

■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	すべての測定結果を更新

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 = 2 (* 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	892	1,333
2	BPF対象銘柄	7,000	6,319	10,073
3	BPF対象銘柄	15,000	13,433	23,124
4	BPF対象銘柄	30,000	26,791	46,940
5	BPF対象銘柄	50,000	80,400	—

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:ALL	OutputType:ALL	OutputType:ALL	OutputType:ALL
1	口座数	5,000	3,639	1,094	642	814
	1口座あたりの平均ポジション数	2				
2	口座数	10,000	7,189	2,109	1,187	1,546
	1口座あたりの平均ポジション数	2				
3	口座数	25,000	17,848	5,257	2,855	3,796
	1口座あたりの平均ポジション数	2				
4	口座数	50,000	35,928	10,839	6,275	7,778
	1口座あたりの平均ポジション数	2				
5	口座数	10	305	175	135	132
	1口座あたりの平均ポジション数	100				
6	口座数	50	205	104	98	106
	1口座あたりの平均ポジション数	10				
7	口座数	500	594	213	154	169
	1口座あたりの平均ポジション数	3				
8	口座数	1,000	801	258	179	214
	1口座あたりの平均ポジション数	2				
9	口座数	2,000	2,174	673	408	452
	1口座あたりの平均ポジション数	3				
10	口座数	4,000	6,957	2,073	1,167	1,229
	1口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:ALL	OutputType:ALL
1	口座数	5,000	5,585	2,821
	1口座あたりの平均ポジション数	2		
2	口座数	10	488	333
	1口座あたりの平均ポジション数	100		
3	口座数	1,000	1,319	832
	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 = 2 (* 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	81	81	79	78
2		3	90	237	232	234
3		10	90	432	741	749
4		50	90	426	1297	1,782
5		100	90	430	1324	1,723
6		1,000	90	433	1321	1,721

*例えば、server.tomcat.threads.max=5 Setting=100の場合、TPS=395となっていますが、1秒間に395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 $100/395=253(\text{ms})$ 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	51	51
2		3	59	127
3		10	60	158
4		50	61	156
5		100	59	157

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	マシンA (Core:48)			
			pool.size.core=1	pool.size.core=5	pool.size.core=20	pool.size.core=50
			OutputType:ALL	OutputType:ALL	OutputType:ALL	OutputType:ALL
1	BPF対象銘柄	7,000	合計：7,994	合計：7,479	合計：7,450	合計：7,538
	口座数	1,000	ファイル読込：5,833	ファイル読込：5,798	ファイル読込：5,814	ファイル読込：5,789
	1口座あたりの平均ポジション数	2	証拠金計算：1,142	証拠金計算：685	証拠金計算：635	証拠金計算：769
2	BPF対象銘柄	7,000	合計：9,512	合計：8,551	合計：8,502	合計：8,613
	口座数	2,000	ファイル読込：6,520	ファイル読込：6,492	ファイル読込：6,488	ファイル読込：6,519
	1口座あたりの平均ポジション数	2	証拠金計算：1,954	証拠金計算：1,042	証拠金計算：992	証拠金計算：1,064
3	BPF対象銘柄	7,000	合計：12,788	合計：10,370	合計：10,044	合計：10,244
	口座数	5,000	ファイル読込：7,507	ファイル読込：7,484	ファイル読込：7,514	ファイル読込：7,510
	1口座あたりの平均ポジション数	2	証拠金計算：4,277	証拠金計算：1,879	証拠金計算：1,526	証拠金計算：1,740
4	BPF対象銘柄	7,000	合計：16,502	合計：11,762	合計：11,090	合計：11,451
	口座数	10,000	ファイル読込：7,735	ファイル読込：7,732	ファイル読込：7,793	ファイル読込：7,761
	1口座あたりの平均ポジション数	2	証拠金計算：7,774	証拠金計算：3,004	証拠金計算：2,285	証拠金計算：2,683
5	BPF対象銘柄	30,000	合計：21,360	合計：21,587	合計：21,606	合計：21,624
	口座数	10	ファイル読込：19,943	ファイル読込：20,066	ファイル読込：20,027	ファイル読込：20,083
	1口座あたりの平均ポジション数	100	証拠金計算：404	証拠金計算：507	証拠金計算：541	証拠金計算：556
6	BPF対象銘柄	30,000	合計：21,351	合計：21,230	合計：21,250	合計：21,246
	口座数	50	ファイル読込：20,003	ファイル読込：19,926	ファイル読込：19,938	ファイル読込：19,984
	1口座あたりの平均ポジション数	10	証拠金計算：341	証拠金計算：310	証拠金計算：306	証拠金計算：267
7	BPF対象銘柄	30,000	合計：22,367	合計：21,926	合計：22,012	合計：22,135
	口座数	500	ファイル読込：20,528	ファイル読込：20,452	ファイル読込：20,528	ファイル読込：20,547
	1口座あたりの平均ポジション数	3	証拠金計算：820	証拠金計算：483	証拠金計算：486	証拠金計算：596
8	BPF対象銘柄	1,000	合計：2,972	合計：2,688	合計：2,720	合計：2,806
	口座数	500	ファイル読込：1,288	ファイル読込：1,288	ファイル読込：1,306	ファイル読込：1,285
	1口座あたりの平均ポジション数	2	証拠金計算：686	証拠金計算：410	証拠金計算：397	証拠金計算：515
9	BPF対象銘柄	1,000	合計：3,776	合計：3,053	合計：3,007	合計：3,277
	口座数	1,000	ファイル読込：1,319	ファイル読込：1,327	ファイル読込：1,324	ファイル読込：1,347
	1口座あたりの平均ポジション数	3	証拠金計算：1,475	証拠金計算：730	証拠金計算：692	証拠金計算：915
10	BPF対象銘柄	1,000	合計：6,495	合計：4,094	合計：3,858	合計：4,026
	口座数	2,000	ファイル読込：1,440	ファイル読込：1,425	ファイル読込：1,422	ファイル読込：1,421
	1口座あたりの平均ポジション数	5	証拠金計算：4,052	証拠金計算：1,675	証拠金計算：1,427	証拠金計算：1,610

