

# 性能トラブル解決の手引き-事例偏

性能トラブルが発生すると、原因究明までに時間がかかることが多い。また解決までの時間は、担当者の経験や知識、技能によって大きく異なる。

この問題を解決するために、性能トラブルが発生した時の原因調査の手順と観点を表形式でまとめることにした。

また、各現象ごとに5W1H的な観点から分類も付加した。

どこまで事例としてお役に立てるかわかりませんが、同様な悩みを持つ方の参考になれば幸いです。

UNIXは、The Open Groupの米国ならびに他の国における登録商標です。

Windowsは、米国Microsoft Corporationの米国およびその他の国における登録商標または商標です。

2015.10.13

(株)日立ソリューションズ

鈴木 勝彦

性能トラブル解決の手引き-事例編

(株)日立ソリューションズ

鈴木 勝彦

2015.10.12

・具体的な現象と調査ポイント並びに分類

現象	調査ポイント1	確認結果1	調査ポイント2	確認結果2	原因	調査場所大分類	調査場所中分類	調査場所小分類	観点	調査観点	備考	誰が(Who)		どうした(How)		目的語(What)	いつ(When)	誰に(Whom)	どこで(Where)				
												調査するプロセス(ソース)の特定		原因となる動作の特定						原因となる資源の特定	原因となるきっかけ	原因を誘発したものの	原因となる場所
												大項目	中項目	大項目	中項目								
1 処理が終わらない(1時間経過しても何も変わらない)	CPU使用率の確認	該当プロセスのCPU使用率が0%のままで推移。	(1.1)トレースログでWAIT場所を特定する。 または、 (1.2)プロセスのスナップダンプを採取し、PCの値を確認する。 かつ (2.1)ソースからタイマー値の確認 または、	(1)PCの場所がタイマー設定後のWAITである。 (2)タイマー値の値が大きい	UPのタイマー設定する時間を誤って長時間の値を設定した。	ソース	WAIT処理のパラメタ		WAIT処理	・スナップダンプによる解析(WAIT箇所を見つけ、タイマーの値を確認する)		自プロセス	①自分のプロセスで自分のソースの範囲内	WAIT	タイマー値大	時間	該当せず	自プロセス	自分のソース				
2 処理が終わらない(1時間経過しても何も変わらない)	CPU使用率の確認	該当プロセスのCPU使用率が0%のままで推移。	(1.1)トレースログでWAIT場所を特定する。 または、 (1.2)プロセスのスナップダンプを採取し、PCの値を確認する。 かつ (2.1)ソースからタイマー値の確認 または、	(1)PCの場所がタイマー設定後のWAITである。 (2)タイマー値の値が大きい	システムコールエラー時のリトライ時に設定するタイマーが長すぎる。	ソース	WAIT処理のパラメタ		WAIT処理	・スナップダンプによる解析(WAIT箇所を見つけ、タイマーの値を確認する)		自プロセス	①自分のプロセスで自分のソースの範囲内	WAIT	タイマー値大(リトライ)	時間	システムコールエラー	自プロセス	自分のソース				
3 処理が終わらない(1時間経過しても何も変わらない)	CPU使用率の確認	該当プロセスのCPU使用率が0%のままで推移。	(1.1)トレースログでWAIT場所を特定する。 または、 (1.2)プロセスのスナップダンプを採取し、PCの値を確認する。 かつ (2.1)ソースからタイマー値の確認 または、	(1)PCの場所がタイマー設定後のWAITである。 (2)無限待ちをしている。	システムコールエラー時に無限待ちをしているが、通信障害によって戻らない状況である。	ソース	WAIT処理のパラメタ		WAIT処理	・スナップダンプによる解析(WAIT箇所を見つけ、無限待ちかを確認する)		自プロセス	①自分のプロセスで自分のソースの範囲内	WAIT	無限待ち	時間	通信エラー	該当せず	ハード障害				
4 処理が終わらない(1時間経過しても何も変わらない)	CPU使用率の確認	該当プロセスのCPU使用率が0%のままで推移。	(1.1)トレースログでWAIT場所を特定する。 または、 (1.2)プロセスのスナップダンプを採取し、PCの値を確認する。 かつ (2.1)ソースからタイマー値の確認 または、	(1)PCの場所がタイマー設定後のWAITである。 (2)無限待ちをしている。	システムコールエラー時に無限待ちをしているが、通信相手のプログラム不良により戻らない状況である。	ソース	WAIT処理のパラメタ		WAIT処理	・スナップダンプによる解析(WAIT箇所を見つけ、無限待ちかを確認する)		自プロセス	①自分のプロセスで自分のソースの範囲内	WAIT	無限待ち	時間	他の業務UPの不良	他の業務UP	他プロセス				
5 処理が終わらない(1時間経過しても何も変わらない)	CPU使用率の確認	該当プロセスのCPU使用率が0%のままで推移。 かつ (2.1)ソースから資源名の確認 または、	(1.1)トレースログでWAIT場所を特定する。 または、 (1.2)プロセスのスナップダンプを採取し、PCの値を確認する。 かつ (2.1)ソースから資源名の確認 または、	(1)PCの場所が排他設定後のWAITで無限待ち。 (2)排他資源が永久に開放されない状況にある。	資源が解放されない可能性があるものに対して排他待ちしている。	ソース	排他処理のパラメタ		排他処理	・スナップダンプによる解析(WAIT箇所を見つけ、タイマーの値と排他処理の資源を確認する) ・他のプロセスが不当に排他している場合には、他のプロセス		自プロセス	①自分のプロセスで自分のソースの範囲内	WAIT	無限待ち	排他資源	該当せず	自分以外のプログラム	自分のソース				
6 処理が終わらない(1時間経過しても何も変わらない)	CPU使用率の確認	該当プロセスのCPU使用率が0%のままで推移。 かつ (2.1)ソースから資源名の確認 または、	(1.1)トレースログでWAIT場所を特定する。 または、 (1.2)プロセスのスナップダンプを採取し、PCの値を確認する。 かつ (2.1)ソースから資源名の確認 または、	(1)PCの場所が排他設定後のWAITで無限待ち。 (2)排他資源が永久に開放されない状況にある。	資源が解放されない可能性があるものに対して排他待ちしている。タイマーが大きい値となっている。	ソース	排他処理のパラメタ		排他処理	・スナップダンプによる解析(WAIT箇所を見つけ、タイマーの値と排他処理の資源を確認する) ・他のプロセスが不当に排他している場合には、他のプロセス		自プロセス	①自分のプロセスで自分のソースの範囲内	WAIT	タイマー値大	排他資源	該当せず	自プロセス	自分のソース				
7 処理が終わらない(1時間経過しても何も変わらない)	CPU使用率の確認	該当プロセスのCPU使用率が0%のままで推移。 かつ (2.1)ソースから資源名の確認 または、	(1.1)トレースログでWAIT場所を特定する。 または、 (1.2)プロセスのスナップダンプを採取し、PCの値を確認する。 かつ (2.1)ソースから資源名の確認 または、	(1)PCの場所が排他設定後のWAITで無限待ち。 (2)排他資源が永久に開放されない状況にある。	排他を解除するプロセスが開放時の資源名を誤ったため、排他解除がされていない状況となっていました。	ソース	排他処理のパラメタ		排他処理	・スナップダンプによる解析(WAIT箇所を見つけ、タイマーの値と排他処理の資源を確認する) ・他のプロセスの資源解放処理を見直す必要がある。		自プロセス	⑤他プロセスだが自プロセスと違うソース	WAIT	無限待ち	排他資源	他の業務UPの不良	他の業務UP	他プロセス				
8 処理が終わらない(1時間経過しても何も変わらない)	CPU使用率の確認	該当プロセスのCPU使用率が0%のままで推移。 かつ (2.1)ソースから資源名の確認 または、 (2.2)ダンプから資源名の確認 かつ (3)自スレッドが既に別の排他も確保済みであるかを確認。 (4)同一マシン内にも同じ現象のスレッドが発生しているかを	(1.1)トレースログでWAIT場所を特定する。 または、 (1.2)プロセスのスナップダンプを採取し、PCの値を確認する。 かつ (2.1)ソースから資源名の確認 または、 (2.2)ダンプから資源名の確認 かつ	(1)複数のプロセスのPCの場所が排他設定後のWAITで、無限待ちの設定となっている。 (2)ロックが纏がけ状態にある。	複数のスレッド間で、デッドロックになっている。	ソース	排他処理のパラメタ		排他処理	・スナップダンプによる解析(WAIT箇所を見つけ、タイマーの値と排他処理の資源を確認する) ・他のプロセスでもデッドロックが発生していないかを確認する。		自プロセス	①自分のプロセスで自分のソースの範囲内	WAIT	デッドロック	排他資源	該当せず	自プロセス	自分のソース				
9 処理が終わらない(1時間経過しても何も変わらない)	CPU使用率の確認	該当プロセスのCPU使用率が0%のままで推移。 かつ (2.1)ソースからタイマー値の確認 または、 (2.2)ダンプからタイマー値の確認 かつ	(1.1)トレースログでWAIT場所を特定する。 または、 (1.2)プロセスのスナップダンプを採取し、PCの値を確認する。 かつ (2.1)ソースからタイマー値の確認 または、 (2.2)ダンプからタイマー値の確認 かつ	(1)PCの場所がタイマー設定後のWAITである。 (2)ハードウェアエラーがある。	I/Oエラー時のリトライ時に設定するタイマーが長すぎる。	ソース	WAIT処理のパラメタ	ログからI/Oエラーの確認	エラー処理	・スナップダンプによる解析(WAIT箇所を見つけ、タイマーの値を確認する)		自プロセス	①自分のプロセスで自分のソースの範囲内	WAIT	タイマー値大(リトライ)	時間	I/Oエラー	自プロセス	自分のソース				
10 処理が終わらない(1時間経過しても何も変わらない)	CPU使用率の確認	該当プロセスのCPU使用率が0%のままで推移。 かつ (2.1)ソースからタイマー値の確認 または、 (2.2)ダンプからタイマー値の確認 かつ	(1.1)トレースログでWAIT場所を特定する。 または、 (1.2)プロセスのスナップダンプを採取し、PCの値を確認する。 かつ (2.1)ソースからタイマー値の確認 または、 (2.2)ダンプからタイマー値の確認 かつ	(1)PCの場所がタイマー設定後のWAITである。 (2)通信エラーがある。	通信エラー時のリトライ時に設定するタイマーが長すぎる。	ソース	WAIT処理のパラメタ	ログから通信エラーの確認	エラー処理	・スナップダンプによる解析(WAIT箇所を見つけ、タイマーの値を確認する)		自プロセス	①自分のプロセスで自分のソースの範囲内	WAIT	タイマー値大(リトライ)	時間	通信エラー	自プロセス	自分のソース				
11 処理が終わらない(1時間経過しても何も変わらない)	CPU使用率の確認	該当プロセスのCPU使用率が0%のままで推移。 かつ	(1.1)トレースログでrecv場所を特定する。 または、 (1.2)プロセスのスナップダンプを採取し、PCの値を確認する。 かつ	(1)PCの場所が通信の受信待ち(recv)である。 (2)セッションが切れている。	セッションが切れたことを検知するログがない。	ソース	WAIT処理のパラメタ	自プロセスの通信処理のログ	エラー処理	・スナップダンプによる解析(WAIT箇所を見つけ、タイマーの値を確認する)		自プロセス	①自分のプロセスで自分のソースの範囲内	WAIT	無限待ち	時間	通信エラー	自プロセス	自分のソース				
12 処理が終わらない(1時間経過しても何も変わらない)	CPU使用率の確認	該当プロセスのCPU使用率が0%のままで推移。 かつ	(1.1)トレースログでrecv場所を特定する。 または、 (1.2)プロセスのスナップダンプを採取し、PCの値を確認する。 かつ	(1)PCの場所が通信の受信待ち(recv)である。 (2)通信相手が送信/返信しない状況にある。	通信相手が送信/返信しないケース(正常/不良)がある。	ソース	WAIT処理のパラメタ	他プロセスの通信相手の処理ログ	エラー処理	・スナップダンプによる解析(WAIT箇所を見つけ、タイマーの値を確認する)		自プロセス	①自分のプロセスで自分のソースの範囲内	WAIT	無限待ち	時間	他の業務UPの不良	自プロセス	自分のソース				
13 処理が終わらない(1時間経過しても何も変わらない)	CPU使用率の確認	該当プロセスのCPU使用率が0%に近い状態のままで推移。	(1.1)トレースログで排他資源を確保したままで、資源開放する前の状態を確認する。 または、 (1.2)プロセスのスナップダンプを複数回採取し、PCの値を確認する。 かつ (2)CPU使用率が高いプロセスが2つ以上あり、プライオリティの低いCPU使用率が0%近いプロセスが1個あるかを確認にする。 かつ	(1)プロセスのPCの場所のほとんどが排他処理の前後の値になっている。 (2)2つ以上のプロセスが短時間に連続して排他のシステムコールを発行している。 (3)プライオリティの低いプロセスは排他資源の開放ができずにCPUが0%近いままである。	自プロセスがプライオリティの低いプロセスが資源を確保している状態、他の複数のプライオリティの高いプロセスが間隔を空けずにリトライ(スピンループ)すると、低いプロセスがディスパッチされず無限に待ちになることがある。OSによっては発生しない場合もある。	ソース	排他処理のパラメタ	OS情報(プライオリティ)	プライオリティ	・他のプロセスが高いプライオリティで排他資源を確保できずループしていないかを確認する。	(システムとしてはこの【20番】の現象と同じ)最近ではマルチcoreのCPUが主流になったので、システム全体が停止することは少ない。(1coreのシステムだとシステム全体のCPU使用率が100%になる)	自プロセス	①自分のプロセスで自分のソースの範囲内	ムラな処理	プライオリティ	CPU	該当せず	他の業務UP	OS				

