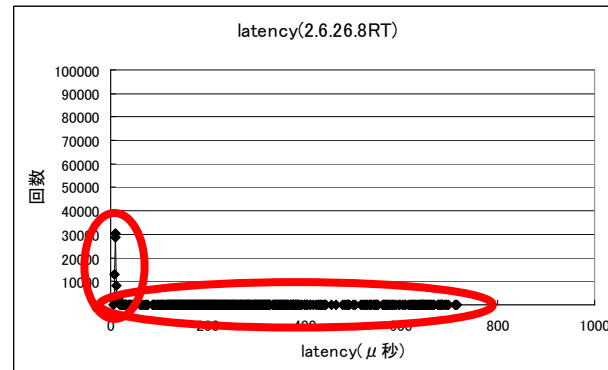
測定結果(周期1000us/メモリ確保・開放負荷)

CPU負荷

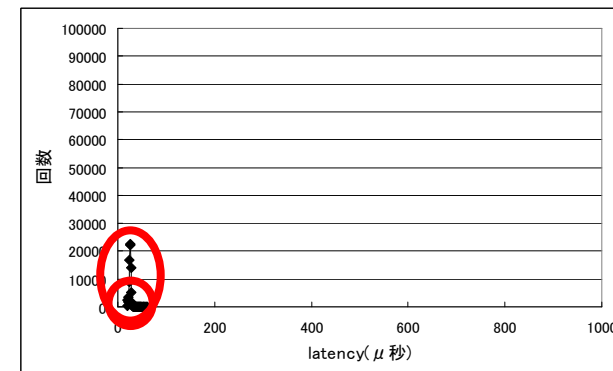
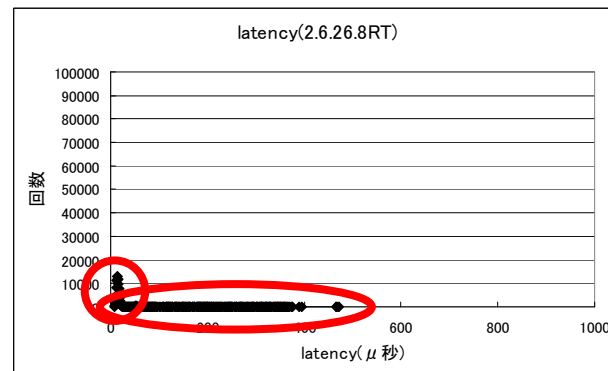
0 %



50 %



100 %



測定結果(周期1000us/メモリ確保・開放負荷)

カーネル	CPU負荷率	平均レイテンシ	最大レイテンシ	標準偏差	周期超過回数
2.6.31.12	0%	7.455	27395.703	13.629	1
	50%	13.184	715.078	29.190	0
	100%	17.143	472.499	15.572	0
2.6.31.12-RT	0%	19.361	27207.563	12.593	1
	50%	23.451	86.734	2.839	0
	100%	25.760	60.128	2.328	0

CPU負荷のみ	カーネル	CPU負荷率	平均レイテンシ	最大レイテンシ	標準偏差	周期超過回数
	2.6.31.12	0%	7.455	27395.703	13.629	1
		50%	9.881	433.542	3.534	0
		100%	8.122	380.988	6.216	0
	2.6.31.12-RT	0%	19.361	27207.563	12.593	1
		50%	24.162	34.459	1.041	0
		100%	39.129	70.020	5.369	0

測定結果(周期1000us/メモリ確保・開放負荷)

カーネル	CPU負荷率	平均レイテンシ	最大レイテンシ	標準偏差	周期超過回数
2.6.31.12	0%	7.455	27395.703	13.629	1
	50%	13.184	715078	29.190	0
	100%	17.143	472.499	15.572	0
2.6.31.12-RT	0%	19.361	27207.563	12.593	1
	50%	23.451	86.734	2.839	0
	100%	25.760	60.128	2.328	0

- Vanilla、RTカーネル共に同じ傾向のようにみえる
- 全周期を通して、ばらつきはRTカーネルの方が小さい

メモリ確保・開放＋アクセス違反回避負荷について

■ ページ確保・開放負荷＋アクセス違反回避プログラム

1. mmapを利用して領域を確保 (PROT_NONE 指定をする)
2. 確保した領域の各ページに何か書く
 - SIGSEGVシグナルが返る
 - SIGSEGVが発生したアドレスを読み書き可能にする
3. CPU負荷がかかり過ぎないようにsleepで調整
4. unmapして開放
5. 1～4を永遠に繰り返す