*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	マシンB (Core:4)	
			pool.size.core=1	pool.size.core=5
			OutputType:ALL	OutputType:ALL
1	BPF対象銘柄	7,000	合計：12,344	合計：11,909
	口座数	1,000	ファイル読込：8,275	ファイル読込：8,314
	1口座あたりの平均ポジション数	2	証拠金計算：2,264	証拠金計算：1,781
2	BPF対象銘柄	30,000	合計：30,390	合計：30,767
	口座数	10	ファイル読込：27,812	ファイル読込：28,314
	1口座あたりの平均ポジション数	100	証拠金計算：777	証拠金計算：669
3	BPF対象銘柄	1,000	合計：5,510	合計：5,220
	口座数	500	ファイル読込：2,366	ファイル読込：2,292
	1口座あたりの平均ポジション数	2	証拠金計算：1,319	証拠金計算：1,112

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 10 (22H2) 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

(単位：ms)

#	項目	設定値	マシンC (Core:4)		
			pool.size.core=1	pool.size.core=4	pool.size.core=8
			OutputType:ALL	OutputType:ALL	OutputType:ALL
1	BPF対象銘柄	7,000	合計：9,282	合計：9,739	合計：9,243
	口座数	100	ファイル読込：8,393	ファイル読込：8,867	ファイル読込：8,405
	1口座あたりの平均ポジション数	5	証拠金計算：512	証拠金計算：433	証拠金計算：440
2	BPF対象銘柄	7,000	合計：10,007	合計：10,548	合計：9,902
	口座数	200	ファイル読込：8,688	ファイル読込：9,251	ファイル読込：8,701
	1口座あたりの平均ポジション数	5	証拠金計算：734	証拠金計算：601	証拠金計算：596
3	BPF対象銘柄	30,000	合計：36,870	合計：36,067	合計：36,195
	口座数	100	ファイル読込：33,636	ファイル読込：33,611	ファイル読込：33,752
	1口座あたりの平均ポジション数	50	証拠金計算：2,410	証拠金計算：1,676	証拠金計算：1,639
4	BPF対象銘柄	30,000	合計：45,893	合計：42,082	合計：42,020
	口座数	200	ファイル読込：38,679	ファイル読込：38,416	ファイル読込：37,892
	1口座あたりの平均ポジション数	50	証拠金計算：6,081	証拠金計算：2,551	証拠金計算：3,013
5	BPF対象銘柄	1,000	合計：3,936	合計：3,733	合計：3,878
	口座数	100	ファイル読込：2,252	ファイル読込：2,232	ファイル読込：2,247
	1口座あたりの平均ポジション数	20	証拠金計算：1,149	証拠金計算：953	証拠金計算：1,010
6	BPF対象銘柄	1,000	合計：4,877	合計：4,293	合計：4,380
	口座数	200	ファイル読込：2,247	ファイル読込：2,233	ファイル読込：2,171
	1口座あたりの平均ポジション数	20	証拠金計算：1,884	証拠金計算：1,278	証拠金計算：1,411