

The InterSystems logo, featuring the word "INTERSYSTEMS" in a blue, sans-serif font with a registered trademark symbol.

You'll make breakthroughs

A small rectangular logo with a blue border. Inside, it says "MAKING BREAKTHROUGHS" in small blue letters, "InterSystems Symposia" in white, and "2014" in orange.

Caché モニタリングツールによるシステム監視

インターシステムズジャパン株式会社
カスタマーサポート部
掛地 めぐみ



You'll make breakthroughs

アジェンダ

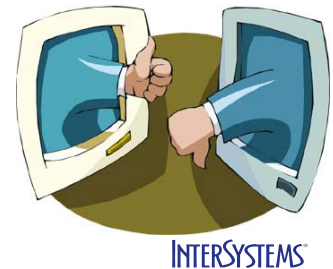
- ① モニタリング(監視)の目的
- ② 管理ポータル
- ③ パフォーマンスモニタリングツール
- ④ Cachéモニタリングツール
- ⑤ サードパーティモニタリングツール

The InterSystems logo, featuring the word "INTERSYSTEMS" in a blue, sans-serif font with a registered trademark symbol.

①モニタリング(監視)の目的

このセッションとモニタリングの目的

- システムを安全かつ健全に動作させるために、システムの状態を正確に把握することが重要
- 既知の問題の調査・予防保全、両方の目的で用いる
- Caché が提供するツールを有効に使用し、監視を行うことで効率の良い運用が可能
- 問題検出の簡素化
- 迅速かつ正確なアラート通知
- シームレスなトラブルシューティング、診断、および処置



2

②管理ポータル システムダッシュボード

- システムダッシュボード は、システムパフォーマンスの指標を表示
- システムオペレーション → システムダッシュボード

システムパフォーマンス	
グローバル/秒:	863.00
グローバル参照数:	2,939,787
グローバル更新数:	276,276
ルーチン参照:	428,871
論理要求:	864,894
ディスク読込:	7,395
ディスク書出:	828
キャッシュ効率:	357.51

ECPとシャドウイング	
アプリケーションサーバ:	正常
アプリケーションサーバトラフィック:	0.00
データサーバ:	正常
データサーバトラフィック:	0.00
シャドウソース:	正常
シャドウサーバ:	

システム時刻	
システム稼働時間:	0d 3h 36m
最終バックアップ:	なし

システム使用	
データベース容量:	正常
データベースジャーナル:	正常
ジャーナル空き:	正常
ジャーナルエントリ:	29,123
ロックテーブル:	正常
ライトデーモン:	正常

トランザクション	
プロセス:	16
CSPセッション:	2
最もアクティブなプロセス:	
プロセス	コマンド
3604	173,655
5404	147,900
4440	4,845
11456	3,315
5400	3,060

エラーと警告	
深刻な警告:	0
アプリケーションエラー:	0

ライセンス	
ライセンス制限:	1,000
現在のライセンス使用:	
ライセンス使用最高値:	

タスクマネージャ		
実行予定タスク	時刻	ステータス
ジャーナルファイル削除	00:30	Scheduled
タスク削除	01:00	Scheduled
エラーとログファイルを削除	01:00	Scheduled
ロギングアクティビティチェック	01:00	Scheduled
バックアップログの削除	01:30	Scheduled

INTERSYSTEMS

3

You'll make breakthroughs

②管理ポータル システム使用

- 管理ポータル:
システムオペレーション → システム使用
- 現在のシステムの使用状況を表示可能
- 稼働中の統計情報を確認するときに利用
- ^GLOSTATユーティリティでも同様の情報を取得可能

システム負荷なし

システム使用	
グローバル参照(すべて)	3,629,111
グローバル更新参照	287,969
ルーチン呼び出し	459,774
ルーチンバッファ読み込みと保存	1,503
論理ブロック要求	900,149
ブロック読み込み	7,436
ブロック書き込み	915
WIJ書き込みプロセス	1,296
ジャーナルエントリ	29,285
ジャーナルブロック書き込み	50
ルーチン行	77,403,795
最終更新	2014-05-12 14:16:43.766
更新間隔 (秒): <input type="text" value="60"/> <input type="button" value="更新"/> <input type="button" value="クリア"/>	