■ 測定パラメータ

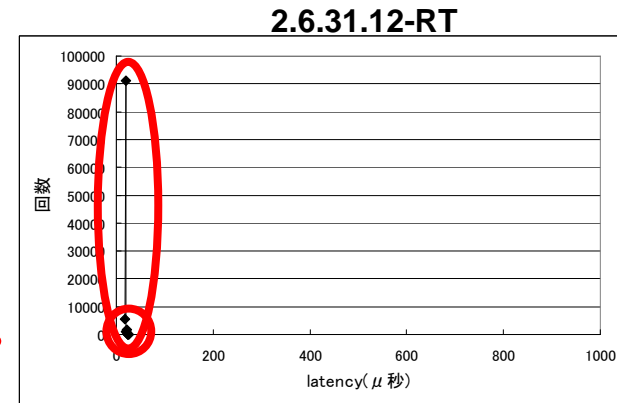
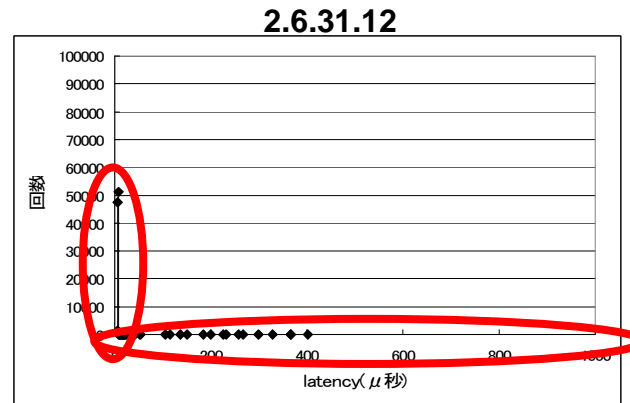
- 周期
 - 300us, 500us, 1000us
- 負荷プログラムによるCPU負荷
 - (負荷なし), 50%, 100%



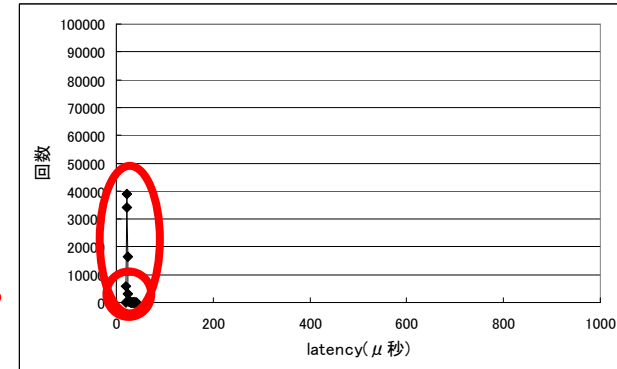
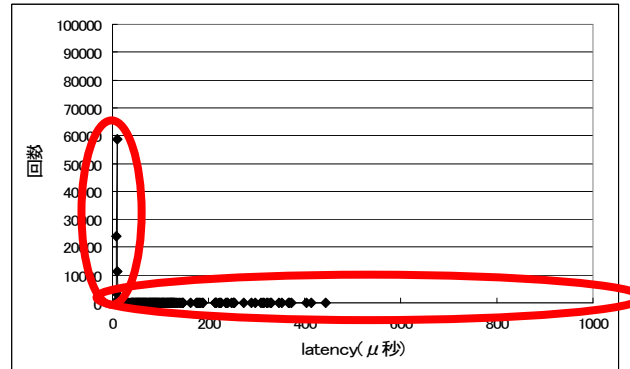
測定結果(周期300us/確保・開放+保護違反処理)

CPU負荷

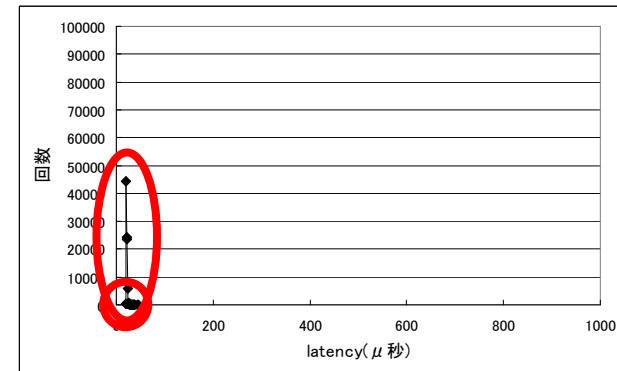
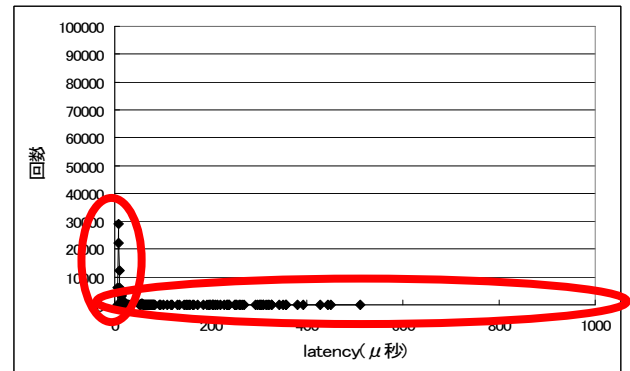
0 %



50 %



100 %



測定結果(周期300us/確保・開放＋保護違反処理)

カーネル	CPU負荷率	平均レイテンシ	最大レイテンシ	標準偏差	周期超過回数
2.6.31.12	0%	7.036	402.131	3.167	4
	50%	10.077	443.753	8.430	15
	100%	10.529	510.858	9.367	21
2.6.31.12-RT	0%	19.162	25.629	0.422	0
	50%	22.347	42.461	1.123	0
	100%	21.531	61.000	1.279	0

カーネル	CPU負荷率	平均レイテンシ	最大レイテンシ	標準偏差	周期超過回数
2.6.31.12	0%	7.036	402.131	3.167	4
	50%	10.768	430.681	4.054	9
	100%	13.043	361.986	3.131	2
2.6.31.12-RT	0%	19.162	25.629	0.422	0
	50%	23.340	32.353	0.979	0
	100%	44.624	28282.187	12.87	1

