

■性能テスト結果

本資料は、VaR証拠金計算ソフトウェアの処理時間を記載したものです。
処理時間は環境構築の検討にご利用いただくために開示するものですが、
利用環境や利用方法等により異なりますので、あくまで目安としてお取扱いください。

- 1.APIアプリケーション（追記箇所あり）
- 2.APIアプリケーション（複数リクエスト）
- 3.CLIアプリケーション
- 4.GUIアプリケーション

変更履歴

項番	変更日付	バージョン	シート	変更内容
1	2022/11/1	1.0		新規作成
2	2023/1/31	1.01	概要	目次に新規シートを追加 文言削除
3			1.APIアプリケーション	異なるマシンでの検証結果を追記
4			3.CLIアプリケーション 4.GUIアプリケーション	シートの追加
5	2024/1/30	2.0	1.APIアプリケーション 2.APIアプリケーション（複数リクエスト） 3.CLIアプリケーション 4.GUIアプリケーション	すべての計測結果を更新
6	2024/12/9	3.0	1.APIアプリケーション 2.APIアプリケーション（複数リクエスト） 3.CLIアプリケーション 4.GUIアプリケーション	すべての計測結果を更新
7	2025/11/28	4.0	1.APIアプリケーション 2.APIアプリケーション（複数リクエスト） 3.CLIアプリケーション 4.GUIアプリケーション	すべての計測結果を更新
8			4.GUIアプリケーション	GUIの測定に利用しているOSをWindows10からWindows11に変更

1.APIアプリケーション

測定内容

本編「2.5.1 入力ファイルの読み込み」および「2.5.2 計算」のAPIリクエストについて処理時間を計測しております。

「入力ファイルの読み込み」の測定結果は、1つのBPFに収録される銘柄数を「設定値」に設定したBPF1ファイル、VPFおよびAPFの計3ファイルの読み込みにかかる時間です。

計算処理の測定結果は、1つのリクエストに測定結果の表のとおり口座およびポジションを設定したリクエストを処理するのに必要な処理時間です。

測定環境

以下のAWS上の仮想マシンで確認を行いました。

A Amazon EC2 インスタンス：m5d.12xlarge CPU：48Cores 2.5GHz メモリ：192GiB SSD OS：Red Hat Enterprise Linux Server release 7.7 (Maipo) Java：openjdk version "1.8.0_332"

B Amazon EC2 インスタンス：m5ad.xlarge CPU：4Cores 2.5GHz メモリ：16GiB SSD OS：Red Hat Enterprise Linux Server release 7.7 (Maipo) Java：openjdk version "1.8.0_332"

以下のパラメータ設定で計測しました。（処理時間に影響しないパラメータは割愛）

```
pool.size.max = pool.size.core
pool.size.queue = 0 (* スレッドに割り当てられなかった計算処理はすべてキューに送信され待機中となります)
pool.threshold = 1 (* 口座単位で計算処理がスレッドに割り当てられて処理されます)
server.tomcat.threads.max = 200
server.tomcat.threads.min-spare = 10
server.tomcat.accept-count = 100
```

測定結果

入力ファイルの読み込み (単位：ms)

#	項目	設定値	マシンA (Core:48)	マシンB (Core:4)
			pool.size.core=1	
1	BPF対象銘柄	1,000	270	652
2	BPF対象銘柄	7,000	1,591	5,123
3	BPF対象銘柄	15,000	3,308	11,917
4	BPF対象銘柄	30,000	6,514	24,280
5	BPF対象銘柄	50,000	10,045	—

計算

(単位：ms)

#	項目	設定値	マシンA (Core:48)			
			pool.size.core=1	pool.size.core=5	pool.size.core=20	pool.size.core=50
			OutputType:SAM	OutputType:SAM	OutputType:SAM	OutputType:SAM
1	□座数	5,000	4,521	1,287	743	1057
	1□座あたりの平均ポジション数	2				
2	□座数	10,000	8,916	2,516	1,414	2,059
	1□座あたりの平均ポジション数	2				
3	□座数	25,000	22,215	6,276	3,324	5,029
	1□座あたりの平均ポジション数	2				
4	□座数	50,000	44,115	13,130	7,004	10,574
	1□座あたりの平均ポジション数	2				
5	□座数	10	358	177	159	146
	1□座あたりの平均ポジション数	100				
6	□座数	50	237	121	102	105
	1□座あたりの平均ポジション数	10				
7	□座数	500	731	235	159	181
	1□座あたりの平均ポジション数	3				
8	□座数	1,000	950	317	202	243
	1□座あたりの平均ポジション数	2				
9	□座数	2,000	2,654	791	469	523
	1□座あたりの平均ポジション数	3				
10	□座数	4,000	8,555	2,445	1,338	1,415
	1□座あたりの平均ポジション数	5				

(単位：ms)

#	項目	設定値	マシンB (Core:4)	
			pool.size.core=1	pool.size.core=5
			OutputType:SAM	OutputType:SAM
1	口座数	5,000	6,988	3,386
	1口座あたりの平均ポジション数	2		
2	口座数	10	614	408
	1口座あたりの平均ポジション数	100		
3	口座数	1,000	1,630	911
	1口座あたりの平均ポジション数	2		

