



最終更新日:2006/10/02

キーワード検索:  検索[性能問題チェックシート](#)[性能問題チェックシート](#)[ちゅーとろ](#)[概要](#)  
[ログイン](#)  
[利用手引き](#)

下記にアプリケーションの性能問題をチェックする項目を列挙します。アプリケーションのプロファイラ情報を採取し、各項目をチェックすることで、アプリケーションに性能改善の余地があるかが分かります。  
なお、プロファイラの採取方法は[こちら](#)を参照してください。

性能上の問題箇所が見つければ、[CeNSSチューニングガイド](#)を参照し、プログラムの改善を行ってください。

[CeNSSチューニングガイド](#)

## (1)Sampling Cost

[チューニング事例](#)

## 【チェック内容】

サブルーチンのSampling Costが10%を越えている。

## 【説明】

基本的にチューニングは高コストルーチンから着手します。高コストのルーチンほど、性能を改善することで得られる効果が大きいからです。  
Sampling Costをチェックし、10%以上を占めているサブルーチンについてはチューニングを検討してください。

Sampling Cost					
	Samples	% Run	Barrier	Start	End
1.	683700e+04	<b>35.77</b>	0.000000e+00	18	269 <b>aaa_</b>
6.	117000e+03	<b>12.99</b>	0.000000e+00	80	238 <b>bbb_</b>
3.	474000e+03	7.38	0.000000e+00	20	72 <b>ccc_</b>
2.	138000e+03	4.54	0.000000e+00	111	256 <b>ddd_</b>
1.	894000e+03	4.02	0.000000e+00	269	342 <b>eee_</b>
1.	697000e+03	3.60	0.000000e+00	18	78 <b>fff_</b>

## (2)L2-miss率

## 【チェック内容】

L2-miss率が1%を越えている。

## 【説明】

L2-miss率が1%を越えているサブルーチンは、キャッシュが効果的に利用されていない可能性があります。キャッシュミスに着目してチューニング内容を検討します。

Statistics Performance								
	CPU	Commit	MIPS	MFLOPS	L2-miss	mTLB-ir	mTLB-or	Cover
8.	732334e+02	8.499728e+11	9.733626e+02	9.116430e+01	0.1302	0.0000	0.9809	51.8 <b>aaa_</b>
3.	739155e+02	2.808601e+11	7.511325e+02	2.142955e+02	0.6746	0.0000	0.8679	61.1 <b>bbb_</b>
3.	472051e+02	4.687254e+10	1.349996e+02	1.901587e+01	<b>9.4103</b>	0.0000	0.0009	98.5 <b>ccc_</b>
2.	127186e+02	1.889833e+11	8.884192e+02	2.751608e+02	0.3547	0.0000	0.0000	99.5 <b>ddd_</b>
1.	888756e+02	1.322634e+11	7.002672e+02	1.167205e+02	0.6511	0.0000	0.0000	99.5 <b>eee_</b>
1.	692195e+02	2.704895e+10	1.598454e+02	2.587049e+01	<b>12.0801</b>	0.0000	0.0001	99.5 <b>fff_</b>

## 【改善方法】

具体的な改善方法については、[CeNSSチューニングガイド](#)のp.22「6.1.1.2次キャッシュミスの削減」を参照してください。

## (3)TLB-miss率

## 【チェック内容】

TLB-miss率が0.02%を越えている。

## 【説明】

TLB-miss率が0.02%を越えているサブルーチンは、チューニングの余地があると思われます。アドレス空間へのアクセスに着目してチューニング内容を検討します。

Statistics Performance								
	CPU	Commit	MIPS	MFLOPS	L2-miss	mTLB-ir	mTLB-or	Cover
8.	732334e+02	8.499728e+11	9.733626e+02	9.116430e+01	0.1302	0.0000	<b>0.9809</b>	51.8 <b>aaa_</b>
3.	739155e+02	2.808601e+11	7.511325e+02	2.142955e+02	0.6746	0.0000	<b>0.8679</b>	61.1 <b>bbb_</b>
3.	472051e+02	4.687254e+10	1.349996e+02	1.901587e+01	9.4103	0.0000	<b>0.0009</b>	98.5 <b>ccc_</b>
2.	127186e+02	1.889833e+11	8.884192e+02	2.751608e+02	0.3547	0.0000	0.0000	99.5 <b>ddd_</b>
1.	888756e+02	1.322634e+11	7.002672e+02	1.167205e+02	0.6511	0.0000	0.0000	99.5 <b>eee_</b>
1.	692195e+02	2.704895e+10	1.598454e+02	2.587049e+01	12.0801	0.0000	0.0001	99.5 <b>fff_</b>