INTERSYSTEMS

4

You'll make breakthroughs

③パフォーマンスモニタリングツール

- ^GLOSTAT (グローバル活動の統計情報を収集)
- ^mgstat (基本的なパフォーマンス・データを収集)
- ^PERFMON (パフォーマンスモニタ)
- ^%SYS.MONLBL (ルーチン行単位の監視モニタ)
- ^PROFILE (ルーチン／クラスのパフォーマンス測定)

システムやアプリケーションに問題が発生した時や
システムのパフォーマンスチューニング時に使用

INTERSYSTEMS

5

You'll make breakthroughs

③パフォーマンスモニタリングツール

^GLOSTAT

- ブロックタイプ単位で統計情報を表示
- 統計情報を収集する際は、統計値をリセットする
その後、秒数を指定して情報を収集
- 特に、以下の2つの情報に注目すると便利
 - キャッシュ効率 (Cache Efficiency) の大小
 - Block reads/writesの大小

キャッシュ効率 (Cache Efficiency) =
グローバル参照 / (物理読込+書出)

Statistics	Local	Remote	Total
Global references (all):	3,683,715	0	3,683,715
Global update references:	294,783	0	294,783
Routine calls:	500,782	0	500,782
Routine buffer loads & saves:	1,585	0	1,585
Routine commands:			78,509,782
Routine not cached:			2,142
Logical block requests:	932,143	n/a	932,143
Block reads:	7,443	0	7,443
Block writes:	1,041	n/a	1,041
WIJ writes:	1,605	n/a	1,605
Cache Efficiency:	434	no gets	
Journal Entries:	29,536	n/a	29,536
Journal Block Writes:	50	n/a	50

定常時の動きを把握
しておくで便利

システム負荷なし

INTERSYSTEMS

③パフォーマンスモニタリングツール

^mgstat

GLOSTATでシステム全体の状況を確認後は・・・①

問題となる期間について時系列でさらにグローバル
アクセスを詳細に調査する場合

mgstatを利用

“問題となるアクセスを行っている ルーチン”や“問題
となるアクセスの対象グローバル”を特定する場合

PREFMONを利用

- システム統計情報を収集
 - ^mgstatはバックグラウンドで回数を指定して実行する
実行例) `job ^mgstat(秒数, 回数, ファイル名)`
 - 秒数: 統計時間を秒数で指定する ※推奨指定は、2～5秒
 - 回数: 統計情報を収集する回数を指定する
 - ✓ 例えば、2秒毎24時間連続で計測したい場合の指定は、43200回となる (24時間は86400秒)
 - ファイル名: 出力用ファイルパスと名前を指定する
 - 出力結果ファイルを、Excelなどでグラフ化してパフォーマンス解析に利用

システム負荷なし

INTERSYSTEMS

7

You'll make breakthroughs

③パフォーマンスモニタリングツール

^PERFMON

^GLOSTATでシステム全体の状況を確認後は・・・②

問題となる期間について時系列でさらにグローバルアクセスを詳細に調査する場合



mgstatを利用

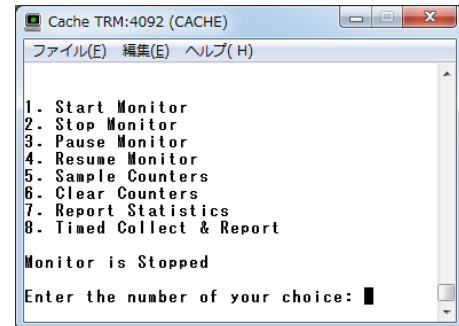
“問題となるアクセスを行っている ルーチン”や“問題となるアクセスの対象グローバル”を特定する場合



PERFMONを利用

- Cachéの動作に関する統計情報を表示
どのような頻度で個々のルーチン およびグローバルがアクセスされているか
- 統計情報収集に伴い、パフォーマンスのオーバーヘッドが生じるので注意が必要