CPU負荷のみ

測定結果(周期300us/確保・開放＋保護違反処理)

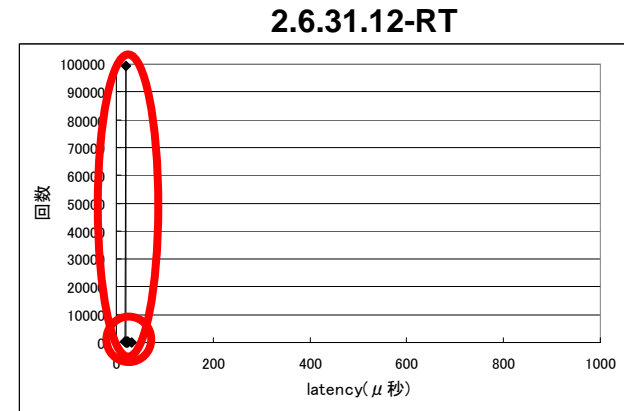
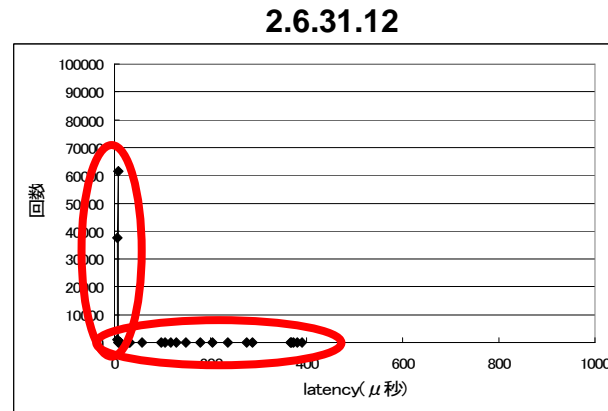
カーネル	CPU負荷率	平均レイテンシ	最大レイテンシ	標準偏差	周期超過回数
2.6.31.12	0%	7.036	402.131	3.167	4
	50%	10.077	443.753	8.430	15
	100%	10.529	510.858	9.367	21
2.6.31.12-RT	0%	19.162	25.629	0.422	0
	50%	22.347	42.461	1.123	0
	100%	21.531	61.000	1.279	0

- VanillaカーネルではCPU負荷のみと比べ、周期超過回数が多くなっている
- RTカーネルでは安定して動作しているように思われる

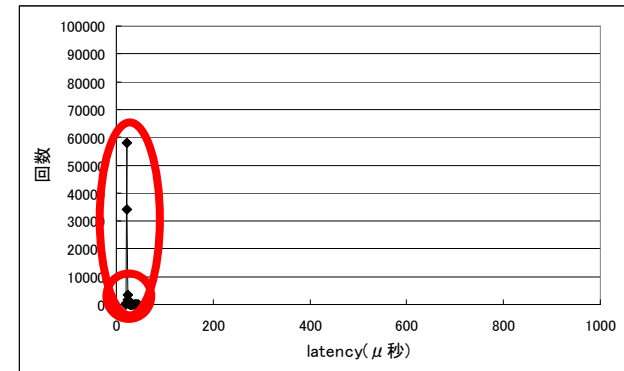
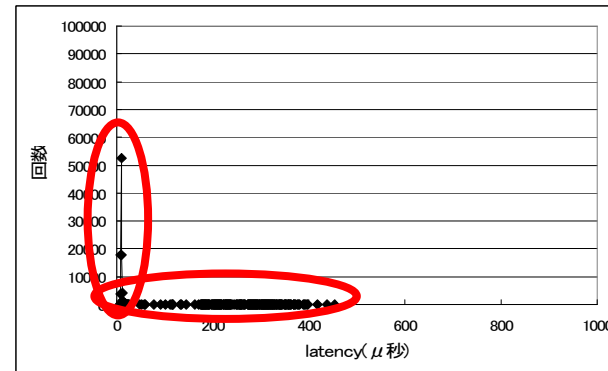
測定結果(周期500us/確保・開放+保護違反処理)

CPU負荷

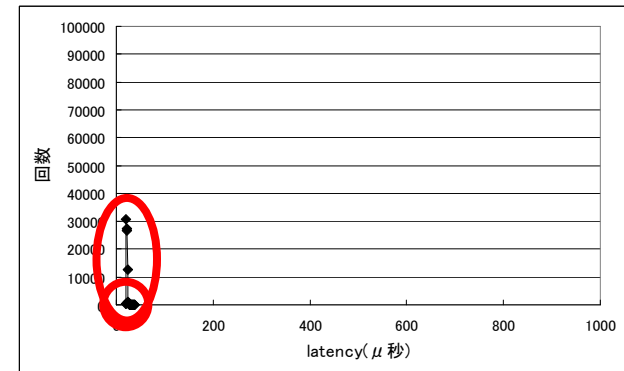
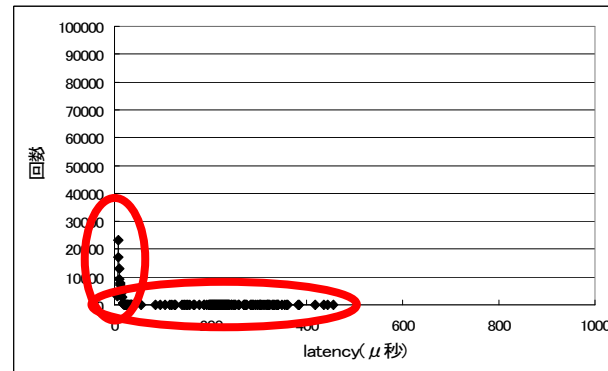
0 %