14	処理が終わらない(1時間経過しても何も変わらない)	CPU使用率の確認	該当プロセスのCPU使用率が100%のままで推移。	(1)1プロセスログで同じログが大量に出力されているかを確認する。 または、 (1)プロセスのスナップダンプを複数回採取し、PCの値を確認する。 かつ (2)システムコールトレースを確認する。	(1)プロセスのPCの場所のほとんどがUPのループ処理の中になっている。 (2)システムコールが発行されていない。	UPが無限ループしている。(ループ範囲内にシステムコールが存在しない。)	ソース	ループ処理		無限ループ	・スナップダンプを複数回採取しループ場所を特定する。 ・ログからループ場所を特定する。	最近マルチcoreのCPUが主流になったので、システム全体が停止することは少ない。	自プロセス	①自分のプロセスで自分のソースの範囲内	ループ	無限ループ	CPU	該当せず	自プロセス	自分のソース
15	処理が終わらない(1時間経過しても何も変わらない)	CPU使用率の確認	該当プロセスのCPU使用率が100%のままで推移。	(1)1プロセスログで同じログが大量に出力されているかを確認する。 または、 (1)プロセスのスナップダンプを複数回採取し、PCの値を確認する。 かつ (2)システムコールトレースを確認する。	(1)プロセスのPCの場所のほとんどがシステムコールの前後の値になっている。 (2)短時間に連続してシステムコールが発行されている。	UPが無限ループしている。(ループ範囲内にシステムコールが存在する。)	ソース	ループ処理		無限ループ	・スナップダンプを複数回採取しループ場所を特定する。 ・ログからループ場所を特定する。	最近マルチcoreのCPUが主流になったので、システム全体が停止することは少ない。	自プロセス	①自分のプロセスで自分のソースの範囲内	ループ	無限ループ	CPU	該当せず	自プロセス	自分のソース
16	処理が終わらない(1時間経過しても何も変わらない)	CPU使用率の確認	該当プロセスのCPU使用率が100%のままで推移。	(1)1プロセスログに大量の排他エラーが出力されているかを確認する。 または、 (1)プロセスのスナップダンプを複数採取し、PCの値を確認する。 かつ (2)1ソースから排他エラー時のリトライ間隔とリトライ回数を確認する。 または、 (2)ダンプから資源名の確認 かつ (3)システムコールトレースを確認する。(排他関数の発行状況の確認)	(1)プロセスのPCの場所のほとんどが排他システムコールの前後の値になっている。 (2)排他エラーに対して、リトライ間隔を設けずに直ぐにリトライしている。 (3)排他エラー時のリトライ回数が非常に大きい	UPが排他エラーに対して、無限リトライしている。	ソース	排他処理のロジック	ログから排他エラーの確認	リトライ	・スナップダンプを複数回採取しループ場所を特定する。 ・ログからループ場所を特定する。	最近マルチcoreのCPUが主流になったので、システム全体が停止することは少ない。	自プロセス	①自分のプロセスで自分のソースの範囲内	ループ	無限ループ	CPU	排他エラー	自プロセス	自分のソース
17	処理が終わらない(1時間経過しても何も変わらない)	CPU使用率の確認	該当プロセスのCPU使用率が100%のままで推移。	(1)1プロセスログに大量のシステムコールエラー(関数の延長も含む)関連のログが出力されているかを確認する。 または、 (1)プロセスのスナップダンプを複数採取し、PCの値を確認する。 かつ (2)1ソースからシステムコールエラー時(関数の延長も含む)のリトライ間隔とリトライ回数を確認する。 または、 (2)ダンプから発行しているシステムコールの特定する。 かつ (3)システムコールトレースを確認する。(システムコールの発行状況の確認)	(1)プロセスのPCの場所のほとんどがシステムコールの前後の値になっている。 (2)短時間に連続してシステムコールが発行されている。	UPがシステムコールエラーに対して、無限リトライしている。	ソース	システムコール処理のロジック	ログからシステムコールエラーの確認	リトライ	・スナップダンプを複数回採取しループ場所を特定する。 ・ログからループ場所を特定する。 ・システムコールの処理が軽い場合。	最近マルチcoreのCPUが主流になったので、システム全体が停止することは少ない。	自プロセス	①自分のプロセスで自分のソースの範囲内	ループ	無限ループ	CPU	システムコールエラー	自プロセス	自分のソース
18	処理が終わらない(1時間経過しても何も変わらない)	CPU使用率の確認	該当プロセスのCPU使用率が100%に近い状態のままで推移。	(1)1プロセスログに大量のI/Oエラー(関数の延長も含む)関連のログが出力されているかを確認する。 または、 (1)プロセスのスナップダンプを複数採取し、PCの値を確認する。 かつ (2)1ソースからI/Oエラー時(関数の延長も含む)のリトライ間隔とリトライ回数を確認する。 または、 (2)ダンプからI/Oを発行している場所かを特定する。 かつ (3)I/Oの発行状況を確認する。	(1)プロセスのPCの場所のほとんどがI/Oの前後の値になっている。 (2)短時間に連続してシステムコールが発行されている。	UPがI/Oエラーに対して、無限リトライしている。	ソース	I/O処理のロジック	ログからI/Oエラーの確認	リトライ	・スナップダンプを複数回採取しループ場所を特定する。 ・ログからループ場所を特定する。 ・I/Oの単価が軽い場合。	最近マルチcoreのCPUが主流になったので、システム全体が停止することは少ない。 I/OによってはCPU100%にならないこともある。	自プロセス	①自分のプロセスで自分のソースの範囲内	ループ	無限ループ	CPU	I/Oエラー	自プロセス	自分のソース
19	処理が終わらない(1時間経過しても何も変わらない)	CPU使用率の確認	該当プロセスのCPU使用率が100%に近い状態のままで推移。	(1)1プロセスログに大量の通信エラー関連のログが出力されているかを確認する。 または、 (1)プロセスのスナップダンプを複数採取し、PCの値を確認する。 かつ (2)1ソースから通信エラー時のリトライ間隔とリトライ回数を確認する。 または、 (2)ダンプから通信処理の場所かを特定する。 かつ (3)1)通信の状況を確認する。 または (3)2)バケットトレースを確認する。	(1)プロセスのPCの場所のほとんどが通信処理の前後の値になっている。 (2)短時間に連続してシステムコールが発行されている。	UPが通信エラーに対して、無限リトライしている。	ソース	通信処理のロジック	ログから通信エラーの確認	リトライ	・スナップダンプを複数回採取しループ場所を特定する。 ・ログからループ場所を特定する。 ・通信の単価が軽い場合。	最近マルチcoreのCPUが主流になったので、システム全体が停止することは少ない。 通信の種類によってはCPU100%にならないこともある。	自プロセス	①自分のプロセスで自分のソースの範囲内	ループ	無限ループ	CPU	通信エラー	自プロセス	自分のソース
20	処理が終わらない(1時間経過しても何も変わらない)	CPU使用率の確認	該当プロセスのCPU使用率が100%に近い状態のままで推移。	(1)1プロセスログに大量の排他エラーが出力されているかを確認する。 または、 (1)プロセスのスナップダンプを複数回採取し、PCの値を確認する。 かつ (2)CPU使用率が高いプロセスが2つ以上あり、プライオリティの低くCPU使用率が0%近いプロセスが1個あるかを確認する。 かつ	(1)プロセスのPCの場所のほとんどが排他処理の前後の値になっている。 (2)2つ以上のプロセスが短時間に連続して排他のシステムコールを発行されている。 (3)プライオリティの低いプロセスは排他資源の開放ができずにCPUが0%近いままである。 (4)排他エラー時のリトライ間隔がない。	他のプライオリティの低いプロセスが資源を確保している状態で、自プロセスを含む複数のプライオリティの高いプロセスが間隔を空けずにリトライ(スピンループ)すると、低いプロセスがディスパッチされず無限に待ちになることがある。OSによっては発生しない場合もある。	ソース	OS情報	プロセスのプライオリティ	プライオリティ	・他のプロセスが低いプライオリティで排他資源を確保したまま解放できない状態を確認する。	(システムとしてはこの【13番】の現象と同じ)最近マルチcoreのCPUが主流になったので、システム全体が停止することは少ない。(1coreのシステムだとシステム全体のCPU使用率が100%になる)	自プロセス	①自分のプロセスで自分のソースの範囲内	ムラな処理	プライオリティ	CPU	排他エラー	自分以外のプログラム	OS
21	処理が遅くなることがある。(ゆっくりとしか動かない)	CPU使用率の確認	該当プロセスのCPU使用率が高い状態のままで推移。	(1)1プロセスログに大量の排他エラーが出力されているかを確認する。 または、 (1)プロセスのスナップダンプを複数回採取し、PCの値を確認する。 かつ (2)CPU使用率が高いプロセスが2つ以上あり、プライオリティの低くCPU使用率が0%近いプロセスが1個あるかを確認する。 かつ	(1)プロセスのPCの場所のほとんどが排他処理の前後の値になっている。 (2)2つ以上のプロセスが短時間に連続して排他のシステムコールを発行されている。 (3)プライオリティの低いプロセスは排他資源の開放ができずにCPUが0%近いままである。 (4)排他エラー時のリトライ間隔の値が小さい。	他のプライオリティの低いプロセスが資源を確保している状態で、自プロセスを含む複数のプライオリティの高いプロセスが短い間隔でリトライすると、低いプロセスがディスパッチされず無限に待ちになることがある。OSによっては発生しない場合もある。	ソース	プロセスのプライオリティ	排他処理のパラメタ	プライオリティ	・他のプロセスが低いプライオリティで排他資源を確保したまま解放できない状態を確認する。	(システムとしてはこの【28番】の現象と同じ)最近マルチcoreのCPUが主流になったので、システム全体が停止することは少ない。(1coreのシステムだとシステム全体のCPU使用率が100%になる)	自プロセス	①自分のプロセスで自分のソースの範囲内	ムラな処理	プライオリティ	CPU	該当せず	自分以外のプログラム	OS
22	処理が遅くなることがある。	CPU使用率の確認	該当プロセスのCPU使用率が高い状態のままで推移。	I/O時のデータの書き込み先のファイルが暗号化対象かを確認する。	(1)該当プロセスの延長で暗号化処理が動作するかを確認する。	・該当プロセスの延長で暗号処理に時間がかかっている。 ・暗号/複合化処理などが同時に沢山実行されている。	システム内のPP	暗号化製品		I/Oの延長での処理	・プロセスのCPUが高くなっていないかを確認する。 ・暗号化製品が同居しているかを確認する。		自プロセス	③自分のプロセスだがAPIの延長	その他	同居製品固有の実装ロジック	CPU	該当せず	暗号化製品	ライブラリ
23	処理が遅くなることがある。	CPU使用率の確認	該当プロセスのCPU使用率が高い状態のままで推移。	I/O時のデータの読み込み先のファイルが複合化対象かを確認する。	(1)該当プロセスの延長で複合化処理が動作するかを確認する。	・該当プロセスの延長で複合処理に時間がかかっている。 ・暗号/複合化処理などが同時に沢山実行されている。	システム内のPP	暗号化製品		I/Oの延長での処理	・プロセスのCPUが高くなっていないかを確認する。 ・暗号化製品が同居しているかを確認する。		自プロセス	③自分のプロセスだがAPIの延長	その他	同居製品固有の実装ロジック	CPU	該当せず	暗号化製品	ライブラリ
24	処理が遅くなることがある。	CPU使用率の確認	システム全体のCPU使用率が高く、該当プロセスのCPU使用率もそこそこ高い状態のままで推移。	(1)1プロセスログに大量のシステムコールエラー(パイプ通信など)関連のログが出力されているかを確認する。 または、 (1)ダンプから発行しているシステムコールを特定する。 かつ (2)システムコールトレースを確認する。(システムコールの発行状況の確認) かつ (3)システムコール(パイプ通信など)のエラー種別がビジーの時のリトライ間隔と回数の確認。 かつ (4)相手のプロセスの状況の確認。	(1)該当プロセスのPCの場所のほとんどがシステムコールエラー(パイプ通信など)関連の値になっている。 (2)該当プロセスからリトライが大量にあるかの確認。 (3)相手のプロセスがビジーの状況かの確認。 (4)リトライ回数の確認。 (5)リトライ間隔の確認。	他プロセスのビジーが解消されるまで長くなる可能性があるにも関わらず、リトライ間隔が短くリトライ回数が多すぎる。	ソース	システムコール処理のロジック	ログからビジーのエラーの確認	リトライ	・該当プロセスのCPU使用率を確認 ・PCの値 ・スナップダンプによる解析(WAIT箇所を見つける) ・システムコールトレース ・システムコール(パイプ通信など)のエラー種別がビジーのリトライ処理の確認		自プロセス	①自分のプロセスで自分のソースの範囲内	ループ	タイマー値小(リトライ多発)	CPU	相手がビジーによるリトライ	他の業務UP	自分のソース