## 【改善方法】

具体的な改善方法については、[CeNSSチューニングガイド](#)のp.27「6.1.2.TLBミスの削減」を参照してください。

## (4)MFLOPS値

## 【チェック内容】

MFLOPS値が400以下である。

## 【説明】

MFLOPS値が400を下回っているサブルーチンは、改善の余地があると思われます。

Statistics Performance							
	CPU	Commit	MIPS	MFLOPS	L2-miss	mTLB-ir	mTLB-or

8. 732334e+02	8. 499728e+11	9. 733626e+02	<b>9. 116430e+01</b>	0. 1302	0. 0000	0. 9809	51. 8	<b>aaa_</b>
3. 739155e+02	2. 808601e+11	7. 511325e+02	<b>2. 142955e+02</b>	0. 6746	0. 0000	0. 8679	61. 1	<b>bbb_</b>
3. 472051e+02	4. 687254e+10	1. 349996e+02	<b>1. 901587e+01</b>	9. 4103	0. 0000	0. 0009	98. 5	<b>ccc_</b>
2. 127186e+02	1. 889833e+11	8. 884192e+02	<b>2. 751608e+02</b>	0. 3547	0. 0000	0. 0000	99. 5	<b>ddd_</b>
1. 888756e+02	1. 322634e+11	7. 002672e+02	<b>1. 167205e+02</b>	0. 6511	0. 0000	0. 0000	99. 5	<b>eee_</b>
1. 692195e+02	2. 704895e+10	1. 598454e+02	<b>2. 587049e+01</b>	12. 0801	0. 0000	0. 0001	99. 5	<b>fff_</b>

## 【改善方法】

具体的な改善方法については、[CeNSSチューニングガイド](#) のp.30「6.1.3.その他のスカラーチューニング」を参照してください。

## (5)MIPS値

## 【チェック内容】

MIPS値が1000以下である。

## 【説明】

MIPS値が1000を下回っているサブルーチンは、改善の余地があると思われます。

Statistics Performance								
CPU	Commit	MIPS	MFLOPS	L2-miss	mTLB-ir	mTLB-or	Cover	
8. 732334e+02	8. 499728e+11	<b>9. 733626e+02</b>	9. 116430e+01	0. 1302	0. 0000	0. 9809	51. 8	<b>aaa_</b>
3. 739155e+02	2. 808601e+11	<b>7. 511325e+02</b>	2. 142955e+02	0. 6746	0. 0000	0. 8679	61. 1	<b>bbb_</b>
3. 472051e+02	4. 687254e+10	<b>1. 349996e+02</b>	1. 901587e+01	9. 4103	0. 0000	0. 0009	98. 5	<b>ccc_</b>
2. 127186e+02	1. 889833e+11	<b>8. 884192e+02</b>	2. 751608e+02	0. 3547	0. 0000	0. 0000	99. 5	<b>ddd_</b>
1. 888756e+02	1. 322634e+11	<b>7. 002672e+02</b>	1. 167205e+02	0. 6511	0. 0000	0. 0000	99. 5	<b>eee_</b>
1. 692195e+02	2. 704895e+10	<b>1. 598454e+02</b>	2. 587049e+01	12. 0801	0. 0000	0. 0001	99. 5	<b>fff_</b>