50 %



100 %



測定結果(周期500us/確保・開放+保護違反処理)

カーネル	CPU負荷率	平均レイテンシ	最大レイテンシ	標準偏差	周期超過回数
2.6.31.12	0%	7.068	391.005	3.259	0
	50%	11.461	454.867	20.858	0
	100%	13.090	456.513	24.011	0
2.6.31.12-RT	0%	19.269	32.615	0.246	0
	50%	22.149	45.367	0.955	0
	100%	21.859	38.625	1.313	0

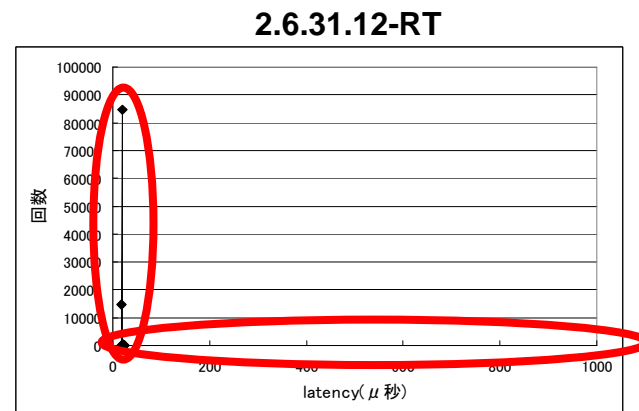
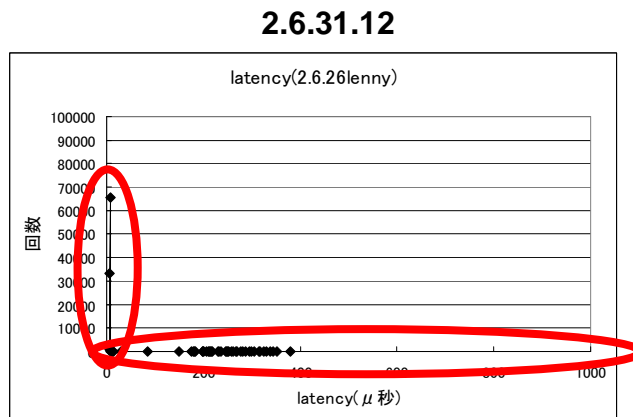
カーネル	CPU負荷率	平均レイテンシ	最大レイテンシ	標準偏差	周期超過回数
2.6.31.12	0%	7.068	391.005	3.259	0
	50%	10.420	428.748	4.107	0
	100%	12.370	375.133	8.661	0
2.6.31.12-RT	0%	19.269	32.615	0.246	0
	50%	23.763	33.263	0.802	0
	100%	40.802	69.411	7.02	0

CPU負荷のみ

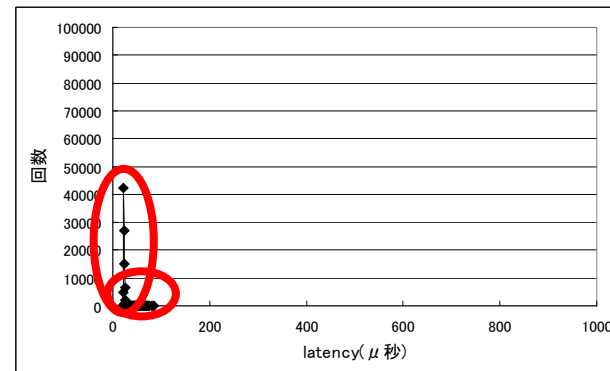
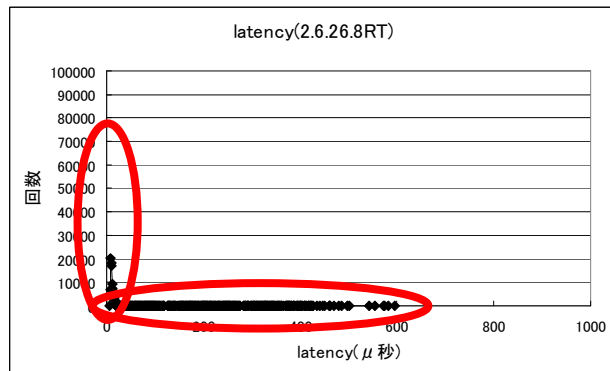
測定結果(周期1000 μ s/確保・開放+保護違反処理)