25	処理が遅くなることがある。	CPU使用率の確認	システム全体のCPU使用率が高く、該当プロセスのCPU使用率もそこそこ高い状態のままで推移。	(1)1)トレースログでWAIT場所を特定する。 または、 (2)プロセスのスナップダンブを採取し、PCの値を確認する。 かつ (2)パイプ通信の所で止まっている。	パイプ詰まりが発生していないか。	パイプ通信の相手方がビジーな状態で、パイプからデータをなかなか取り出さないことで送信元も動けない状態に陥っている。	ソース	システムコール処理のロジック	ログからパイプ通信の相手方がビジーかの確認	通信相手のプロセスの状況	・該当プロセスと通信相手のCPU使用率を確認 ・PCの値 ・スナップダンブによる解析(WAIT箇所を見つける) ・システムコールトレース ・パイプ通信の書き込み時間の確認 ・通信相手のプロセスのパイプ通信の取り出し時間の確認。	他プロセス	④他プロセスだが自プロセスと同じソース	ループ	タイマー値小(リトライ多発)	CPU	相手がビジーによるリトライ	同様の業務UP	他プロセス
26	処理が遅くなることがある。	CPU使用率の確認	システム全体のCPU使用率が高く、該当プロセスのCPU使用率もそこそこ高い状態のままで推移。	(1)1)トレースログに大量のシステムコールエラーが「not ready」エラーのログが出力されているかを確認する。 または、 (2)ダンプから発行しているシステムコールを特定する。 かつ (2)システムコールトレースを確認する。(システムコールの発行状況の確認) かつ (3)システムコールエラーが「not ready」エラー時のリトライ間隔と回数の確認。	(1)該当プロセスのPCの場所のほとんどがシステムコール処理の前後の値になっている。 (2)ログにシステムコールエラーの値が「not ready」エラーかを確認する。 (3)該当プロセスからリトライが大量にあるかの確認。 (4)リトライ回数の値の確認。 (5)リトライ間隔の確認。	「not ready」が解消されるまで長くなる可能性があるにも関わらず、リトライ間隔が短くリトライ回数が多すぎる。	ソース	システムコール処理のロジック	ログから「not ready」でのエラーかの確認	リトライ	・該当プロセスのCPU使用率を確認 ・PCの値 ・スナップダンブによる解析(WAIT箇所を見つける) ・システムコールトレース ・システムコールのエラー種別が「not ready」のリトライ処理の確認	自プロセス	①自分のプロセスで自分のソースの範囲内	ループ	タイマー値小(リトライ多発)	CPU	相手が「not ready」時のリトライ	他の業務UP	自分のソース
27	処理が遅くなることがある。	該当プロセスのCPU使用率の確認	該当プロセスのCPU使用率が50%以下ぐらゐの状態のままで推移。	(1)1)複数プロセスのトレースログで排他処理が多いかを確認する。 または、 (2)複数プロセスのスナップダンブを複数回採取し、PCの値を確認する。 かつ (2)排他処理名を確認する。 かつ (3)システムコールトレースを確認する。	(1)該当プロセスのPCの場所のほとんどが排他処理の前後の値になっている。 (2)他のプロセスのPCの場所のほとんどが排他処理の前後の値になっている。 (3)同じ排他処理を使用していないかの確認。	1つの排他処理に対して、複数のプロセスが競合している。	ソース	排他処理	他プロセスとの競合	・スナップダンブによる解析(WAIT箇所を見つける) ・排他処理の排他処理名が他のプロセスでもよく使用するものかを確認する	対策としては、 ・排他する範囲を短くする。 ・排他する回数を減らす。	自プロセス	①自分のプロセスで自分のソースの範囲内	競合	不足	CPU	該当せず	同様の業務UP	自分のソース
28	処理が遅くなることがある。	該当プロセスのCPU使用率の確認	該当プロセスのCPU使用率が50%以下ぐらゐの状態のままで推移。	(1)1)トレースログでプロセス排他処理があるかを確認する。 または、 (2)複数プロセスのスナップダンブを複数回採取し、PCの値を確認する。 かつ (2)システムコールトレースを確認する。 かつ (3)他のプロセスも同様かを確認する。	(1)該当プロセスのPCの場所のほとんどがプロセス排他の値になっている。 (2)他のプロセスのPCの場所のほとんどがプロセス排他の値になっている。	プロセス排他処理があり、競合している。	ソース	排他処理	他プロセスとの競合	・スナップダンブによる解析(WAIT箇所を見つけ、プロセス排他処理の有無を確認する)	対策としては、 ・排他する範囲を短くする。 ・排他する回数を減らす。	自プロセス	①自分のプロセスで自分のソースの範囲内	競合	不足	CPU	該当せず	同様の業務UP	自分のソース
29	処理が遅くなることがある。	該当プロセスのCPU使用率の確認	該当プロセスのCPU使用率が50%以下ぐらゐの状態のままで推移。	(1)1)トレースログでスレッド排他処理があるかを確認する。 または、 (2)複数プロセスのスナップダンブを複数回採取し、PCの値を確認する。 かつ (2)システムコールトレースを確認する。 かつ (3)他のスレッドも同様かを確認する。	(1)該当プロセスのPCの場所のほとんどがスレッド排他の値になっている。 (2)他のプロセスのPCの場所のほとんどがスレッド排他の値になっている。	スレッド排他処理があり、競合している。	ソース	排他処理	他スレッドとの競合	・スナップダンブによる解析(WAIT箇所を見つけ、スレッド排他処理の有無を確認する)	対策としては、 ・排他する範囲を短くする。 ・排他する回数を減らす。	自プロセス	①自分のプロセスで自分のソースの範囲内	競合	不足	CPU	該当せず	同様の業務UP	自分のソース
30	処理が遅くなることがある。	CPU使用率の確認	システム全体のCPU使用率が高く、該当プロセスのCPU使用率が低い状態のままで推移。	(1)1)トレースログに大量の排他エラーが出力されているかを確認する。 または、 (2)プロセスのスナップダンブを複数回採取し、PCの値を確認する。 かつ (2)CPU使用率が高いプロセスが2つ以上あり、プライオリティの低くCPU使用率が近いプロセスが1個あるかを確認する。 かつ	(1)プロセスのPCの場所が排他処理の開放処理の値になっている。 (2)CPU使用率の低いプロセスのプライオリティが低く、排他処理の開放がディスパッチされずなかなか開放できない状態のままである。 (3)他の2つ以上のプロセスが短時間に連続して排他処理を確保するが、エラー時のリトライ間隔の値が小さい。	自プロセスがプライオリティの低いプロセスが資源を解放できない状態、他の複数のプライオリティの高いプロセスが短い間隔でリトライすると、低いプロセスがディスパッチされず無限に待ちになることがある。OSによっては発生しない場合もある。	ソース	排他処理のパラメータ プロセスのプライオリティ	プライオリティ	・他のプロセスが高いプライオリティで排他処理を確保できずループしていないかを確認する	システムとしてはこの(21番)の現象と同じ。最近ではマルチcoreのCPUが主流になったので、システム全体が停止することは少ない。(1coreのシステムだとシステム全体のCPU使用率が100%になる)	自プロセス	①自分のプロセスで自分のソースの範囲内	ムラな処理	プライオリティ	CPU	該当せず	自プロセス	OS
31	処理が遅くなることがある。	CPU使用率の確認	システム全体のCPU使用率が高く、該当プロセスのCPU使用率も高い状態のままで推移。	(1)他のプロセスのCPU使用率の状況も確認する。 (2)システム全体のI/Oなどの書き込みのサイズと回数を確認する。 (3)システムコールトレースを確認する。	(1)他のプロセスの延長で暗号化処理が動作するかを確認する。 (2)別のプロセスも遅くならないかを確認する。 (3)暗号化の対象データサイズが通常より大きい、または数が多いかを確認する。	・他のプロセスの延長で暗号化処理に時間がかかっている。 ・暗号処理などが同時に沢山実行されている。 ・他のプロセスが大量にCPUを使用した結果、該当プロセスのCPUが十分に供給されない。	システム内のPP	暗号化製品	I/Oの延長での処理	・システムプロセスのCPUが高くなっているかを確認する。 ・暗号化製品が同居しているかを確認する。	他プロセス	⑤他プロセスだが自プロセスと違うソース	競合	同居製品固有の実装ロジック	CPU	該当せず	暗号化製品	他プロセス	
32	処理が遅くなることがある。	CPU使用率の確認	システム全体のCPU使用率が高く、該当プロセスのCPU使用率も高い状態のままで推移。	(1)他のプロセスのCPU使用率の状況も確認する。 (2)システム全体のI/Oなどの書き込みのサイズと回数を確認する。 (3)システムコールトレースを確認する。	(1)他のプロセスの延長で複合化処理が動作するかを確認する。 (2)別のプロセスも遅くならないかを確認する。 (3)複合化の対象データサイズが通常より大きい、または数が多いかを確認する。	・他のプロセスの延長で複合化処理に時間がかかっている。 ・複合化処理などが同時に沢山実行されている。 ・他のプロセスが大量にCPUを使用した結果、該当プロセスのCPUが十分に供給されない。	システム内のPP	暗号化製品	I/Oの延長での処理	・プロセスのCPUが高くなっているかを確認する。 ・暗号化製品が同居しているかを確認する。	他プロセス	⑤他プロセスだが自プロセスと違うソース	競合	同居製品固有の実装ロジック	CPU	該当せず	暗号化製品	他プロセス	
33	処理が遅くなることがある。	CPU使用率の確認	システム全体のCPU使用率が高く、該当プロセスのCPU使用率が低い状態のままで推移。	(1)ウイルスチェックプログラムのパターンファイルの更新日時を確認する。 (2)該当プロセスだけでなく、他のプロセスの状況も確認する。 (3)システム全体のI/Oなどのサイズと回数を確認する。 (4)システムコールトレースを確認する。	(1)パターンファイルの更新後から遅くなったかを確認する。 (2)別のプロセスも遅くならないかを確認する。 (3)ウイルスチェックの対象データサイズが通常より大きい、または数が多いかを確認する。	・ウイルスチェック処理などが同時に沢山実行されている。 ・他のプロセスが大量にCPUやI/Oがある場合、該当プロセスのCPUが十分に供給されなかったり、I/Oに時間がかかる。	システム内のPP	ウイルス監視ソフト	競合	・ウイルス監視製品が同居しているかを確認する。 ・パターンファイルが変わったことで、性能がグレードしていないか他のシステムなどの状況も確認する。	他プロセス	⑤他プロセスだが自プロセスと違うソース	競合	同居製品固有の実装ロジック	CPU	該当せず	ウイルス監視製品	他プロセス	
34	処理が遅くなることがある。	CPU使用率の確認	システム全体のCPU使用率が高いが、該当プロセスのCPU使用率が低い状態のままで推移。	(1)該当プロセスだけでなく、他のプロセスのCPU使用率の状況も確認する。 (2)システムコールトレースを確認する。	(1)同時に実行している他のプロセスがCPUを大量に使用しているかを確認する。	・他のプロセスが大量にCPUを使用した結果、該当プロセスのCPUが十分に供給されない。	システム全体のプロセス状況	競合	競合	・システム全体のCPU使用率と自プロセスのCPU使用率を確認する。 ・自プロセスのCPUが予想より低い値で推移していないかを確認する。	CPUを沢山消費するのは、 ・暗号化、複合化 ・圧縮、解凍 ・画像処理 ・印刷処理 ・pdf化処理 ・DBアクセス ・full GCなどがある。	他プロセス	⑤他プロセスだが自プロセスと違うソース	競合	不足	CPU	該当せず	自分以外のプログラム	他プロセス
35	処理が遅くなることがある。	CPU使用率の確認	システム全体のCPU使用率が高いが、該当プロセスのCPU使用率が低い状態のままで推移。	(1)該当プロセスだけでなく、他のプロセスのCPU使用率の状況も確認する。 (2)システムコールトレースを確認する。	(1)同時に実行している他のプロセスがCPUを大量に使用しているかを確認する。 (2)該当プロセスのプライオリティが低くならないかを確認する。	・他のプロセスが大量にCPUを使用した結果、該当の低いプライオリティのプロセスのCPUが十分に供給されない。	システム全体のプロセス状況	プライオリティの確認	プライオリティ	・システム全体のCPU使用率と自プロセスのCPU使用率を確認する。 ・自プロセスのCPUが予想より低い値で推移していないかを確認する。 ・プロセスのプライオリティの確認。	1coreの場合、2つのプロセスがループするとプライオリティの低いプロセスはほとんどディスパッチされないことがある。但しLOSによるcore数より多くのスレッドがループしている状況下だと、プライオリティの低いプロセスはほとんどディスパッチされない。	他プロセス	⑤他プロセスだが自プロセスと違うソース	競合	不足	CPU	該当せず	自分以外のプログラム	他プロセス
36	処理が遅くなることがある。	CPU使用率の確認	該当プロセスのCPU使用率が低い状態のままで推移。	(1)1)トレースログに大量のシステムコール(関数の延長も含む)関連のログが出力されているかを確認する。 または、 (2)ダンプから発行しているシステムコールの特定する。 かつ (2)システムコールトレースを確認する。(システムコールの発行状況の確認) かつ (3)システムコールエラー時のリトライ回数の間隔と回数の確認。	(1)該当プロセスのPCの場所のほとんどがシステムコールの前後の値になっている。 (2)同じシステムコールが頻繁にコールされている。 (3)UNIXの場合システム全体のsysのCPU使用率が高くないかを確認する。	UPがシステムコールエラーに対して、連続してリトライしている。	ソース	システムコール処理のロジック	ログからシステムコールエラーの確認	リトライ	・スナップダンブを複数回採取しループ場所を特定する。 ・ログからループ場所を特定する。 ・システムコールの処理が重い場合。	自プロセス	①自分のプロセスで自分のソースの範囲内	ループ	無限ループ	CPU	システムコールエラー	自プロセス	自分のソース