## 【改善方法】

具体的な改善方法については、[CeNSSチューニングガイド](#) のp.30「6.1.3.その他のスカラーチューニング」を参照してください。

## (6)網羅率(Cover)

## 【チェック内容】

網羅率が95%以下である。

## 【説明】

網羅率が95%を下回っているサブルーチンは改善の余地があると思われます。

Statistics Performance								
CPU	Commit	MIPS	MFLOPS	L2-miss	mTLB-ir	mTLB-or	Cover	
8. 732334e+02	8. 499728e+11	9. 733626e+02	9. 116430e+01	0. 1302	0. 0000	0. 9809	<b>51. 8</b>	<b>aaa_</b>
3. 739155e+02	2. 808601e+11	7. 511325e+02	2. 142955e+02	0. 6746	0. 0000	0. 8679	<b>61. 1</b>	<b>bbb_</b>
3. 472051e+02	4. 687254e+10	1. 349996e+02	1. 901587e+01	9. 4103	0. 0000	0. 0009	98. 5	ccc_
2. 127186e+02	1. 889833e+11	8. 884192e+02	2. 751608e+02	0. 3547	0. 0000	0. 0000	99. 5	ddd_
1. 888756e+02	1. 322634e+11	7. 002672e+02	1. 167205e+02	0. 6511	0. 0000	0. 0000	99. 5	eee_
1. 692195e+02	2. 704895e+10	1. 598454e+02	2. 587049e+01	12. 0801	0. 0000	0. 0001	99. 5	fff_

## 【補足】

網羅率 = CPU時間/経過時間\*100

## 【改善方法】

網羅率が低下する原因には、次の項目が考えられます。

- ・データTLBミス率が高い。
  - ・プログラムの実行時に入出力処理が多い。
  - ・プログラムの実行時に領域の確保、解放処理が多い。
  - ・スレッド並列プログラムの場合、並列化される範囲が少ない。または、スレッド間の同期待ち時間が長い。
- これらを改善改善することで網羅率が改善する可能性があります。

## (7)(プロセス/スレッド)バランス

## 【チェック内容】

プロセス(もしくはスレッド)並列バランスが悪い。

## 【説明】

プロセス(もしくはスレッド)並列バランスが悪いサブルーチンは改善の余地があると思われます。

- 0 +

*****	*****	+583%	3. 394000e+03	Process	0
*****	*****	- 88%	6. 000000e+01	Process	1
*****	*****	- 84%	8. 100000e+01	Process	2
*****	*****	- 67%	1. 620000e+02	Process	3
*****	*****	- 79%	1. 050000e+02	Process	4
*****	*****	- 47%	2. 640000e+02	Process	5
*****	*****	- 82%	8. 800000e+01	Process	6
*****	*****	- 20%	3. 990000e+02	Process	7
*****	*****	- 80%	9. 700000e+01	Process	8
*****	**	+ 7%	5. 290000e+02	Process	9
*****	*****	- 78%	1. 070000e+02	Process	10
*****	*****	+ 36%	6. 740000e+02	Process	11

Balance against average time per Process

## 【改善方法】

具体的な改善方法については、[CeNSSチューニングガイド](#) の p.37「6.2.4.ロードバランスの改善」もしくは p.39「6.3.2.ロードバランスの改善」を参照してください。

## (8)Invalid値

## 【チェック内容】

Invalid値が高い。

## 【説明】

Invalid値が高い場合、FALSE SHARINGが発生している可能性があります。

Cache Overlap					
CPU	Commit	Invalid	WriteBack	Cover	
1.609896e+02	1.127468e+11	<b>1.4074</b>	0.1196	98.3	aaa_
1.562683e+02	1.448637e+11	0.0000	0.2737	98.0	bbb_
1.106383e+02	1.070335e+11	0.0000	0.1814	98.3	ccc_
1.048984e+02	8.034469e+10	0.0000	0.4074	97.9	ddd_
9.838431e+01	1.591080e+10	0.0001	3.5347	98.0	eee_

## 【改善方法】

具体的な改善方法については、[CeNSSチューニングガイド](#) のp.18「5.3.並列性能の分析手法」を参照してください。

## (9)WriteBack値

## 【チェック内容】

WriteBack値が高い。

## 【説明】

この値が高い場合、キャッシュメモリーの使用が偏っている可能性があります。

Cache Overlap					
CPU	Commit	Invalid	WriteBack	Cover	
1.609896e+02	1.127468e+11	1.4074	0.1196	98.3	aaa_
1.562683e+02	1.448637e+11	0.0000	0.2737	98.0	bbb_
1.106383e+02	1.070335e+11	0.0000	0.1814	98.3	ccc_
1.048984e+02	8.034469e+10	0.0000	0.4074	97.9	ddd_
9.838431e+01	1.591080e+10	0.0001	<b>3.5347</b>	98.0	eee_