CPU負荷

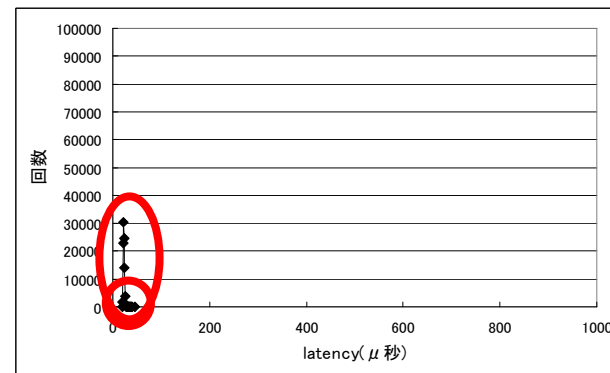
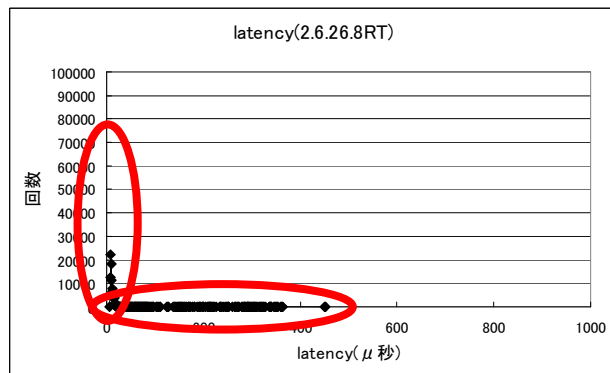
0 %



50 %



100 %



測定結果(周期1000us/確保・開放+保護違反処理)

カーネル	CPU負荷率	平均レイテンシ	最大レイテンシ	標準偏差	周期超過回数
2.6.31.12	0%	7.455	27395.703	13.629	1
	50%	12.401	596.112	19.733	0
	100%	11.840	451.635	11.632	0
2.6.31.12-RT	0%	19.361	27207.563	12.593	1
	50%	23.471	84.172	1.861	0
	100%	23.107	46.094	1.486	0

CPU負荷のみ

カーネル	CPU負荷率	平均レイテンシ	最大レイテンシ	標準偏差	周期超過回数
2.6.31.12	0%	7.455	27395.703	13.629	1
	50%	9.881	433.542	3.534	0
	100%	8.122	380.988	6.216	0
2.6.31.12-RT	0%	19.361	27207.563	12.593	1
	50%	24.162	34.459	1.041	0
	100%	39.129	70.020	5.369	0

メモリアクセス負荷による測定

■ メモリアクセス負荷プログラム

1. volatile指定した変数カウンタを更新
2. CPU負荷がかかり過ぎないようにsleepで調整
3. 1～2を繰り返す

■ 測定パラメータ

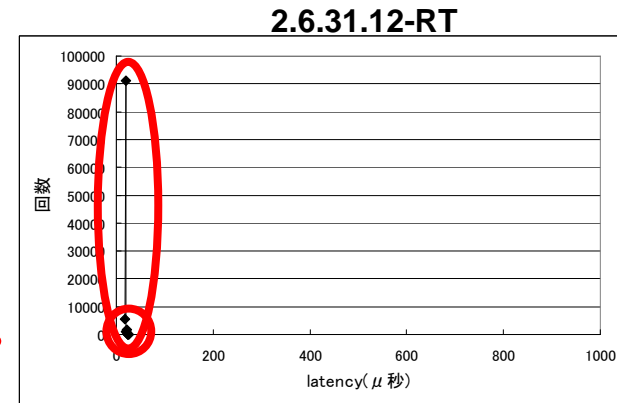
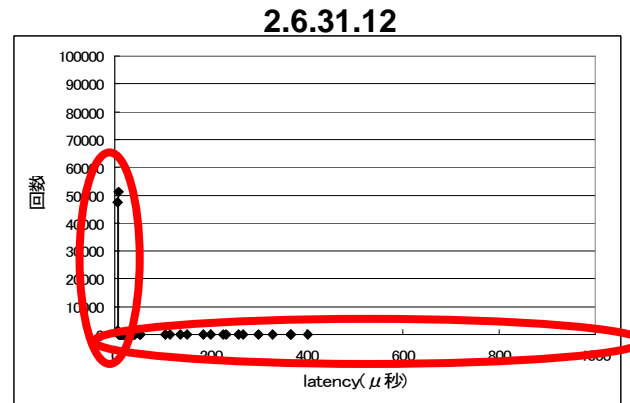
- 周期
 - 300us, 500us, 1000us
- 負荷プログラムによるCPU負荷
 - (負荷なし), 50%, 100%



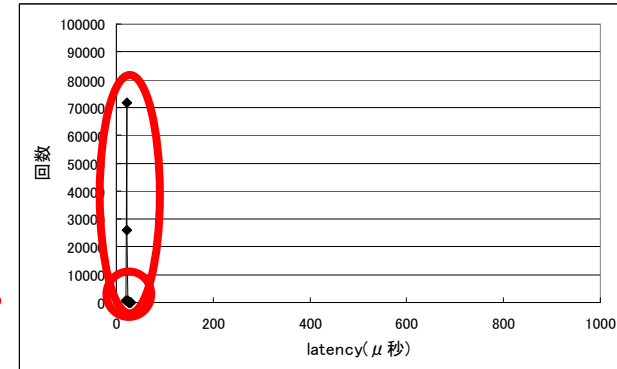
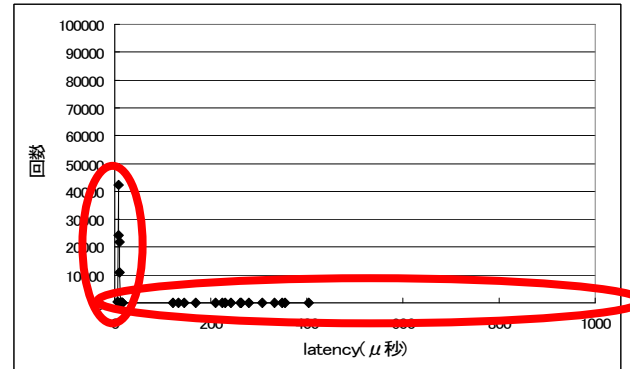
測定結果(周期300us/メモリアクセス負荷)