システム負荷あり



INTERSYSTEMS

8

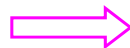
You'll make breakthroughs

③パフォーマンスモニタリングツール

^%SYS.MONLBL

^PERFMONでアクセス頻度の高いグローバルorルーチンを特定後は・・・

特定ルーチンに対して各行実行時のアクセス状態を調査する場合



%SYS.MONLBL を利用

グローバルアクセス、ルーチンアクセスに対してCachéのキャッシュ効率を確認する場合



GLOSTAT, mgstat を利用

- ^%SYS.MONLBLは、特定したルーチンの各行の実行時間やデータベースへのアクセス数を計測するツール
 - ルーチン単体レベルで行毎の処理時間が計測できるため、開発時に利用可能
 - また^PERFMON実行後、何らかの問題発生原因として予測されたルーチンに対しても、どの行の処理で実行時間がかかるかを特定可能
- ^PERFMONと同様に、実行時パフォーマンスのオーバーヘッドが生じるので注意が必要

システム負荷あり

INTERSYSTEMS

9

③パフォーマンスモニタリングツール

^PROFILE

- ^PROFILEは、開発者がアプリケーションのパフォーマンス分析に利用できる
- 以下のような構成で実行される
 - データを収集し、ルーチンレベルで並べ替え、最も“機能”しているルーチンやクラスを特定できる
 - 詳細レベルでデータ(サブルーチン、プロシージャ、および個別の行)の収集、および表示が必要なルーチンを選択し、個別のルーチンへの“ドリルダウン”を可能にする
- 情報収集の為、共有メモリを使用する

システム負荷あり

	RtnLine	Time	CPU	RtnLoad	GloRef	GloSet
1.	41.62%	0.45%	0.00%	2.83%	8.10%	0.00%
2.	%SYS.ProcessQuery.1.INT (CACHE\$SYS)	1.14%	33.33%	14.51%	4.35%	0.00%
3.	%Library.ResultSet.1.INT (CACHELIB)	3.19%	44.68%	1.94%	19.19%	27.27%
4.	%SYS.Database.1.INT (CACHE\$SYS)	0.29%	0.00%	51.39%	1.50%	0.00%
5.	%SS.INT (CACHE\$SYS)	0.19%	0.00%	4.24%	3.60%	0.00%
11.	%SYS.Monitor.SystemSensors.1.INT (CACHE\$SYS)	0.02%	0.00%	0.21%	1.95%	0.00%

10

You'll make breakthroughs

④Cachéモニタリングツール

- Cachéモニタ(^MONMGR)
- Caché システムモニタ(^%SYSMONMGR)
 - システムモニタ
 - ヘルスマニタ
 - アプリケーションモニタ
 - 履歴モニタ

より快適な環境で運用するために使用
問題が起こるのを防ぐため、もしくは問題を早期発見するために使用

INTERSYSTEMS

11

You'll make breakthroughs

④ Caché モニタリングツール

^MONMGR (Caché モニタ)

- 定義された時間間隔でコンソールログ(cconsole.log)を定期的にスキャンして、構成された深刻度レベルのエントリを検出し、アラートログ(alerts.log) に書き込むか、もしくは指定された受信者に電子メールを送信
- 既定の動作
 - Caché モニタは、インスタンスの稼働中は常に稼働
 - コンソールログは 10 秒ごとにスキャンされる
 - 通知は、深刻度 2 (重大) および深刻度 3 (致命的) のコンソールログのエントリに対して生成される
 - 通知はアラートログに書き込まれる
 - Caché の開始で、既存のアラートログは削除される

```
Cache TRM:12948 (^MONMGR)
ファイル(F) 編集(E) ヘルプ(H)
%SYS>d ^MONMGR

1) Start/Stop/Update MONITOR
2) Manage MONITOR Options
3) Exit

Option? 2

1) Set Monitor Interval
2) Set Alert Level
3) Manage Email Options
4) Exit

Option?
```