37	処理が遅くなること がある。	CPU使用率の確認	該当プロセスの CPU使用率が低い 状態のままで推移。	(1)トレースログにI/Oエラーのログが出力されているかを 確認する。 または、 (1)ダンプから発行しているI/O処理かを特定する。 かつ (2)該当プロセスのI/O回数を確認する。 かつ (3)システムコールトレースを確認する。	(1)該当プロセスのPCの場所のほとんど がI/Oの前後の値になっている。 (2)該当プロセスからI/Oが大量にあるかの 確認。	UPがI/Oエラーに対して、連続 してリトライしている。	ソース	I/O処理 のロジック	ログから I/Oエラー の確認	リトライ	・スナップダンプを複数回採取 しループ場所を特定する。 ・ログからループ場所を特定する。 ・I/Oの単価が重い場合(大きな データの同期I/Oで時間がかか るような時)。	最近ではマルチcoreの CPUが主流になったの で、システム全体が停 止することは少ない。 I/Oによっては CPU100%にならないこ ともある。	自プロセス	①自分のプロセスで自分 のソースの範囲内	ループ	無限ループ	CPU	I/Oエラー	自プロセス	自分のソース
38	処理が遅くなること がある。	CPU使用率の確認	該当プロセスの CPU使用率が低い 状態のままで推移。	(1)トレースログに通信エラーのログが出力されているかを 確認する。 または、 (1)ダンプから発行しているI/O処理かを特定する。 かつ (2)該当プロセスの通信状況を確認する。 かつ (3)システムコールトレースを確認する。	(1)該当プロセスのPCの場所のほとんど が通信処理の前後の値になっている。 (2)該当プロセスから通信処理が大量に あるかの確認。	UPが通信エラーに対して、連 続リトライしている。	ソース	通信処理 のロジック	ログから 通信エ ラーの確 認	リトライ	・スナップダンプを複数回採取 しループ場所を特定する。 ・ログからループ場所を特定する。 通信の 単価が重い場合(大きな データの通信で通信時間に 時間がかかる時)。	最近ではマルチcoreの CPUが主流になったの で、システム全体が停 止することは少ない。 通信の種類によっては CPU100%にならないこ ともある。	自プロセス	①自分のプロセスで自分 のソースの範囲内	ループ	無限ループ	CPU	通信エラー	自プロセス	自分のソース
39	処理が遅くなること がある。	CPU使用率の確認	該当プロセスの CPU使用率が低い 状態のままで推移。	(1)トレースログに大量のシステムコールエラー(パイプ通 信など)関連のログが出力されているかを確認する。 または、 (1)ダンプから発行しているシステムコールを特定する。 かつ (2)システムコールトレースを確認する。(システムコールの 発行状況の確認) かつ (3)システムコール(パイプ通信など)のエラー種別がビジー の時のリトライ間隔と回数の確認。 かつ (4)相手のプロセスの状況の確認。	(1)該当プロセスのPCの場所のほとんど がシステムコールエラー(パイプ通信な ど)関連の前後の値になっている。 (2)ログに該当プロセスからリトライが大 量にあるかの確認。 (3)相手のプロセスがビジーの状況かの 確認。 (4)リトライ間隔の値の確認。	他プロセスのビジーが解消され るまでの時間に対して、リトライ 間隔が長すぎる。	ソース	システム コール処 理のロジ ック	ログから ビジーで のエラー かの確認	リトライ	・該当プロセスのCPU使用率を 確認 ・PCの値 ・スナップダンプによる解析 (WAIT箇所を見つける) ・システムコールトレース ・システムコール(パイプ通信な ど)のエラー種別がビジーのリ トライ処理の確認		自プロセス	①自分のプロセスで自分 のソースの範囲内	ループ	タイマー値大(リトライ)	時間	相手がビジーに よるリトライ	他の業務UP	自分のソース
40	処理が遅くなること がある。	CPU使用率の確認	該当プロセスの CPU使用率が低い 状態のままで推移。	(1)トレースログに大量のシステムコールエラーが「not ready」エラーのログが出力されているかを確認する。 または、 (1)ダンプから発行しているシステムコールを特定する。 かつ (2)システムコールトレースを確認する。(システムコールの 発行状況の確認) かつ (3)システムコールエラーが「not ready」エラー時のリトライ間 隔と回数の確認。	(1)該当プロセスのPCの場所のほとんど がシステムコール処理の前後の値に なっている。 (2)ログに該当プロセスからリトライが大 量にあるかの確認。 (3)該当プロセスからリトライが大量にあ るかの確認。 (4)リトライ間隔の値の確認。	「not ready」が解消されるまで の時間に対して、リトライ間隔 が長すぎる。	ソース	システム コール処 理のロジ ック	ログから 「not ready」で のエラー かの確認	リトライ	・該当プロセスのCPU使用率を 確認 ・PCの値 ・スナップダンプによる解析 (WAIT箇所を見つける) ・システムコールトレース ・システムコールのエラー種別 が「not ready」のリトライ処理 の確認		自プロセス	①自分のプロセスで自分 のソースの範囲内	ループ	タイマー値大(リトライ)	時間	相手がnot ready時のリト ライ	自プロセス	自分のソース
41	処理が遅くなること がある。	装置(VM環境)全体 のCPU使用率が高い 状態かの確認	該当VMのcore数 とVM装置全体の CPU使用率の確認。	(1)該当VMだけでなく、他のVMの状況も確認する。 (2)該当VMのcore数が動的に変化する場合には、core数の 推移を確認する。	同時に実行している他のVMのcore数と CPUを大量に使用しているかを確認す る。	・他のVMが大量にCPUを使用 した結果、該当VMのCPUが十分 に供給されていない。	VM全体			競合	VMの場合には、自マシンだけ のCPUの性能情報だけでは負 荷がわからないので、サーバー 全体のCPUの状況を確認する 必要がある。 特に動的にcore数が変化する ような設定の場合には、他VM の影響を受けることがあるので 注意が必要である。		他ホスト	⑦他ホストプロセスだが 自プロセスと違うソース	競合	不足	CPU	該当せず	他ホスト	他ホスト
42	件数が増加すると指数 関数的に処理に 時間がかかる	件数を変化させて性能 測定し、グラフ化 する。 ・いろいろな件数で 処理時間の計測 ・CPU使用率の確認 ・システムコール トレース	件数による傾向を 分析する。	・グラフの傾向から該当しそうなソース箇所を洗い出す。 ・CPUの使用率の傾向から該当しそうなソース箇所を洗い 出す。 ・システムコールトレースの発行状況から該当しそうなソ ース箇所を洗い出す。 該当箇所に対して、ソース解析して、回数の多い処理に関 して、ループがあるかを確認する。 ログが不十分な場合には、ログ強化して該当箇所を絞り込	件数がn件の時に、n回のループの中に n回のループがある。ループがネストす る処理になっている。 特にループの中にシステムコールがあ ると、非常に悪化する。	ループ構造	ソース	ループ処 理	ループ	ループ	扱うデータ件数の規模を確認 する。		自プロセス	①自分のプロセスで自分 のソースの範囲内	ムダな 処理	ループ中のループ	CPU	大規模になった 時	該当せず	自分のソース
43	件数が増加すると指数 関数的に処理に 時間がかかる	件数を変化させて性能 測定し、グラフ化 する。 ・いろいろな件数で 処理時間の計測 ・CPU使用率の確認 ・I/O回数の確認 ・システムコール トレース	件数による傾向を 分析する。	ソースで「ソート」処理がないかを確認する。 ソート処理の対象がメモリ上の場合には、CPU使用率の傾 向を分析する。 ソート処理の対象がディスク上の場合には、I/O回数の傾 向を分析する。	ソート処理のロジックが悪く、性能測定 のグラフの傾向と一致するかを確認す る。	ソート処理の実装ミス	ソース	ソート処 理	ソート	ソート	扱うデータ件数の規模を確認 する。		自プロセス	①自分のプロセスで自分 のソースの範囲内	ムダな 処理	ソート、コンペアなど の実装が悪い	CPU	大規模になった 時	該当せず	自分のソース
44	件数が増加すると指数 関数的に処理に 時間がかかる	件数を変化させて性能 測定し、グラフ化 する。 ・いろいろな件数で 処理時間の計測 ・CPU使用率の確認 ・I/O回数の確認 ・システムコール トレース	件数による傾向を 分析する。	ソースで「コンペア」処理がないかを確認する。 コンペア処理の対象がメモリ上の場合には、CPU使用率の傾 向を分析する。 コンペア処理の対象がディスク上の場合には、I/O回数の傾 向を分析する。	コンペア処理のロジックが悪く、性能測定 のグラフの傾向と一致するかを確認す る。	コンペア処理の実装ミス	ソース	コンペア 処理	コンペア	コンペア	扱うデータ件数の規模を確認 する。		自プロセス	①自分のプロセスで自分 のソースの範囲内	ムダな 処理	ソート、コンペアなど の実装が悪い	CPU	大規模になった 時	該当せず	自分のソース
45	件数が増加すると指数 関数的に処理に 時間がかかる	件数を変化させて性能 測定し、グラフ化 する。 ・いろいろな件数で 処理時間の計測 ・CPU使用率の確認 ・I/O回数の確認 ・システムコール トレース	件数による傾向を 分析する。	ソースで「サーチ」処理がないかを確認する。 サーチ処理の対象がメモリ上の場合には、CPU使用率の傾 向を分析する。 サーチ処理の対象がディスク上の場合には、I/O回数の傾 向を分析する。	サーチ処理のロジックが悪く、性能測定 のグラフの傾向と一致するかを確認す る。	サーチ処理の実装ミス 件数が多い場合でも、シーケン シャルサーチである。 最低でもバイナリサーチは必 要。	ソース	サーチ処 理	サーチ	サーチ	・扱うデータ件数の規模を確認 する。 ・ヒットするデータの場所は、 「最初」、「中間」、「最後」などと 変化させて性能測定し、グラフ 化する。		自プロセス	①自分のプロセスで自分 のソースの範囲内	ムダな 処理	サーチ、インデックス などの実装が悪い	CPU	大規模になった 時	該当せず	自分のソース
46	件数が増加すると指数 関数的に処理に 時間がかかる	件数を変化させて性能 測定し、グラフ化 する。 ・いろいろな件数で 処理時間の計測 ・CPU使用率の確認 ・I/O回数の確認 ・システムコール トレース	件数による傾向を 分析する。	ソースで「データ処理時に、「インデクス」が存在するかを確 認する。 処理がメモリ上の場合には、CPU使用率の傾向を分析する。 処理がディスク上の場合には、I/O回数の傾向を分析する。	「インデクス」が存在せず、性能測定 のグラフの傾向と一致するかを確認す る。	件数が非常に多い場合がある にも関わらず、「インデクス」が ない。 インデクスがないと、追加、更 新、参照、削除の総てに時間 がかかる。	ソース	インデクス 処理	インデクス	インデクス	扱うデータ件数の規模を確認 する。		自プロセス	①自分のプロセスで自分 のソースの範囲内	ムダな 処理	サーチ、インデックス などの実装が悪い	CPU	大規模になった 時	該当せず	自分のソース
47	件数が増加すると指数 関数的に処理に 時間がかかる	件数を変化させて性能 測定し、グラフ化 する。 ・いろいろな件数で 処理時間の計測 ・CPU使用率の確認 ・I/O回数の確認 ・システムコール トレース	件数による傾向を 分析する。	ソースで「キューイング」する処理がないかを確認する。 キューイング処理がメモリ上の場合には、CPU使用率の傾 向を分析する。 キューイング処理がディスク上の場合には、I/O回数の傾向 を分析する。	キューイングのロジックが悪く、性能測定 のグラフの傾向と一致するかを確認す る。	要求をキューイングして実装し ているがキューの管理方式が 悪いかやり方が悪い キューのつなぎ方は使われ方 によってFIFO、FILOを使い分け る。 また、場合によってはバック チェーンが必要なこともある。	ソース	キューイ ング処理	キューイ ング	キューイ ング	・キューに登録してある件数を 変化させ、登録件数によるグラ フ化する。 ・同時に沢山の登録で問題な いかなを確認する。 ・同時に登録と取り出しを実行 して問題ないかを確認する。		自プロセス	①自分のプロセスで自分 のソースの範囲内	ムダな 処理	キューなどの実装が 悪い	CPU	大規模になった 時	該当せず	自分のソース