CPU負荷

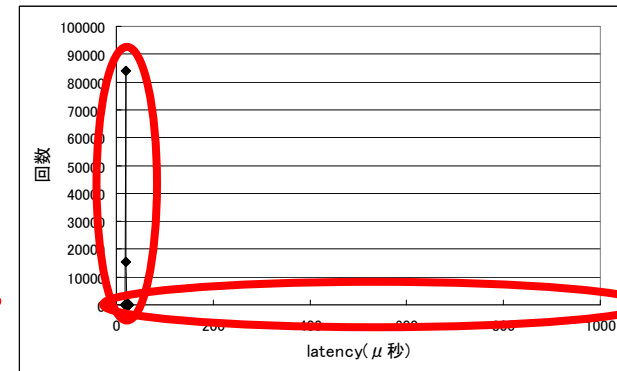
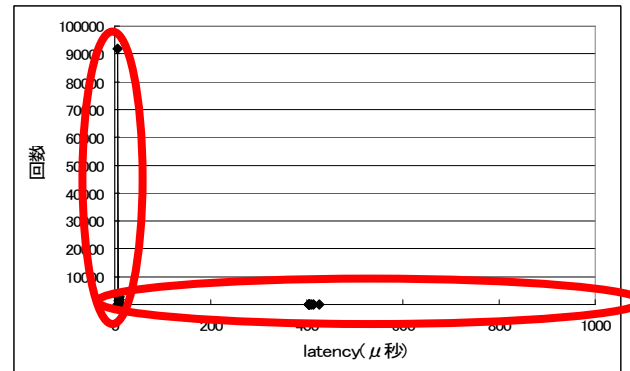
0%



50%



100%



測定結果(周期300us/メモリアクセス負荷)

カーネル	CPU負荷率	平均レイテンシ	最大レイテンシ	標準偏差	周期超過回数
2.6.31.12	0%	7.036	402.131	3.167	4
	50%	8.602	404.506	3.414	5
	100%	7.086	28143.442	13.569	16
2.6.31.12-RT	0%	19.162	25.629	0.422	0
	50%	21.985	30.018	0.616	0
	100%	20.176	28160.933	12.594	1

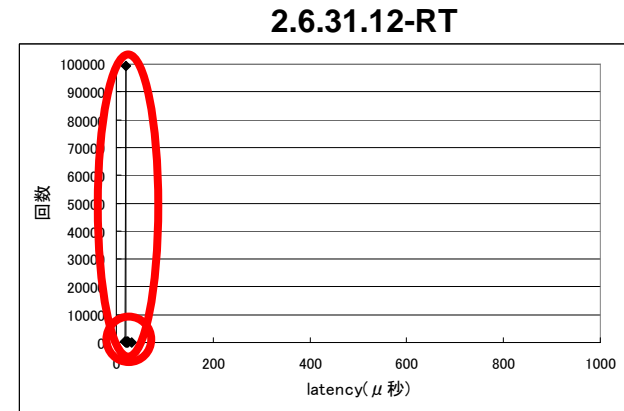
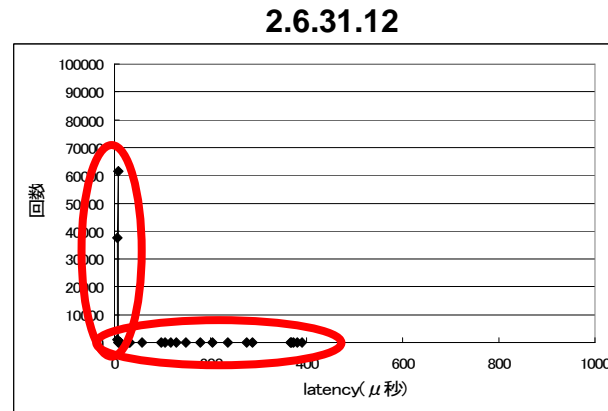
CPU負荷のみ

カーネル	CPU負荷率	平均レイテンシ	最大レイテンシ	標準偏差	周期超過回数
2.6.31.12	0%	7.036	402.131	3.167	4
	50%	10.768	430.681	4.054	9
	100%	13.043	361.986	3.131	2
2.6.31.12-RT	0%	19.162	25.629	0.422	0
	50%	23.340	32.353	0.979	0
	100%	44.624	28282.187	12.87	1