INTERSYSTEMS

12

You'll make breakthroughs

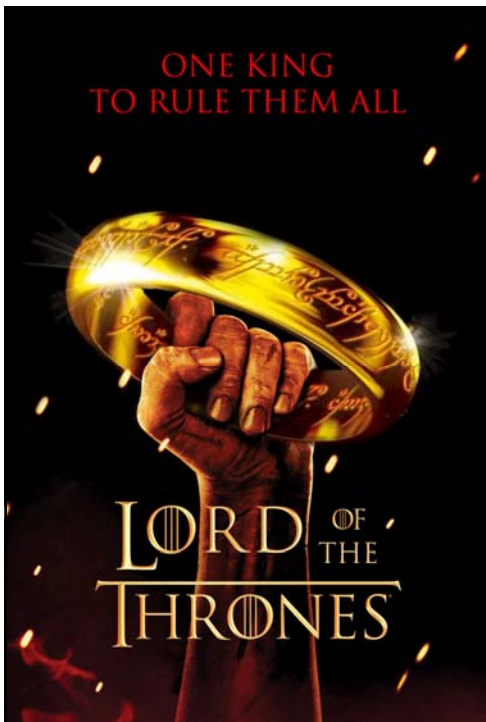
④ Caché モニタリングツール

Caché2012 以前の Caché モニタリングツール

- Caché システムモニタ
 - Caché モニタ (^MONMGR)
- Caché アプリケーションモニタ
 - ^%MONAPPMGR ルーチンで管理
- Caché System ヘルスモニタ
 - ^MONHEALTHMGR で管理
- Caché 履歴モニタ
 - ^%MONAPPMGR ルーチンで管理

INTERSYSTEMS

13



You'll make breakthroughs

Caché2013から、
^%SYSMONMGR で
システムモニタリングツールが
まとめて管理できるようになりました！

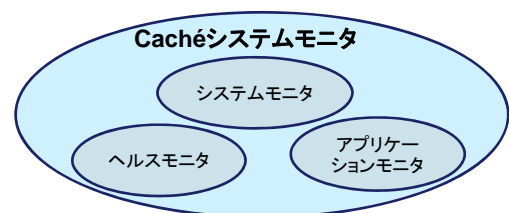
INTERSYSTEMS

You'll make breakthroughs

④ Caché モニタリングツール : Cachéシステムモニタ

Cachéシステムモニタのコンポーネント(^%SYSMONMGR)

- システムモニタ
 - システムの状態やリソース使用量を監視
 - 固定のステータスおよびしきい値に基づき、メッセージをログに出力
- ヘルスモニタ
 - 特定の期間に様々なメトリックの値を監視
 - 構成済みの閾値、またはその期間の通常値と比較し、メッセージをログに出力
- アプリケーションモニタ
 - 主要なシステムメトリックを収集し、ローカルネームスペースのグローバルに保存
 - ユーザ定義のアラートで評価し、メールを送信もしくはクラスメソッドを実行
- 履歴モニタ
 - 一般的なパフォーマンスやシステムの使用率に関するデータをデータベースに保存
 - ✓ パフォーマンスの問題(基準値からの逸脱)や長期的なシステム使用の分析に役立てる



これは CACHE モニタ (^MONMGR) とは異なります！！

INTERSYSTEMS

You'll make breakthroughs

④ Caché モニタリングツール : Cachéシステムモニタ

Caché システムモニタのフレームワーク

Cachéシステムモニタは以下のクラスを順に実行する

1. センサクラス: GetSensors()

- センサクラスは、一定間隔ごと(既定: 30秒)にメトリック情報を収集する

メトリック情報の取得

2. サブスクライバクラス: Readings()