2.APIアプリケーション（複数リクエスト）

実施内容

測定結果の表上の「設定値」で定義された数の本編「2.5.2 計算」のAPIリクエストをAPIアプリケーションで同時に処理し、これらのリクエストの処理時間の平均を測定しております。

計算対象（Output Type）はVaR Marginです。

1つのリクエストについて1口座のみを収録しております。

また、1つの口座に収録されるポジション数は50です。

測定環境

以下2台のマシンで確認を行いました。

A: Amazon EC2 インスタンス：m5d.12xlarge CPU：48Cores 2.5GHz メモリ：192GiB SSD OS：Red Hat Enterprise Linux Server release 7.7 (Maipo) Java：openjdk version "1.8.0_332"

B: Amazon EC2 インスタンス：m5ad.xlarge CPU：4Cores 2.2GHz メモリ：16GiB SSD OS：Red Hat Enterprise Linux Server release 7.7 (Maipo) Java：openjdk version "1.8.0_332"

設定

以下のパラメータ設定で計測しました。（処理時間に影響しないパラメータは割愛）

pool.threshold = 1 (* すべてのリクエストをマルチスレッドで処理する）（tomcatのスレッドの影響のみを計測）

server.tomcat.threads.min-spare =server.tomcat.threads.max

測定結果

(単位：TPS)

#	項目	設定値	マシンA (Core:48)			
			server.tomcat.threads.max=1	server.tomcat.threads.max=5	server.tomcat.threads.max=20	server.tomcat.threads.max=50
			OutputType:VaRMargin	OutputType:VaRMargin	OutputType:VaRMargin	OutputType:VaRMargin
1	同時リクエスト数	1	64	64	64	65
2		3	72	184	187	187
3		10	72	342	593	602
4		50	72	340	1178	1436
5		100	70	344	1176	1429
6		1,000	71	337	1170	1436

*例えば、server.tomcat.threads.max=5 Setting=100の場合、TPS=**344**となっていますが、1秒間に**344**件のリクエストを処理したという意味であり、合計でかかる時間は100/344=**290**(ms) です。

*なお、サーバのCore数よりもserver.tomcat.threads.maxを大きくした場合は、性能の大幅な向上が見込めないことを確認しております。

(単位：TPS)

#	項目	設定値	マシンB (Core:4)	
			server.tomcat.threads.max=1	server.tomcat.threads.max=5
			OutputType:VaRMargin	OutputType:VaRMargin
1	同時リクエスト数	1	45	43
2		3	48	100
3		10	50	122
4		50	46	128
5		100	47	134

3.CLIアプリケーション

測定内容

本編「2.2.2 計算の実行」に記載の手順で計算を行った際の処理時間を計測しております。
パラメータファイルの読み込み時間と、測定結果の表のとおり口座およびポジションを設定したリクエストを処理するのに必要な処理時間とを記載しております。

測定環境

- 以下のAWS上の仮想マシンで確認を行いました。
- A Amazon EC2 インスタンス：m5d.12xlarge CPU：48Cores 2.5GHz メモリ：192GiB SSD OS：Red Hat Enterprise Linux Server release 7.7 (Maipo) Java：openjdk version "1.8.0_332"
 - B Amazon EC2 インスタンス：m5ad.xlarge CPU：4Cores 2.5GHz メモリ：16GiB SSD OS：Red Hat Enterprise Linux Server release 7.7 (Maipo) Java：openjdk version "1.8.0_332"

以下のパラメータ設定で計測しました。（処理時間に影響しないパラメータは割愛）

pool.size.max = pool.size.core (* スレッド拡張にかかる時間は計測せず)
pool.size.queue = 0 (* リクエストのタイムアウトなし)
pool.threshold = 1 (* すべてのリクエストをマルチスレッドで処理する)

測定結果

計算

(単位：ms)