## 【改善方法】

具体的な改善方法については、[CeNSSチューニングガイド](#) のp.16「5.2.スカラー性能の分析手法」を参照してください。

## (10)スレッド非並列化部分

## 【チェック内容】

非並列処理部分がある。

## 【説明】

スレッド並列プログラムにおいて、スレッド並列実行されていない場合、並列実行の阻害要因を取り除くことで性能向上する可能性があります。

Sampling Cost					
Samples	% Run	Barrier	Start	End	
7.784000e+03	6.51	7.000000e+00	468	639	aaa_
5.960000e+03	4.99	0.000000e+00	2	247	MAIN_
5.869000e+03	4.91	1.100000e+01	4	466	bbb_
5.689000e+03	4.76	1.000000e+00	2	169	ccc_
5.069000e+03	4.24	3.000000e+00	341	653	ddd_
3.816000e+03	3.19	0.000000e+00	722	764	eee_ _PRL_2_
3.720000e+03	3.11	1.700000e+01	488	796	fff_
3.663000e+03	3.07	0.000000e+00	230	274	ggg_ _PRL_10_
3.482000e+03	2.91	0.000000e+00	418	462	hhh_ _PRL_2_

## 【補足】

\_PRL\_X\_がについていないサブルーチンは逐次実行を表します。

## 【改善方法】

コンパイルオプションに -Qp を指定してコンパイルします。すると、コンパイル時にプログラムリスト(list)が出力されるようになります。プログラムリストには、最適化情報が出力されており、並列化が阻害されている要因がわかります。他にもインライン展開の状況や、アンローリングの展開数がわかります。

## プロファイラ情報の採取手順

## ①翻訳

コンパイル時に、-Ktl.trtのオプションを追加します。

```
f90ns -Umpi -Ktl.trt sample.f90 -o sample.exe
```

## ②実行

以下の環境変数を設定してジョブを実行します。

```
TRT_ENV="PMP=on" : export TRT_ENV
PROF_STATS=7 : export PROF_STATS
PROF_PA="sta,cov" : export PROF_PA
PROF_USERFUNC=1 : export PROF_USERFUNC
```

## ③解析

①②の手順でジョブを実行すると、カレントディレクトリに以下の名前のファイルが作成されます。

プロセス並列数=nnnのジョブを実行した場合

```
GProf_XXXXX.prof.pri
```

```
DProf_XXXXX.000.prof.pri ~ DProf_XXXXX.nnn.prof.pri
```

これらのファイルを、mppprofコマンドにより解析します。

以下にmppprofコマンドの実行例を示します。

```
mppprof GProf_12345.prof.pri > GProf_12345.txt
```

※注意

プロファイラ情報を取得するためにジョブを実行する場合、通常よりも実効時間が増大します。

## ④スレッド並列ジョブに関して

qsub時に、以下の点に注意する必要があります。

■スレッド並列数指定の環境変数を設定する。

- ・自動並列: PARALLEL環境変数
- ・OpenMP: PARALLELおよび、OMP\_NUM\_THREADS環境変数
- qsubオプションipIには演算スレッド並列数+1を指定する。
- ・プロファイラ採取のために、CPUを1つ使用する。

例1: 自動並列の場合

```
#00$-q QJOB
#00$-r Sample00.01-01
#00$-IP 4
#00$-lp 3 ←☆PARALLEL環境変数+1を指定
#00$
:
PARALLEL=2; export PARALLEL
:
mpiexec -n 4 Sample.out
```

例2: OpenMPの場合

```
#00$-q QJOB
#00$-r Sample00.01-01
#00$-IP 4
#00$-lp 3 ←☆PARALLEL環境変数+1を指定
#00$
:
PARALLEL=2; export PARALLEL
OMP_NUM_THREADS=2; export OMP_NUM_THREADS
:
mpiexec -n 4 Sample.out
```

[お問い合わせ](#)

Copyright©2004 JAXA