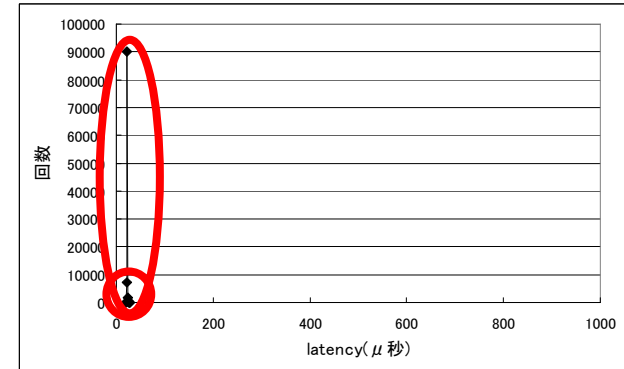
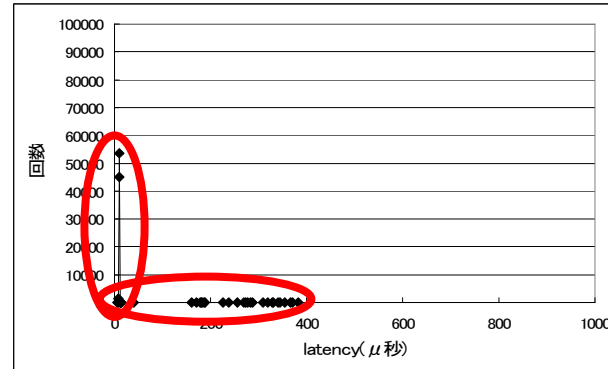
測定結果(周期500us/メモリアクセス負荷)

CPU負荷

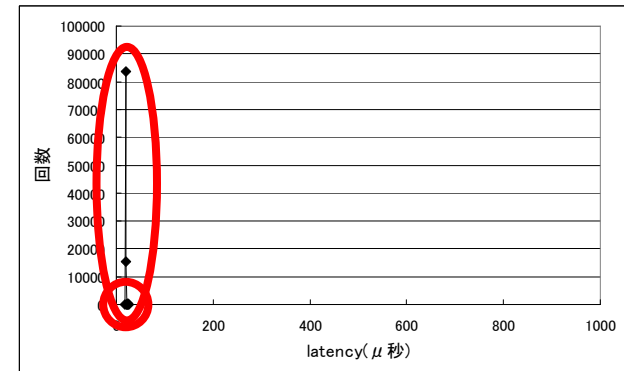
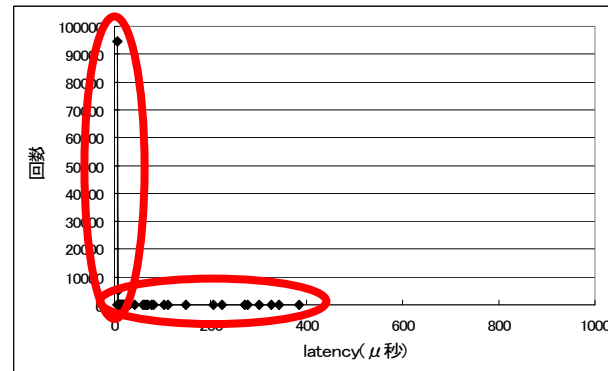
0 %



50 %



100 %



測定結果(周期500us/メモリアクセス負荷)

カーネル	CPU負荷率	平均レイテンシ	最大レイテンシ	標準偏差	周期超過回数
2.6.31.12	0%	7.068	391.005	3.259	0
	50%	10.084	383.569	4.334	0
	100%	6.532	385.648	2.957	0
2.6.31.12-RT	0%	19.269	32.615	0.246	0
	50%	22.261	28.953	0.464	0
	100%	19.865	25.194	0.494	0

カーネル	CPU負荷率	平均レイテンシ	最大レイテンシ	標準偏差	周期超過回数
2.6.31.12	0%	7.068	391.005	3.259	0
	50%	10.42	428.748	4.107	0
	100%	12.37	375.133	8.661	0
2.6.31.12-RT	0%	19.269	32.615	0.246	0
	50%	23.763	33.263	0.802	0
	100%	40.802	69.411	7.02	0

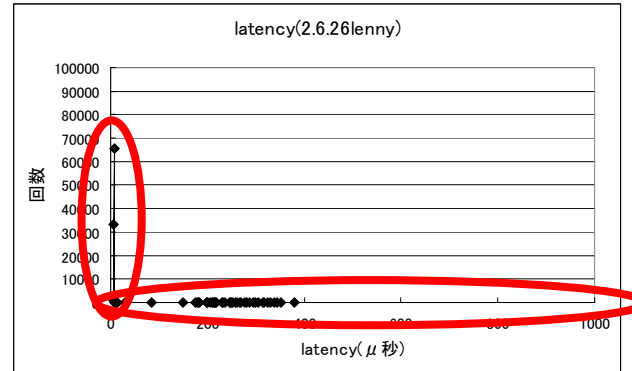
CPU負荷のみ

測定結果(周期1000us/メモリアクセス負荷)