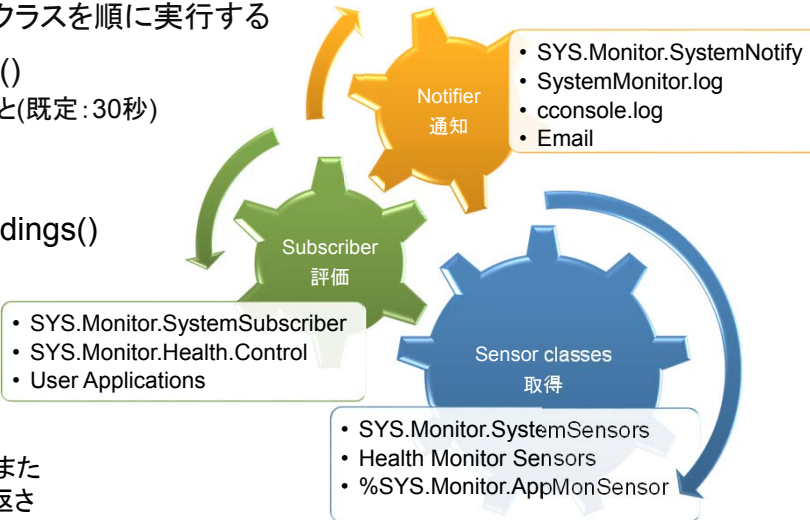
- サブスクライバクラスは、メトリックを評価し、通知を生成する

メトリック情報の評価

3. 通知クラス: Post()

- 通知クラスは、センサクラスまたはサブスクライバクラスから返された通知を、各アラートメソッドに渡す

アラートの生成



INTERSYSTEMS

16

You'll make breakthroughs

④ Caché モニタリングツール: Cachéシステムモニタ : サブスクライバクラス

センサクラス * メトリック情報の取得

センサクラスは①または②の青字の処理を行う

- ① **メトリック値の取得** ⇒ **メトリック値の評価** ⇒ **アラートの生成(通知)**
 - ② **それ自体の評価を実行して、それ自体のアラートを生成(通知)**
- ※ `%SYS.Monitor.AppMonSensor` : アプリケーションモニタはこちら

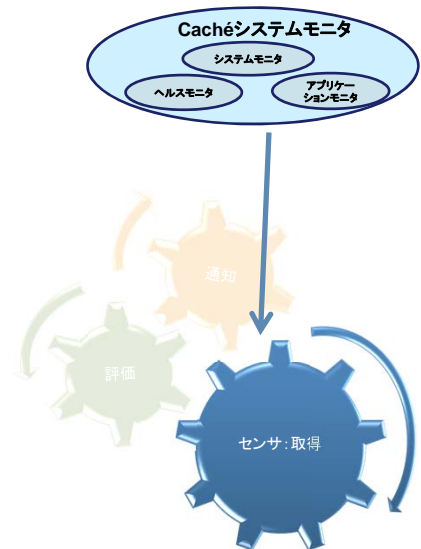
- センサクラスは、既定では30秒毎にメトリック値を取得する
- Cachéシステムモニタによってサブスクライバクラスに渡されるメトリックの名前と値の組み合わせを返す

✓ システムセンサ

- さまざまなシステムセンサが事前定義されている。例えば...
 - データベースディスク空き容量
 - 使用中のロック・テーブルの割合
 - CPU使用率 などなど...

✓ カスタムセンサ

- ユーザ定義のカスタムセンサが構成可能。例えば...
 - # 時間単位で焼けるピザの数...
 - # 分単位に生成される部品の数... などなど...



INTERSYSTEMS

17

You'll make breakthroughs

④ Caché モニタリングツール: Cachéシステムモニタ : サブスクライバクラス

サブスクライバクラス * メトリック情報の評価

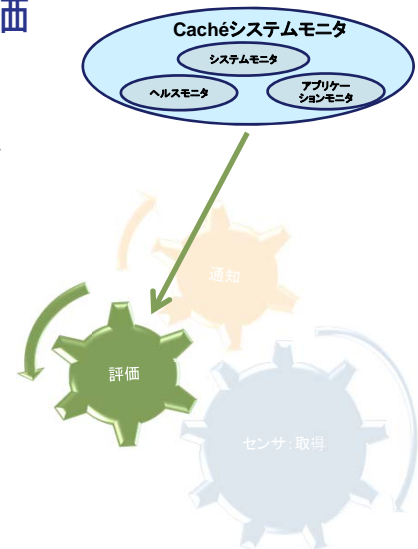
- 固定値による評価
 - 事前定義された最大値および警告値を超えたとき
 - 例: 温度が100度を超えたら常にアラートを出す
- 統計値による評価
 - 特定期間に収集されたサンプル値により評価された通常値の範囲を逸脱したとき