48	マシンをリブレースまたはVM化したら、性能が悪化した。	ソースでメモリアクセスを頻繁に行うかを確認する。	メモリアクセスが非常に多いプログラムである。	・マシン構成の確認	SMP(Symmetric Multi Processor)構成のマシンかを確認する。	SMP(Symmetric Multi Processor)構成のマシン4台のサーバを1個のOSで1個のマシンように扱うため、他のマシンCPUのメモリアクセスまで同期が必要になるので効率が悪く、また、排他処理の時は他のマシンも止まる。	サーバー	マシン構成	SMP構成	・UPがオンメモリでいろいろと処理を実施しているかを確認する。	SMP構成下のVMもあるので、注意が必要。	自プロセス	①自分のプロセスで自分のソースの範囲内	キャッシュ	遅いキャッシュ	メモリ	該当せず	該当せず	メモリ性能
49	マシンをリブレースまたはVM化したら、性能が悪化した。	ソースでメモリアクセスを頻繁に行うかを確認する。特に科学的な処理があるかも確認する。	メモリアクセスが非常に多いプログラムである。	・OSの設定内容の確認	CPUに対して、ディスパッチの前後でできるだけ同じCPUを割り当てるように設定かを確認する。	科学計算などの場合には、プロセスとCPUを紐づけしておく必要がある。	OSの設定		メモリ性能	科学計算などと同じメモリに対して、大量にメモリアクセスが発生する処理があるかを確認する。	・CPUに近い側からL1(レベル1)キャッシュ、L2(レベル2)キャッシュ、L3(レベル3)キャッシュまでである。通常のプログラムではあまり気にする必要がないが、科学計算などでは影響を受ける場合がある。 ・影響を受けるプログラムの場合には、プロセスとCPUを紐づける場合もある。 ・科学技術計算などのように同じメモリ領域に連続して参照/更新を繰り返す場合など。CPUにあるメモリアクセスがディスパッチの度に無効になることで性能がダウンする。これはメモリだけでなく、レジスタなども影響を受ける。	自プロセス	①自分のプロセスで自分のソースの範囲内	キャッシュ	遅いキャッシュ	メモリ	該当せず	該当せず	メモリ性能
50	マシンをリブレースまたはVM化したら、性能が悪化した。	VMのスペックの確認	各VMIに対するcore数とメモリサイズを確認する。	(1)core数などが動的に変化するかの確認。 かつ (2)動作時の該当VMIに割り当てられたcore数の変化を確認する。	現象発生時の割り当てられているCPU数やcore数を確認する。	現象発生時は、他のVMのCPU使用量が大きく、該当VMのCPU資源が予定より少ない状況である。	VMの設定		VM構成	・該当VMIに割り当てられているCPU数とcore数 ・システム全体のCPU使用率が低いのか？ ・該当プロセスのCPU使用率が低いのか？ ・該当VMの最小CPU数、core数の確認		自プロセス	⑧その他	競合	不足	CPU	該当せず	他ホスト	他ホスト
51	マシンをリブレースまたはVM化したら、性能が悪化した。	OSが何bitアドレッシングモードかを確認	32bitアドレッシングモードOSかを確認	ソースで64bit演算処理の箇所を洗い出す	64bit演算処理が大量に実行されるかを確認する。	64bit演算が大量にあるにもかかわらず、32bitマシンである。	サーバー		OSのアドレッシングモード	マシンリブレース前後でOSの変化の有無を確認		自プロセス	①自分のプロセスで自分のソースの範囲内	ムラな処理	型の不一致など	CPU	該当せず	他ホスト	CPU性能(クロック)
52	処理が遅い	・ページング回数 ・物理メモリ使用量 ・仮想メモリ使用量	自プロセスのページングが多発している。			自プロセスのメモリ使用量が大きすぎる。	サーバー	メモリ	リソース不足	自プロセスのページング回数。		自プロセス	①自分のプロセスで自分のソースの範囲内	ムダな処理	メモリ確保回数を頻発	メモリ	該当せず	該当せず	メモリサイズ
53	処理が遅い	・ページング回数 ・物理メモリ使用量 ・仮想メモリ使用量	システム全体でページングが多発している。			システム全体で使用するメモリ量が大きすぎる。	サーバー	メモリ	リソース不足	システム全体のページング回数。		他プロセス	⑤他プロセスだが自プロセスと違うソース	競合	不足	メモリ	該当せず	自プロセス	メモリサイズ
54	処理が遅い	・ページング回数 ・物理メモリ使用量 ・仮想メモリ使用量	スラッシングが発生している。 プログラムがWAIT→READYになってしばらくは動きがほとんどない状態かを確認する。			システム全体で使用するメモリ量が大きすぎる。	サーバー	メモリ	リソース不足	・該当プロセスのページイン回数の傾向		他プロセス	⑤他プロセスだが自プロセスと違うソース	競合	不足	メモリ	該当せず	自プロセス	メモリサイズ
55	処理が遅い	メモリ使用量の確認	メモリ使用量が急激に増加しているかを確認する。	(1)メモリサイズの時間の経過による推移を確認する。 (2)スナップダンブによる解析(メモリの使用状況の確認) (3)ソースでメモリの確保、開放処理の洗い出し (4)システムコールトレースでメモリ確保関数の回数の確認	(1)メモリサイズの確保が巨大。 または (2)メモリの確保する回数が増える	(1)メモリの確保サイズが巨大。 または (2)メモリの確保する回数が増える	ソース	メモリの確保/開放のロジック	メモリの確保/開放	時々だが急激に遅くなるような状況があるかを確認。	OSによっては、大きなメモリを確保しても、実際にメモリへの書き込みがないと実メモリを割り当てないケースがあるので、大量のメモリへの書き込みも同時に実行しないと発生しないこともある。	自プロセス	③自分のプロセスだがAPIの延長	その他	OS固有の実装ロジック	メモリ	大規模になった時	該当せず	メモリ性能
56	処理が遅い	メモリ使用量の確認	メモリ使用量が急激に増加しているかを確認する。	(1)メモリサイズの時間の経過による推移を確認する。 (2)スナップダンブによる解析(メモリの使用状況の確認) (3)ソースでメモリの確保、開放処理の洗い出し (4)システムコールトレースでメモリ確保関数の回数の確認	(1)メモリサイズの確保/開放のシステムコールが大量に発行されているかを確認する。 (2)スナップダンブによる解析、確保したメモリサイズと数を確認する。 (3)ソースでメモリの確保時のサイズと回数を確認する。	データコピー時にコピー先の領域を1バイトずつ拡張しながらコピーしている。 データサイズが大きくなると指数関数的にメモリを消費する。また、CPUも消費し処理に時間がかかる。 メモリの断片化。	ソース	メモリの確保/開放のロジック	メモリの確保/開放	メモリ確保のシステムコールが大量に発行されていないかわかる	append()関数などだと関数の延長でこのような実装がされているので注意が必要。	自プロセス	②自分のプロセスだがライブラリの延長	その他	ライブラリ固有の実装ロジック	メモリ	大規模になった時	該当せず	ライブラリ
57	処理が遅い (長時間経過後に時間の経過と共に遅くなる)	CPU使用率の確認 UNIX, Linuxの場合には、sysとusrの比率も確認	時間の経過と共にCPU使用率が増加しているかを確認する。	(1)スナップダンブによる解析(メモリの使用状況の確認) (2)ソースでメモリの確保、開放処理の洗い出し (3)システムコールトレースでメモリ確保関数の回数の確認	(1)ダンブを解析し、メモリが断片化していないかを確認する。 (2)ソースでメモリの確保時のサイズより小さいサイズで開放するなどの処理がないかを確認。 (3)メモリの確保サイズが可変で頻繁に実行されているかを確認。	(100kバイトを確保し90kバイトだけ解放のような処理を繰り返すと、メモリが断片化してメモリの確保に時間がかかるようになる。)	ソース	メモリの確保/開放のロジック	メモリの確保/開放	・自分のソースのメモリの確保/解放処理 または、使用しているライブラリの延長で発生していることもあるので、注意が必要。	・メモリ使用量(見た目の使用メモリは少なくてもフラグメンテーションのためメモリ不足エラーになることがある)	自プロセス	①自分のプロセスで自分のソースの範囲内	ムラな処理	断片化	メモリ	該当せず	該当せず	メモリ性能
58	処理が遅い 但し一時的 (長時間経過後に時々急激に遅くなることがある)	CPU使用率の確認 UNIX, Linuxの場合には、sysとusrの比率も確認	CPU使用率の時間の経過による推移を確認する。	時々CPU使用率が高い状態が発生し、やがて解消されるような傾向があるかを確認する。	(1)一時的に遅い時に、CPU使用率が以上に高いかを確認する。 (2)OSがJavaのようなメモリのガーベージコレクションが実行していないかを確認する。 (3)ダンブ解析でメモリの内容を確認する。	OSのガーベージコレクションが実行されている。	OS	ガーベージコレクション処理のインストール方法	ガーベージコレクション処理	時々だが急激に遅くなるような状況があるかを確認。		他プロセス	⑤他プロセスだが自プロセスと違うソース	競合	不足	CPU	該当せず	自プロセス	メモリ性能
59	処理が遅い 但し一時的 (長時間経過後に時々急激に遅くなることがある)	CPU使用率の確認 UNIX, Linuxの場合には、sysとusrの比率も確認	CPU使用率の時間の経過による推移を確認する。	時々CPU使用率が高い状態が発生し、やがて解消されるような傾向があるかを確認する。	(1)一時的に遅い時に、CPU使用率が以上に高いかを確認する。 (2)Javaのメモリのガーベージコレクションが実行していないかを確認する。 (3)ダンブ解析でメモリの内容を確認する。	Javaのガーベージコレクションが実行されている。	Java	ガーベージコレクション処理のインストール方法	ガーベージコレクション処理	時々だが急激に遅くなるような状況があるかを確認。	使用しているJavaによって挙動が異なる。	自プロセス	②自分のプロセスだがライブラリの延長	その他	自プロセスの延長の固有の実装ロジック	CPU	該当せず	該当せず	ライブラリ
60	処理が遅い 突然遅くなる	CPU使用率の確認	CPU使用率が高い状態が続く	(1)システムコールトレースで通信の回数を確認する。 かつ (2)該当プロセスの通信量を計測する。 (3)ソースを解析し、通信処理の周辺のロジックを確認する。 かつ (4)通信相手(他ホストの場合もある)のプロセスも高負荷状態にあるかを確認する。	(1)通信の回数が非常に多いかを確認する。 (2)通信のデータ量が多いかを確認する。 (3)通信相手が受け取れない時に、再送処理があり、輻輳状態になるロジックがあるかを確認する。 (4)通信相手のプロセスがビジーな状態で通信を受け取れない状況であるかを確認する。	他プロセスとのデータの受け渡しがあるが、他プロセスが処理しきれないデータを戻す処理があり、2つのプロセス間の通信処理で輻輳が発生している。	ソース	通信処理のロジック	輻輳	件数とサイズを変化させて性能測定し、グラフ化する。 ・いろいろな件数で処理時間の計測	通信相手のプロセスが処理しきれない間は、問題が発生しない。 受信プロセスの処理性能が送信プロセスの通信量の状態が継続し、限界を超えると急激に遅くなる。	自プロセス	①自分のプロセスで自分のソースの範囲内	ムダな処理	輻輳	CPU	相手プロセスによるリトライ	同様の業務UP	他プロセス