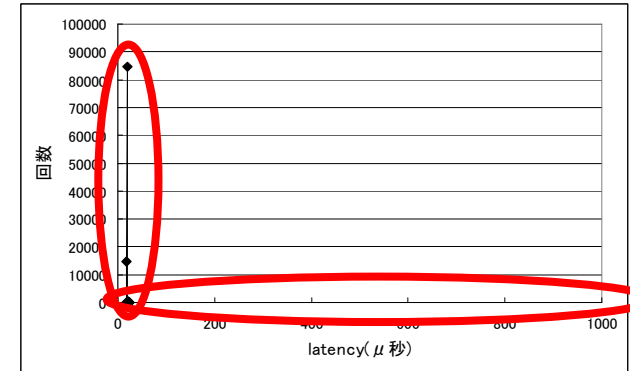
CPU負荷

0%

2.6.31.12

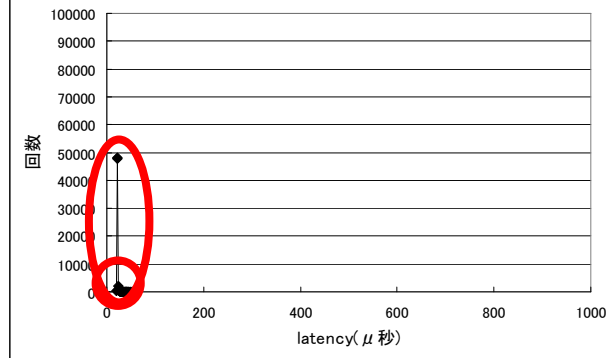
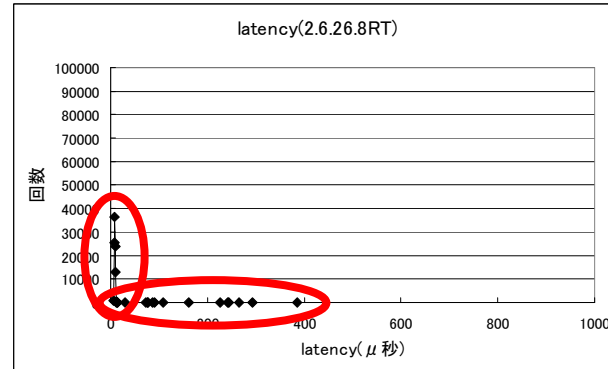


2.6.31.12-RT



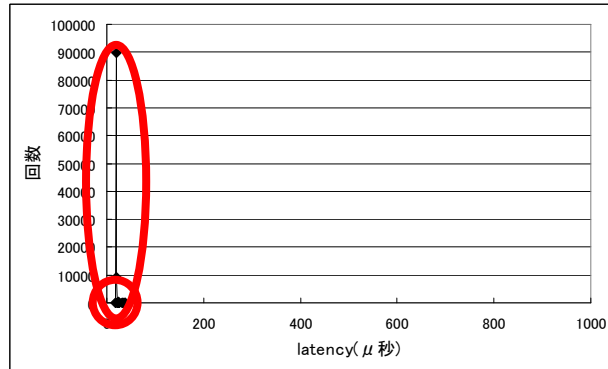
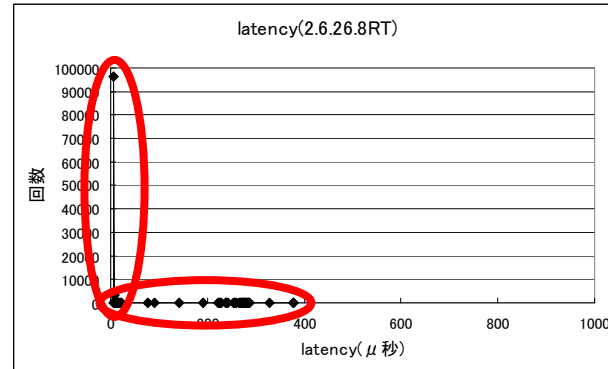
50%

latency(2.6.26.8RT)



100%

latency(2.6.26.8RT)



測定結果(周期1000us/メモリアクセス負荷)

カーネル	CPU負荷率	平均レイテンシ	最大レイテンシ	標準偏差	周期超過回数
2.6.31.12	0%	7.455	27395.703	13.629	1
	50%	8.759	386.848	2.473	0
	100%	6.671	378.242	5.722	0
2.6.31.12-RT	0%	19.361	27207.563	12.593	1
	50%	22.110	56.233	0.784	0
	100%	19.760	38.151	0.466	0

CPU負荷のみ

カーネル	CPU負荷率	平均レイテンシ	最大レイテンシ	標準偏差	周期超過回数
2.6.31.12	0%	7.455	27395.703	13.629	1
	50%	9.881	433.542	3.534	0
	100%	8.122	380.988	6.216	0
2.6.31.12-RT	0%	19.361	27207.563	12.593	1
	50%	24.162	34.459	1.041	0
	100%	39.129	70.020	5.369	0

まとめ

- 全体を通して、VanillaカーネルよりRTカーネルの方が安定
- CPU負荷のみと比べると、メモリ負荷が悪影響となる場合あり
 - ページフォルトによる実メモリ確保のオーバーヘッドが影響している可能性がある
 - キャッシュの影響もあると予想していたが、今回の実験の範囲では大きな差が見られなかった
 - 詳細に調査する必要あり
- 1msの周期タスクで他に何も負荷をかけていない場合に、30msの遅延が生じるケースがある
 - 原因については現在未確認
 - 根本的に違う場所で発生している可能性が高い

TOSHIBA

Leading Innovation >>>