#	項目	設定値	内訳	マシンA (Core:48)			
				pool.size.core=1	pool.size.core=5	pool.size.core=20	pool.size.core=50
				OutputType:SAM	OutputType:SAM	OutputType:SAM	OutputType:SAM
1	BPF対象銘柄	7,000	合計	3,959	3,408	3,360	3,425
	口座数	1,000	ファイル読込	1,753	1,784	1,807	1,781
	1口座あたりの平均ポジション数	2	証拠金計算	1,199	588	527	640
2	BPF対象銘柄	7,000	合計	5,161	3,991	3,808	3,999
	口座数	2,000	ファイル読込	2,048	2,061	2,035	2,058
	1口座あたりの平均ポジション数	2	証拠金計算	2,092	921	742	914
3	BPF対象銘柄	7,000	合計	8,487	5,560	5,021	5,768
	口座数	5,000	ファイル読込	2,417	2,449	2,393	2,402
	1口座あたりの平均ポジション数	2	証拠金計算	5,055	2,087	1,616	2,341
4	BPF対象銘柄	7,000	合計	12,861	6,913	5,998	7,244
	口座数	10,000	ファイル読込	2,591	2,578	2,632	2,574
	1口座あたりの平均ポジション数	2	証拠金計算	9,227	3,343	2,349	3,652
5	BPF対象銘柄	30,000	合計	5,853	5,901	5,943	5,846
	口座数	10	ファイル読込	4,324	4,403	4,407	4,335
	1口座あたりの平均ポジション数	100	証拠金計算	484	487	515	504
6	BPF対象銘柄	30,000	合計	5,591	5,645	5,471	5,649
	口座数	50	ファイル読込	4,190	4,303	4,151	4,352
	1口座あたりの平均ポジション数	10	証拠金計算	388	320	292	265
7	BPF対象銘柄	30,000	合計	6,639	6,048	5,987	6,242
	口座数	500	ファイル読込	4,674	4,485	4,478	4,470
	1口座あたりの平均ポジション数	3	証拠金計算	939	520	486	753
8	BPF対象銘柄	1,000	合計	2,388	2,037	2,026	2,102
	口座数	500	ファイル読込	601	589	594	585
	1口座あたりの平均ポジション数	2	証拠金計算	756	427	406	478
9	BPF対象銘柄	1,000	合計	3,388	2,449	2,390	2,572
	口座数	1,000	ファイル読込	635	627	622	645
	1口座あたりの平均ポジション数	3	証拠金計算	1,726	798	760	910
10	BPF対象銘柄	1,000	合計	6,532	3,677	3,263	3,325
	口座数	2,000	ファイル読込	750	757	760	728
	1口座あたりの平均ポジション数	5	証拠金計算	4,748	1,899	1,480	1,589

*ファイル読込は1つのコアで実施されるため、pool.size.coreを変えても処理時間は変動しません。

*なお、サーバのCore数よりもserver.tomcat.threads.maxを大きくした場合は、性能の大幅な向上が見込めないことを確認しております。

(単位：ms)

#	項目	設定値	内訳	マシンB (Core:4)	
				pool.size.core=1	pool.size.core=5
				OutputType:SAM	OutputType:SAM
1	BPF対象銘柄	7,000	合計	9,374	8,597
	口座数	1,000	ファイル読込	4,975	4,932
	1口座あたりの平均ポジション数	2	証拠金計算	2,436	1,724
2	BPF対象銘柄	30,000	合計	15,129	14,725
	口座数	10	ファイル読込	12,253	12,007
	1口座あたりの平均ポジション数	100	証拠金計算	924	757
3	BPF対象銘柄	1,000	合計	5,163	4,778
	口座数	500	ファイル読込	1,581	1,508
	1口座あたりの平均ポジション数	2	証拠金計算	1,643	1,297

4.GUIアプリケーション

測定内容

本編「2.2.3 計算の実行」に記載の手順で計算を行った際の処理時間を計測しております。

パラメータファイルの読み込み時間と、測定結果の表のとおり口座およびポジションを設定したリクエストを処理するのに必要な処理時間とを記載しております。

測定環境

以下の物理マシンで確認を行いました。

C 物理端末 (Dell Latitude3520) CPU : Core (TM) i7-1165G7 メモリ : 16GB SSD OS : Windows Client 11 (24H2) Java : 11.0.21 (Corretto)

以下のパラメータ設定で計測しました。(処理時間に影響しないパラメータは割愛)

pool.size.max = pool.size.core (* スレッド拡張にかかる時間は計測せず)

pool.size.queue = 0 (* リクエストのタイムアウトなし)

pool.threshold = 1 (* すべてのリクエストをマルチスレッドで処理する)

測定結果
計算

#	項目	設定値	内訳	マシンC (Core:4)		
				pool.size.core=1	pool.size.core=4	pool.size.core=8
				OutputType:SAM	OutputType:SAM	OutputType:SAM
1	BPF対象銘柄	7,000	合計	4,073	4,202	3,947
	口座数	100	ファイル読込	2,950	3,134	2,895
	1口座あたりの平均ポジション数	5	証拠金計算	641	575	586
2	BPF対象銘柄	7,000	合計	4,708	4,713	4,510
	口座数	200	ファイル読込	3,034	3,221	2,994
	1口座あたりの平均ポジション数	5	証拠金計算	1,010	740	811
3	BPF対象銘柄	30,000	合計	15,810	15,662	14,906
	口座数	100	ファイル読込	11,934	12,839	12,148
	1口座あたりの平均ポジション数	50	証拠金計算	2,977	1,896	1,827
4	BPF対象銘柄	30,000	合計	24,960	20,184	19,991
	口座数	200	ファイル読込	15,426	15,397	14,906
	1口座あたりの平均ポジション数	50	証拠金計算	8,053	3,433	3,671
5	BPF対象銘柄	1,000	合計	3,204	2,900	3,319
	口座数	100	ファイル読込	1,081	1,084	1,170
	1口座あたりの平均ポジション数	20	証拠金計算	1,478	1,143	1,417
6	BPF対象銘柄	1,000	合計	4,636	3,473	3,794
	口座数	200	ファイル読込	1,086	1,039	1,134
	1口座あたりの平均ポジション数	20	証拠金計算	2,529	1,506	1,650