61	処理が遅い 突然遅くなる	CPU使用率の確認	CPU使用率が高い状態が続く	(1)システムコールトレースで通信の回数を確認する。 かつ (2)該当プロセスの通信量を計測する。 (3)ソースを解析し、通信処理の周辺のロジックを確認する。 かつ (4)通信相手(他ホストの場合もある)のプロセスも高負荷状態にあるかを確認する。	(1)通信の回数が非常に多いかを確認する。 (2)通信のデータ量が多いかを確認する。 (3)通信相手が受け取れない時に、再送処理があり、輻輳状態になるロジックがあるかを確認する。 (4)通信相手のプロセスがビジー状態で通信を受け取れない状況であるかを	他ホストとのデータの受け渡しがあるが、他プロセスが処理しきれないデータを受信処理があり、2つのプロセス間の通信処理で輻輳が発生している。	ソース	通信処理のロジック	輻輳	件数とサイズを変化させて性能測定し、グラフ化する。 ・いろいろな件数で処理時間の計測	通信相手のプロセスが処理しきれない間は、問題が発生しない。 受信プロセスの処理性能が送信プロセスの送信量の状態が続くと、限界を超えると急激に遅くなる。	自プロセス	①自分のプロセスで自分のソースの範囲内	ムダな処理	輻輳	CPU	相手がビジーによるリトライ	他ホスト	他ホスト
62	処理が遅くなる。	CPU使用率の確認	CPU使用率が高い状態が続く	・システムログの確認 温度に関するログが出力されていないかを確認する。 または、 ・PCの温度の確認 マシン環境で高温環境で使用していないかを確認。 直接マシンに触れて高温になっていないかを確認。 冷却用のファンが正しく動作しているかの確認。		温度が高くなるとCPUクロック数を抑えるCPUがある。	サーバー	ハード環境	温度	システムによっては、システムログに温度異常のログが出力されることもある		自プロセス	①自分のプロセスで自分のソースの範囲内	その他	ハードウェア特性	CPU	該当せず	該当せず	CPU性能(クロック)
63	処理に時間がかかる	CPU使用率の確認	CPU使用率が高い状態が続く	・CPU使用率 ・システムコールトレースでメモリの確保/開放の回数が多いかを確認する。	メモリの確保/開放のサイズと回数を確認する。	小さなメモリの確保/解放を大量に繰り返している。	ソース		メモリの確保/開放	システムコールトレース		自プロセス	①自分のプロセスで自分のソースの範囲内	ムダな処理	メモリ確保回数を頻発	CPU	該当せず	該当せず	自分のソース
64	処理に時間がかかる	CPU使用率の確認	CPU使用率が高い状態が続く	・CPU使用率 ・システムコールトレースでメモリの確保/開放の回数が多いかを確認する。	メモリの確保/開放のサイズと回数を確認する。	小さなメモリの確保/解放を大量に繰り返している。 C++の関数で以下を使用している場合にも発生するケースがある。 ・strcpyなど	ソース	C++の関数	メモリの確保/開放	システムコールトレース		自プロセス	②自分のプロセスだがライブラリの延長	ムダな処理	メモリ確保回数を頻発	CPU	該当せず	該当せず	ライブラリ
65	処理に時間がかかる	CPU使用率の確認	CPU使用率が予想より高い。	ソースの確認。	32BitのCPUで16Bitの数値演算を大量に実行している。	型を合わせるための前処理でCPUを使用している。	ソース	型の確認	コードの実装方法			自プロセス	①自分のプロセスで自分のソースの範囲内	ムラな処理	型の不一致など	CPU	該当せず	該当せず	CPU性能(クロック)
66	処理に時間がかかる	CPU使用率の確認	CPU使用率が予想より高い。	ソースの確認。	64BitのCPUで32bitまたは16Bitの数値演算を大量に実行している。	型を合わせるための前処理でCPUを使用している。	ソース	型の確認	コードの実装方法			自プロセス	①自分のプロセスで自分のソースの範囲内	ムラな処理	型の不一致など	CPU	該当せず	該当せず	CPU性能(クロック)
67	処理に時間がかかる	CPU使用率の確認	CPU使用率が予想より高い。	ソースの確認。	演算式の各値の型が異なる演算を大量に実行している。	型を合わせるための前処理でCPUを使用している。	ソース	型の確認	コードの実装方法			自プロセス	①自分のプロセスで自分のソースの範囲内	ムラな処理	型の不一致など	CPU	該当せず	該当せず	CPU性能(クロック)
68	処理に時間がかかる	CPU使用率の確認	CPU使用率が予想より高い。	ソースの確認。	画面の描画処理を短い周期でループしているかを確認する。	画面の描画処理を短い周期でループしているかとCPU使用率が高くなる。	ソース	ロジック	ループ	GPUの種類の確認。 GPUの性能と特性を確認する。	グラフィック系のプログラムの場合、GPUによっても性能差がでる	自プロセス	①自分のプロセスで自分のソースの範囲内	その他	ハードウェア特性	その他のハード	該当せず	該当せず	CPU性能(クロック)
69	処理に時間がかかる	CPU使用率の確認	CPU使用率が高い状態が続く	・CPU使用率 ・メモリ使用量 ・スナップショットによりコピー箇所を特定 ・データサイズを変化させて処理時間をグラフ化する	メモリのコピー処理かを特定し、コピーするサイズを確認する。	メモリを1バイトずつコピーしている。	ソース	ロジック	コードの実装方法			自プロセス	①自分のプロセスで自分のソースの範囲内	ムダな処理	コピーなどの実装が悪い	CPU	大規模になった時	該当せず	CPU性能(クロック)
70	処理に時間がかかる	CPU使用率の確認	システム全体のCPU使用率が高く、該当プロセスのCPU使用率もそこそこ高い状態のまま推移。	(1.1)トレースログでWAIT場所を特定する。 または、 (1.2)プロセスのスナップショットを採取し、PCの値を確認する。 かつ (2)プロセス間通信のIPCの場所が小さい。	プロセス間通信のIPCのバッファサイズを確認する。	通信データサイズに比較して、プロセス間通信のIPCのバッファが小さい	ソース	通信処理のパラメータ	通信処理			自プロセス	③自分のプロセスだがAPIの延長	ムラな処理	バッファサイズの不適切	通信(ソフト)	該当せず	該当せず	自分のソース
71	処理に時間がかかる	CPU使用率の確認	該当プロセスのCPU使用率が低い状態のまま推移。	(1.1)トレースログでWAIT場所を特定する。 または、 (1.2)プロセスのスナップショットを採取し、PCの値を確認する。 かつ	自分がAgentの場合、他のホスト(Managerなど)が多くのAgentの要求を受け付けていないかを確認する。 かつ	自ホストでなく、通信先の他ホスト(Manager)がビジーで自ホストの要求を待たされている。	他ホスト		システム構成全体の性能		1台のManagerにAgentが1000台のようなケースもあるので、機器構成も入手する	自プロセス	①自分のプロセスで自分のソースの範囲内	WAIT	相手の処理待ち	通信(ソフト)	相手がビジーによる処理待ち	同様の業務UP	該当せず
72	処理に時間がかかる	I/O回数を確認する。	I/O回数が多。	I/Oのデータの内容を確認する。	小さいレコードが大量に出力されている。	小さいレコードを大量に出力している。	ソース	I/O処理のロジック	I/Oの実装方法	例えば、メッセージが大量に出力されている。	ブロッキングして出力する方法もある。	自プロセス	①自分のプロセスで自分のソースの範囲内	ムダな処理	I/O処理を頻発	DISK(ソフト)	該当せず	該当せず	DISKアクセス性能
73	処理に時間がかかる	I/O回数を確認する。	I/O回数が多。	I/O時のデータのバッファサイズを確認する。	バッファサイズが小さい。	出力しようとしているデータサイズに対して、バッファサイズが小さすぎるのでI/O回数が多くなってしまふ。	ソース	I/O処理のロジック	I/Oの実装方法			自プロセス	①自分のプロセスで自分のソースの範囲内	ムラな処理	バッファサイズの不適切	DISK(ソフト)	該当せず	該当せず	DISKアクセス性能
74	件数が増加すると指数関数的に処理に時間がかかる		件数による傾向を分析する。	・グラフの傾向から該当しそうなソース箇所を洗い出す。 ・I/O回数の傾向から該当しそうなソース箇所を洗い出す。 ・システムコールトレースの発行状況から該当しそうなソース箇所を洗い出す。 該当箇所に対して、ソース解析して、回数の多い処理に関して、ループがあるかを確認する。 ログが不十分な場合には、ログ強化して該当箇所を絞り込	件数がn件の時に、n回のループの中にn回のループがある。ループがネストする処理になっている。 特にループの中にシステムコールがあると、非常に悪化する。	ループ構造	ソース	ループ処理	ループ	扱うデータ件数の規模を確認する。		自プロセス	①自分のプロセスで自分のソースの範囲内	ムダな処理	ループ中のループ	DISK(ソフト)	大規模になった時	該当せず	自分のソース
75	処理に時間がかかる	I/O時間を計測する。	I/O時間が長い。	(1)I/O時のデータのバッファサイズを確認する。 (2)該当プロセスのI/Oの読み取りバイト数と書き込みバイト数を確認する。	I/Oのデータサイズが大きい。	I/Oのデータサイズが大きすぎる。 部分書き込み可能なのに、全体を毎回書き込んでいる。	ソース	I/O処理のロジック	I/Oの実装方法			自プロセス	①自分のプロセスで自分のソースの範囲内	ムラな処理	バッファサイズの不適切	DISK(ハード)	大規模になった時	該当せず	DISKアクセス性能
76	処理が遅くなることがある。	I/O時間を計測する。	I/O時間が長い。	(1)該当プロセスがライトしているファイルは、ウイルス監視の対象ファイルになっているかを確認する。 (2)ウイルスチェックプログラムのパターンファイルの更新日時を確認する。 (3)該当プロセスのI/Oのサイズと回数を確認する。 (4)システムコールトレースを確認する。	(1)ウイルスチェック対象ファイルである。 (2)パターンファイルの更新後から遅くなった。 (3)該当プロセスの延長でウイルスチェック処理が動作するかを確認する。 (4)別のプロセスも遅くなっていないかを確認する。 (5)ウイルスチェックの対象データサイズが通常より大きい、または数が多いかを確認する。	・該当プロセスの延長でウイルスチェック処理に時間がかかっている。 ・ウイルスチェックのパターンファイルの更新によってウイルスチェックに時間がかかるようになってしまった。 ・ライトするデータがバイナリの場合、データのバターンがウイルスチェックと相性が悪い内容となっている。	システム内のPP	ウイルス監視ソフト	I/Oの延長での処理	・ウイルス監視製品が同居しているかを確認する。 ・パターンファイルが変わったことで、性能が低下してないか他のシステムなどの状況も確認する。	該当プロセスがライトするファイルがウイルス監視対象ファイルから除外する。 パターンファイルを元に戻す。	自プロセス	③自分のプロセスだがAPIの延長	ムダな処理	同居製品固有の実装ロジック	CPU	該当せず	ウイルス監視製品	ライブラリ
77	処理に時間がかかる	I/O時間を計測する。	I/O時間が長い。	システム全体のI/Oが多く、OSのI/Oキューの状況を確認する。	I/Oのキュー待ちの数が大きくなっている。 他のプロセスも大量のI/Oを発行している。	システム全体のI/O回数が多い。	サーバー	OSの統計情報	システム高負荷			自プロセス	③自分のプロセスだがAPIの延長	競合	不足	DISK(ハード)	該当せず	自分以外のプログラム	DISKアクセス性能
78	処理に時間がかかる	I/O時間を計測する。	I/O時間が長い。	システム全体のI/Oが多く、I/Oの総データ量を確認する。 ディスクがビジーに頻発しているかも確認する。	他のプロセスが大きなバッファサイズでI/Oを頻発に実施している。	システム全体のI/O量が多く、ディスクビジーの状態が続いている	サーバー DISK	OSの統計情報 DISKの統計情報	ディスクが高負荷	ディスクのログでビジーな状態かを確認。 I/O時間の計測 OSから見るI/Oの状態確認を行う。		自プロセス	③自分のプロセスだがAPIの延長	競合	不足	DISK(ハード)	DISKビジーによる処理待ち	自分以外のプログラム	DISKアクセス性能
79	処理に時間がかかる	I/O時間を計測する。	I/O時間が長い。	他のプロセスが同一のディスクの玉に大量にアクセスしていないかを確認する。	別プロセスで大量にアクセスしている。 例えば、バックアップ、ファイルコピー、ファイル転送、データベースソフトなどでRDBに大量のアクセスなど。	複数のプロセスから同一のディスクの玉にアクセスすると、SEEKが発生しやすくなり、I/O時間が間延びする。	サーバー DISK	他プロセスのI/O状況	競合			自プロセス	③自分のプロセスだがAPIの延長	競合	不足	DISK(ハード)	該当せず	自分以外のプログラム	DISKアクセス性能
80	処理に時間がかかる	I/O時間を計測する。	I/O時間が長い。	マシンとディスクの玉のハード構成を確認する。	他のマシンと同一のディスクの玉でI/Oが競合している。 他のVMと同一ディスクの玉にI/Oが競合している。	I/Oするディスクの玉が他マシン/他VMと競合しており、別システムから同一のディスクの玉に大量のI/Oが発行されている。	サーバー DISK	OSの統計情報 DISKの統計情報	競合	ディスクのログでビジーな状態かを確認。 I/O時間の計測 OSから見るI/Oの状態確認を行う。	別のマシンで同一ディスクの玉にバックアップ、ファイルコピー、ファイル転送、データベースソフトなどでRDBに大量のアクセスなどがないかを確認する	自プロセス	③自分のプロセスだがAPIの延長	競合	不足	DISK(ハード)	DISKビジーによる処理待ち	自分以外のプログラム	DISKアクセス性能
81	処理に時間がかかる	I/O時間を計測する。	I/O時間が長い。	システム全体のマシンとディスクの構成を確認する。	複数のマシンと複数のディスクの接続が手づるになっている。	複数のディスク間のケーブルが手づるで共用されており、他のマシンのI/Oが沢山あると、ケーブルビジーになってしまふ。	サーバー DISK	ハード構成	接続ケーブルが高負荷	I/O時間の計測 OSから見るI/Oの状態確認を行う。 複数マシンでディスクビジー率などを確認する。	VMなどの場合には、複数マシンとディスクのハード構成の確認。	自プロセス	③自分のプロセスだがAPIの延長	競合	不足	DISK(ハード)	DISKビジーによる処理待ち	自分以外のプログラム	DISKケーブル



82	処理に時間がかかる	I/O時間を計測する。I/O時間が長い。	自プロセス以外のプロセスでI/Oを沢山発行しているプロセスがないかを確認する。	I/Oを沢山発行しているプロセスがウイルスチェック用のプロセスである。	ウイルスチェックが実行されている間にI/O時間が間延びしている。	システム内のPP	ウイルス監視ソフト		競合	I/Oを沢山発行しているプロセスを特定する。	ウイルスソフトの動作時刻の確認。ウイルスソフトの対象の確認	自プロセス	③自分のプロセスだがAPIの延長	競合	不足	DISK(ハード)	該当せず	ウイルス監視製品	他プロセス
83	処理に時間がかかる	I/O時間を計測する。I/O時間が長い。	VMマシンの場合、ウイルス処理専用のVMを設置する運用であるかを確認する。自マシンの状況を確認しただけでは原因究明が難しい場合がある。	ウイルス処理専用のVMがあり、ウイルスチェックの時間とI/O時間が長い時の時間が一致する。	ウイルスチェックするマシンが別VMであっても、アクセスするディスクの筐体と同じなので、I/Oが競合したのと同じ現象となり、I/Oが間延びする。	同一システム内のPP	ウイルス監視ソフト		競合	VMの場合には、自マシンだけの性能情報ではディスク装置に対してのI/O負荷がわからないので、システム全体のI/Oの状況を確認する必要がある。	他VMのウイルスソフトの動作時刻の確認。ウイルスソフトの対象の確認	自プロセス	③自分のプロセスだがAPIの延長	競合	不足	DISK(ハード)	該当せず	ウイルス監視製品	他プロセス
84	処理に時間がかかる	I/O時間を計測する。I/O時間が長い。	DISKのデフラグツールで断片化状態を確認する。	断片化が進んでいる状態になっている。	断片化により、単一ファイルをアクセスする場合でも、散在する場所にアクセスする必要があり、I/Oに時間がかかる。	DISK	デフラグツールでの確認		断片化	システムとしては、徐々に遅くなる現象となる。	デフラグツールでデフラグを実行することで改善される。	自プロセス	③自分のプロセスだがAPIの延長	ムラな処理	断片化	DISK(ハード)	長時間の経過後	該当せず	DISKアクセス性能
85	処理に時間がかかる	I/O時間を計測する。I/O時間が長い。	特定のファイルが遅い場合には、該当ファイル全体をコピーして、早くなるかを確認する。	コピー後のファイルに対しては、アクセスが早くなった。	ファイルを少しずつ増分しており、1つのファイルが沢山のエクステンション構成になっている。または、クローズした後に、再度追加書きするようなアクセスを隣接するクラスタに交互にI/Oを実施すると実I/Oが多くなる。インデクスとデータが隣接クラスタになると発生する。	DISK			断片化		複数エクステンションになっているかを確認する方法はないと思われる。(確認方法があるかは)	自プロセス	③自分のプロセスだがAPIの延長	ムラな処理	断片化	DISK(ハード)	該当せず	該当せず	DISKアクセス性能
86	処理に時間がかかる	I/O時間を計測する。I/O時間が長い。	JFS/JFS2でUPが順次遅延書き込みを行っている状況下で、クラスタ単位(128Kバイト)で隣接するクラスタに交互にI/Oを実施していないかを確認する。	交互にI/Oを実施している。	隣接するクラスタに交互にI/Oを実施すると実I/Oが多くなる。インデクスとデータが隣接クラスタになると発生する。	OSの処理	OSのI/Oロジック		その他			自プロセス	③自分のプロセスだがAPIの延長	その他	OS固有の実装ロジック	DISK(ソフト)	該当せず	該当せず	OS
87	処理に時間がかかる	I/O時間を計測する。I/O時間が長い。	(1)レコードをアクセスする時の順番がディスク上の後ろから前へ向かう順番でアクセスしている。 (2)ディスクキャッシュのヒット率の確認。 例えば、削除レコードを再利用するロジックを追加した時にレコードをチェンするが、その時の順番がディスク上の場所が後ろから前にアクセスする順番になっているかを確認する。	(1)ディスク上の位置で後ろから前に向かって順次書き込みされている。 (2)ヒット率の低下。	例えば、以下のようなケースで発生する。 最初に書き込みする時には、レコード1、レコード2、レコード3、レコード4、レコード5の順番に書き込みする。そして、レコード削除する時に、 削除レコード5、削除レコード4、削除レコード3、削除レコード2、削除レコード1となり、再利用時削除レコードから書き込みすると、 レコード5、レコード4、レコード3、レコード2、レコード1という順番に並ぶことになり、DISKの書き込みも読み込みも遅くなる。	ソース	プログラムロジック		不連続処理	もしも、全件を削除し再利用する運用の場合には、1回目は早く、2回目は遅く、3回目は早く交互に性能が変わる。		自プロセス	③自分のプロセスだがAPIの延長	その他	自プロセスの延長の固有の実装ロジック	DISK(ソフト)	該当せず	該当せず	DISKアクセス性能
88	処理に時間がかかる	I/O時間を計測する。I/O時間が長い。	システム全体で多くの可変長属性のファイルにアクセスしているかを確認する。	多くの可変長属性のファイルにアクセスしている。	可変長属性のファイルの場合、OSのキャッシュのヒット率の低下に伴いI/O時間が長くなる。	ソース	I/O時のパラメタ		I/Oのキャッシュ			自プロセス	③自分のプロセスだがAPIの延長	キャッシュ	ヒット率低下	DISK(ソフト)	該当せず	該当せず	DISKキャッシュ性能
89	処理に時間がかかる	I/O時間を計測する。I/O時間が長い。	同期I/Oの場合で、I/O回数を減らす目的でまとめてI/Oを実施するように変更した時に、I/OサイズをDISKのキャッシュサイズより大きくしていないかを確認する。	I/Oサイズ > DISKのキャッシュサイズになっている。	同期I/Oの場合、DISKキャッシュサイズを超えると、DISKの実態に書き終えるまでI/Oが戻ってこなくなるので、I/Oに時間がかかるようになってしまう。 例えばバックアップソフトによっては、ファイルを更新する前に局所的なバックアップを実施するので、実I/Oが増加するため、I/O時間が長くなる。	ソース	I/O時のパラメタ		キャッシュ			自プロセス	③自分のプロセスだがAPIの延長	キャッシュ	キャッシュ不足	DISK(ハード)	該当せず	該当せず	DISKキャッシュ性能
90	処理に時間がかかる	I/O時間を計測する。I/O時間が長い。	DISKのデバイス・ドライバに関して、何かサードパーティ製のドライバがくっついていないか確認する。バなど。	サードパーティ製のバックアップ関連のドライバを使用している。		システム内のPP	バックアップソフト		I/Oの延長での処理			自プロセス	③自分のプロセスだがAPIの延長	その他	同居製品固有の実装ロジック	DISK(ソフト)	該当せず	バックアップ、ディザスタ製品	ライブラリ
91	処理に時間がかかる	I/O時間を計測する。I/O時間が長い。	同一装置内でバックアップやリストアなどのレコード長の大きいI/Oが連続して発生していないかを確認する。	バックアップまたはリストアが実行されている。	OSは、I/Oを受け付けた順番に処理するので、UPのI/Oの間にバックアップなどの大きなサイズのI/OがあるとI/O時間が間延びする。	システム内のPP	バックアップソフト		競合			自プロセス	③自分のプロセスだがAPIの延長	競合	不足	DISK(ハード)	該当せず	バックアップ、ディザスタ製品	他プロセス
92	処理に時間がかかる	I/O時間を計測する。I/O時間が長い。	DB関係の空きページ回収などの処理が実行されていないかを確認する。	同時刻に空きページ回収などの処理がスケジューリングされている。	空きページ回収のI/Oとの競合によりI/O時間が間延びする。	システム内のPP	データベース製		競合			自プロセス	③自分のプロセスだがAPIの延長	競合	不足	DISK(ハード)	該当せず	自分以外のプログラム	他プロセス
93	処理に時間がかかる	I/O時間を計測する。I/O時間が長い。	I/O時のブロック長とレコード長を確認する。	ブロック長がレコード長より短い。	1レコードの書き込み時に、I/Oが2回以上になる時がある。	ソース	I/O時のパラメタ		キャッシュ			自プロセス	③自分のプロセスだがAPIの延長	ムラな処理	バッファサイズの不適切	DISK(ソフト)	該当せず	該当せず	OS
94	処理に時間がかかる	I/O時間を計測する。I/O時間が長い。	I/Oの出力先がネットワークドライブを確認する。	ネットワークドライブになっている。	(1)ネットワークの通信速度の影響を受ける。 (2)ネットワークドライブなので、同一DISK筐体に対して、他のマシンからのI/Oが競合している場合もある。	システム全体のDISK構成	共用DISKの使われ方		ネットワーク競合		コマンド出力のリダイレクト先をネットワークドライブにすると、48倍遅い事例があった	自プロセス	③自分のプロセスだがAPIの延長	その他	ハードウェア特性	通信(ハード)	該当せず	該当せず	DISKアクセス性能
95	処理に時間がかかる	I/O時間を計測する。I/O時間が長い。	(1)非同期I/Oになっているかを確認する。 (2)ファイルサイズが2GBなどのような巨大になっている。	非同期I/Oでファイルサイズも2GBを超える。	ファイルサイズが大きくなるとI/OのFlush処理が比例して長くなる可能性がある	ソース	I/O処理のロジック		キャッシュ		オンラインに近い形で運用する場合で巨大なファイルを作成する場合には、途中でFlushなどを実施する。FlushFileBuffers()がファイルサイズに比例して時間がかかる。	自プロセス	①自分のプロセスで自分のソースの範囲内	ムラな処理	同期処理	DISK(ソフト)	該当せず	該当せず	DISKアクセス性能
96	処理に時間がかかる	I/O時間を計測する。I/O時間が長い。	ストレージのキャッシュサイズを確認する。	キャッシュが小さい。	ストレージのキャッシュが小さい	DISK	ハードスベックの確認		キャッシュ			自プロセス	③自分のプロセスだがAPIの延長	キャッシュ	キャッシュ不足	DISK(ハード)	該当せず	該当せず	DISKキャッシュ性能
97	処理に時間がかかる	I/O時間を計測する。I/O時間が長い。	ストレージのキャッシュが有効かを確認する。 例えば、キャッシュの電池が切れなども含めて。	キャッシュが無効になっている。	ストレージのキャッシュが無効になっている。	DISK	ハード障害		キャッシュ		キャッシュの電池切れなど	自プロセス	③自分のプロセスだがAPIの延長	キャッシュ	キャッシュが無効	DISK(ハード)	該当せず	該当せず	DISKキャッシュ性能
98	処理に時間がかかる	I/O時間を計測する。I/O時間が長い。 書き込み時に時間がかかる。	ウイルス対策ソフトがリアルタイムでファイルチェックを実施する設定になっているかを確認する。	リアルタイムでファイルチェックする対象になっている。	ファイルI/Oの延長でウイルスチェック処理が実行されるのでI/O時間が間延びする。	同一装置内のPP	ウイルス監視ソフト		I/Oの延長での処理	ウイルス定義ファイルが更新されるチェック時間も変わる可能性があるため、プログラムが遅くなった時刻と更新時刻を確認する。		自プロセス	③自分のプロセスだがAPIの延長	その他	自プロセスの延長の固有の実装ロジック	DISK(ソフト)	該当せず	ウイルス監視製品	DISKアクセス性能
99	処理に時間がかかる	I/O時間を計測する。I/O時間が長い。 書き込み時に時間がかかる。	DISKのRAID構成の確認をする。	RAID1構成になっている。	RAID1は、単なるミラーリングだけなので、RAIDでないDISKより遅くなる。	DISK	DISKのオプション		ハード構成			自プロセス	③自分のプロセスだがAPIの延長	その他	ハードウェア特性	DISK(ハード)	該当せず	該当せず	DISKアクセス性能
100	処理に時間がかかる	I/O時間を計測する。書き込み時に時間がかかる。	ハードウェアでのディザスタ構成かを確認する。	ハードウェアでのディザスタ構成になっている。	遠隔地にある災対に対してのI/Oの完了を待つケースがあり、I/Oが間延びする。	DISK	DISKのオプション		ハード構成			自プロセス	③自分のプロセスだがAPIの延長	その他	ハードウェア特性	DISK(ハード)	該当せず	該当せず	DISKアクセス性能
101	処理に時間がかかる	I/O時間を計測する。I/O時間が長い。 書き込み時に時間がかかる。	ソフトウェアでのディザスタ構成かを確認する。 同一マシン上にディザスタ用のソフトがあるか確認する。	ソフトウェアでのディザスタ構成になっている。	遠隔地にある災対に対してのI/Oの完了を待つケースがあり、I/Oが間延びする。	同一装置内のPP	ディザスタソフト		I/Oの延長での処理	ハードウェアの構成を見ても判らないので注意が必要。		自プロセス	③自分のプロセスだがAPIの延長	その他	同居製品固有の実装ロジック	DISK(ソフト)	該当せず	該当せず	DISKアクセス性能
102	処理に時間がかかる	I/O時間を計測する。書き込み時に時間がかかる。	I/Oが同期I/Oかを確認する。	同期I/Oになっている。	I/Oを意味もなく同期I/Oになっている。	ソース	I/O処理のオプション		I/Oの実装方法			自プロセス	①自分のプロセスで自分のソースの範囲内	ムダな処理	同期処理	DISK(ソフト)	該当せず	該当せず	DISKアクセス性能



103	処理に時間がかかる	I/O時間を計測する。	I/O時間が長い。書き込み時間に時間がかかる。	意味もなくディスクのマウント時にDIOやCIOを指定しているかを確認する。	DIOまたはCIOになっている。	意味もなくディスクのマウント時にDIOやCIOを指定している。 この指定をするこのファイルに対しては同期I/Oになってしまふ。 ・DIO(直接入出力) アプリが直接ディスクにIO ・バッファ入出力 アプリ→OS→ディスク ・キャッシュ入出力(OSバッファ入出力を包含する) アプリ→OSキャッシュ→ディスク ・CIO(コンカレント入出力) アプリ→ディスク 但し、1個のファイルに対して同時に複数箇所の書き込みが可能。 データベースプログラムのよ	OSの設定	I/Oのオプション			プログラムとオプションの組み合わせの確認。			③自分のプロセスだがAPIの延長	ムダな処理	同期処理	DISK(ソフト)	該当せず	該当せず	DISKアクセス性能
104	処理に時間がかかる	I/O時間を計測する。	I/O時間が長い。	データベースで使用するのにディスクの場合、マウント時にCIO指定をしているかを確認する。	通常のUPではCIO指定することはないが、データベースにアクセスするミドルソフトによってはCIO指定することで高速化を実現している。必要に応じてCIO指定を実施する。	通常のUPでは、プログラム作成が難しくなるので使わない。データベースで使用するのにディスクのマウント時にCIO指定をしていない。CIO指定だと複数のプロセスから同一ファイルに対して同時にI/Oが可能に	OSの設定	I/Oのオプション			プログラムとオプションの組み合わせの確認。			③自分のプロセスだがAPIの延長	その他	OS固有の実装ロジック	DISK(ソフト)	該当せず	該当せず	DISKアクセス性能
105	処理に時間がかかる	実際のデータ送受信に時間がかかっているかをログから確認する。	送受信前に時間がかかっている。	hostsファイルに大量のホスト名が記述されているかを確認する。	ホスト名が大量に記述される。	OSIによっては必ず最後のレコードまで読み込むことがあり、ホスト名解決に時間がかかることがある。	OSの設定	hostsファイル			通信環境設定		OSによっては最後まで読んでいる。例えば、何度もホスト名解決の関数を発行するプログラムかを確認する。	③自分のプロセスだがAPIの延長	その他	OS固有の実装ロジック	通信(ソフト)	大規模になった時	該当せず	OS
106	処理に時間がかかる	実際のデータ送受信に時間がかかっているかをログから確認する。	送受信前に時間がかかっている。	ホスト名解決にかかる時間を計測してDNSサーバーの性能を確認する。	DNSサーバーの性能が悪い。	名前解決に時間がかかってしまい、性能がでない。	他ホスト(DNS)状況				他ホストからの影響		例えば、何度もホスト名解決の関数を発行するプログラムかを確認する。	③自分のプロセスだがAPIの延長	その他	その他	通信(ソフト)	該当せず	他ホスト	該当せず
107	処理に時間がかかる	実際のデータ送受信に時間がかかっているかをログから確認する。	送受信前に時間がかかっている。	ホスト名解決などの関数を何度も発行していないかを確認する。	毎回ホスト名解決も毎回実施している。	ホスト名解決は、単価が高いので性能がでない。	ソース	通信処理のロジック			通信の実装方法			③自分のプロセスだがAPIの延長	その他	自プロセスの延長の固有の実装ロジック	通信(ソフト)	該当せず	他ホスト	自分のソース
108	処理に時間がかかる	実際のデータ送受信に時間がかかっているかをログから確認する。	送受信前に時間がかかっている。	DHCPサーバーの性能を確認する。	DHCPサーバーの性能が悪い。	DHCPサーバーの性能に引きずられて性能がでない	他ホスト(DHCP)状況				他ホストからの影響			③自分のプロセスだがAPIの延長	その他	その他	通信(ソフト)	該当せず	他ホスト	該当せず
109	処理に時間がかかる	実際のデータ送受信に時間がかかっているかをログから確認する。	送受信前に時間がかかっている。	回線の通信速度を確認する。	回線速度が遅い。	回線速度が遅い回線のため性能が出ない。	回線種別				通信環境設定			③自分のプロセスだがAPIの延長	その他	その他	通信(ソフト)	該当せず	他ホスト	通信回線性能(速度)
110	処理に時間がかかる	通信ログの確認。	送受信前に時間がかかっている。	データ件数が多いにも関わらず、プログラムの処理として毎回通信のオープン/クローズを実施しているかを確認する。	毎回オープン/クローズを実施している。	コネクションの接続は単価が高いため、性能がでない。	ソース	通信処理のロジック			通信の実装方法			①自分のプロセスで自分のソースの範囲内	その他	自プロセスの延長の固有の実装ロジック	通信(ソフト)	該当せず	該当せず	自分のソース
111	処理に時間がかかる	通信エラーログの確認。	送受信前に時間がかかっている。	短時間にソケットの生成/消滅を繰り返して、ポート枯渇が発生し、リトライを繰り返していないかを確認する。	短時間にソケットの生成/消滅を繰り返している。	ソケットは、クローズしてもすぐに再利用できないので、リトライでソケットが使用可能になるまでに時間がかかるので、性能がでない。	ソース	通信処理のロジック			通信の実装方法			③自分のプロセスだがAPIの延長	その他	自プロセスの延長の固有の実装ロジック	通信(ソフト)	通信エラー	該当せず	自分のソース
112	時々、時間がかかる	通信エラーログの確認。	通信エラーが発生している。	プログラムの通信エラー時のリトライ間隔を確認する。	リトライ間隔が長すぎる。	通信エラー時にリトライ間隔が長すぎるために、時間がかかっている。	ソース	通信処理のロジック			通信の実装方法			①自分のプロセスで自分のソースの範囲内	ループ	タイマー値大(リトライ)	通信(ソフト)	通信エラー	該当せず	自分のソース
113	時々、時間がかかる	通信エラーログの確認。	通信エラーが発生している。	プログラムの通信エラー時のリトライ回数を確認する。	リトライ回数が多すぎる。	通信エラー時にリトライ回数が多すぎるために、時間がかかっている。	ソース	通信処理のロジック			通信の実装方法			①自分のプロセスで自分のソースの範囲内	ループ	タイマー値小(リトライ多発)	通信(ソフト)	通信エラー	該当せず	自分のソース
114	処理が遅延する	通信の1往復の時間を計測する。	1往復に時間がかかっている。	通信相手の場所を確認する。	通信相手が非常に離れている。	回線が大容量でも、小さいデータのやり取りが多い場合には、相手の距離が遅い場合には、時間がかかる。	ネットワーク構成	通信相手までの距離			ネットワーク構成			③自分のプロセスだがAPIの延長	その他	ハードウェア特性	通信(ハード)	該当せず	該当せず	通信回線性能(距離)
115	処理が遅延する	一連の通信処理の時間を計測する。	一連の処理に時間がかかっている。	パケットキャプチャーで通信相手との間のパケットを確認する。	パケットの再送などが頻繁に発生している。	通信品質が悪くAPIの延長でリトライが実施されている。	ネットワーク構成	回線品質			ネットワーク回線品質			③自分のプロセスだがAPIの延長	その他	ハードウェア特性	通信(ハード)	該当せず	該当せず	通信回線性能(品質)
116	処理が遅延する	一連の通信処理の時間を計測する。	一連の処理に時間がかかっている。	パケットキャプチャーで通信相手との間のパケットを確認する。	パケットの再送などが頻繁に発生している。	回線の許容を超えるデータになり、パケットのロスが発生し、再送が実施されている。	ネットワーク構成	システム全体の通信量			ネットワーク通信料の見積もり			③自分のプロセスだがAPIの延長	その他	ハードウェア特性	通信(ハード)	該当せず	該当せず	通信回線性能(帯域)
117	処理が遅延する	一連の通信処理の時間を計測する。	一連の処理に時間がかかっている。	システム全体の通信量と、自プロセスの通信量を計測する。	自プロセスの通信量は少ないが、システム全体の通信量が多い。	裏で大量の通信を実施するプロセスがあるために、遅延している。 例えば、ファイル転送プログラムが複数起動されている。NASにアクセスするプログラムが複数ある。ネットワーク経由でバックアップ/リストアしている。	同一装置内のPP	ファイル転送ソフトなど			競合	システム全体の状況を把握する。	製品によっては、ファイル転送時に沢山のコネクションを使用してパラレルにデータの送受信を実施している。	③自分のプロセスだがAPIの延長	競合	不足	通信(ハード)	該当せず	バックアップ、ディザスタ製品	通信回線性能(帯域)
118	処理が遅延する	一連の通信処理の時間を計測する。	一連の処理に時間がかかっている。	CPU使用率を確認する。	CPU使用率が100%に近い状態で推移している。	CPU使用率が高いため、通信するために必要なCPUが割り当てられていないために、通信が遅くなっている。	システム全体のプロセス状況				競合			③自分のプロセスだがAPIの延長	競合	不足	CPU	該当せず	自分以外のプログラム	CPU性能(コア数)
119	処理が遅延する	一連の通信処理の時間を計測する。	一連の処理に時間がかかっている。	自プロセスの通信量を計測する。	通信量が非常に大きい。	通信するデータ量が大きすぎる	ソース	通信処理のロジック			通信の実装方法			①自分のプロセスで自分のソースの範囲内	その他	自プロセスの延長の固有の実装ロジック	通信(ソフト)	大規模になった時	該当せず	通信回線性能(帯域)
120	処理が遅延する	一連の通信処理の時間を計測する。	一連の処理に時間がかかっている。	通信相手のシステムの状態を確認する。相手システムのCPU使用率や通信性能など。	通信相手側のプログラムの反応が悪い状態にある。	相手の処理の遅さに引きずられて一連の通信が遅くなっている。	他ホストのプロセス状況				他ホストからの影響		例えば、相手のシステムの負荷が高く、データの受信処理で時間がかかっているなど。	③自分のプロセスだがAPIの延長	その他	自プロセスの延長の固有の実装ロジック	通信(ソフト)	相手がビジーによる処理待ち	他ホスト	他ホスト
121	処理が遅延する	通信スピードを確認する。1秒当たり何バイト通信できている。	通信スピードが遅い。	プログラムが指定している通信バッファのサイズを確認する。	バッファサイズが小さい。	大きいデータの場合には、バッファサイズが小さいと通信スピードが確保できない。	ソース	通信処理のロジック			通信の実装方法			③自分のプロセスだがAPIの延長	ムラな処理	バッファサイズの不適切	通信(ソフト)	大規模になった時	該当せず	通信回線性能(距離)
122	処理が遅延する	通信スピードを確認する。1秒当たり何バイト通信できている。	通信スピードが遅い。	回線の種類と帯域を確認する。	通信帯域が小さい。	大きいデータの場合には、通信帯域が小さいと通信時間がかかる。	ネットワーク構成	ネットワーク帯域			ネットワーク回線種別			③自分のプロセスだがAPIの延長	その他	ハードウェア特性	通信(ハード)	大規模になった時	該当せず	通信回線性能(帯域)
123	処理が遅延する	通信スピードを確認する。1秒当たり何バイト通信できている。	通信スピードが遅い。	パケットキャプチャーで通信相手との間のパケットを確認する。	該当するマシンが大量の件数の受信処理がある。	DDOS攻撃を受けている。OSが受信処理で一杯になっている。	システム全体のプロセス状況	その他			競合			⑧その他	競合	OS固有の実装ロジック	CPU	他の業務UPの不良	他ホスト	OS

# 付録.性能トラブル時の分析観点

誰が(Who)		どうした(How)		目的語(What)	いつ(When)	誰に/誰によって(Whom)	どこで(Where)
原因となるプロセス(ソース)の特定		原因となる動作の特定		原因となる資源の特定	原因となるきっかけ	原因を誘発したもの	発生場所
大項目	中項目	大項目	中項目	中項目	中項目	中項目	中項目
1	①自分のプロセスで自分のソースの範囲内	ループ	無限ループ	CPU	システムコールエラー	暗号化製品	自分のソース
2	②自分のプロセスだがライブラリの延長		スピルループ	メモリ	I/Oエラー	ウイルス監視製品	ライブラリ
3	③自分のプロセスだがAPIの延長		長時間ループ	時間	通信エラー	バックアップ、ディザスタ製品	OS
4	④他プロセスだが自プロセスと同じソース		ダイナミックループ	DISK(ソフト)	一時的なエラー	ファイル転送製品	他プロセス
5	⑤他プロセスだが自プロセスと違うソース		リトライ	DISK(ハード)	恒久エラー	DB製品	他ホスト
6	⑥他ホストプロセスだが自プロセスと同じソース	WAIT	デッドロック	通信(ソフト)	相手がビジーによるリトライ	自分以外のプログラム	CPU性能(クロック)
7	⑦他ホストプロセスだが自プロセスと違うソース		タイマー値大	通信(ハード)	相手がビジーによる処理待ち	同様の業務UP	CPU性能(コア数)
8	⑧その他		タイマー値大(リトライ)	排他資源	相手がnot ready時のリトライ	他の業務UP	メモリ性能
9			タイマー値小(リトライ多発)	その他のハード	DISKビジーによる処理待ち	他ホスト	メモリサイズ
10			相手の処理待ち	その他	排他エラー	自プロセス	DISKアクセス性能
11			無限待ち		大規模になった時	該当せず	DISKキャッシュ性能
12		キャッシュ	ヒット率低下		長時間の経過後		DISKケーブル
13			キャッシュ不足		他の業務UPの不良		通信回線性能(速度)
14			遅いキャッシュ		該当せず		通信回線性能(帯域)
15			キャッシュが無効				通信回線性能(距離)
16		ムダな処理	ループ中のループ				ハード障害
17			乱雑				該当せず
18			シーケンシャル処理				通信回線性能(品質)
19			輻輳				
20			メモリ確保関数を頻発				
21			I/O処理を頻発				
22			同期処理				
23			コピーの実装が悪い				
24			ソート、コンペアの実装が悪い				
25			サーチ、インデックスの実装が悪い				
26			キューなどの実装が悪い				
27		ムラな処理	プライオリティ				
28			断片化				
29			逆読み				
30			バッファサイズの不適切				
31			型の不一致など				
32		競合	不足				
33		その他	同居製品固有の実装ロジック				
34			OS固有の実装ロジック				
35			ライブラリ固有の実装ロジック				
36			自プロセスの延長の固有の実装ロジック				
37			OS不良				
38			他製品不良				
39			ハードウェア特性				
40			その他				