例: 複数サンプル値より、通常値は85~95度と評価



95度を上回った時アラートを出す

85度を下回ったらアラートを出す



INTERSYSTEMS®

18

You'll make breakthroughs

④ Caché モニタリングツール: サブスクライバクラス

固定値による評価

- システムモニタ
- ヘルスマニタの一部

メトリック	説明		通知ルール						
データベース容量	データベース・ディレクトリの空き領域		<ul style="list-style-type: none">● < 250 MB — 警告● < 50 MB — アラート● > 250 (警告 /アラート後) — OK						
ジャーナル空き	ジャーナル・ディレクトリの空き領域		<ul style="list-style-type: none">● < 250 MB — 警告● < 50 MB — アラート● > 250 (警告 /アラート後) — OK						
ページング	使用割合	センサ・オブジェクト	センサ・アイテム	説明	ベース値	最大値	乗数	警告値	警告乗数
		CSPUrlLatency	IP_address.port	IP_address.port/csp/sys/UtilHome.csp からの応答の取得に要した時間 (ミリ秒)。	1000	5000	—	3000	—
		CSPGatewayLatency	IP_address.port	CSP センサ・オブジェクトによって表されるメトリックのフェッチ時に、リストされている CSP ゲートウェイ・サーバ からの応答の取得に要した時間 (ミリ秒)。	1000	2000	—	1000	—
		DBLatency	database_server	リストにあるマウントされたデータベースからのランダム読み取りが完了するまでに要した時間 (ミリ秒)。	1000	3000	—	1000	—

19

④ Caché モニタリングツール : サブスクライバクラス : システムモニタ

固定値による評価: システムモニタ

- システムモニタ
 - システムの状態やリソース使用量を一定間隔で継続取得し、固定のステータスおよびしきい値に基づき、メッセージをログに出力する
 - センサクラスを30秒間隔で呼び出す
 - ✓ 間隔は変更可能
 - 定義済みのシステム通知ルールを使用
 - ✓ しきい値の変更は不可
 - ユーザ独自のセンサクラス、サブスクライバクラス、および通知クラスを定義することも可能
- ヘルスモニタの一部(後述)

INTERSYSTEMS™

20

④ Caché モニタリングツール : サブスクライバクラス : ヘルスモニタ

統計値による評価

- ヘルスモニタ
 - 過去の傾向(統計値: グラフ)を元にメッセージを出す
- 稼働中のCachéインスタンスを監視する
- 重要なメトリック(監視値)は、「正常範囲(通常値)」と判断できる境界内に収まる必要がある
- 過去の傾向を把握するため、特定の期間サンプリングを行う
- WECO 統計的確率規則 ([Western Electric Rules](#)) に基づく規則を使用

INTERSYSTEMS™

21

④ Caché モニタリングツール : サブスクライバクラス : ヘルスモニタ

アラート閾値: WECO 統計的確率規則 (Western Electric Rules)

どの点でも1 点が中心線から +3 σ : SEVERE	+3 σ
連続した3点のうち2点が中心線から +2 σ : WARNING	+2 σ
連続した5点のうち4点が中心線から +1 σ : WARNING	+1 σ
連続した8点が中心線から同じ側: WARNING	平均
連続した8点が中心線から同じ側: WARNING	-1 σ
連続した5点のうち4点が中心線から -1 σ : WARNING	-2 σ
連続した3点のうち2点が中心線から -2 σ : WARNING	-3 σ
どの点でも1 点が中心線から -3 σ : SEVERE	

※ σ : 標準偏差

INTERSYSTEMS

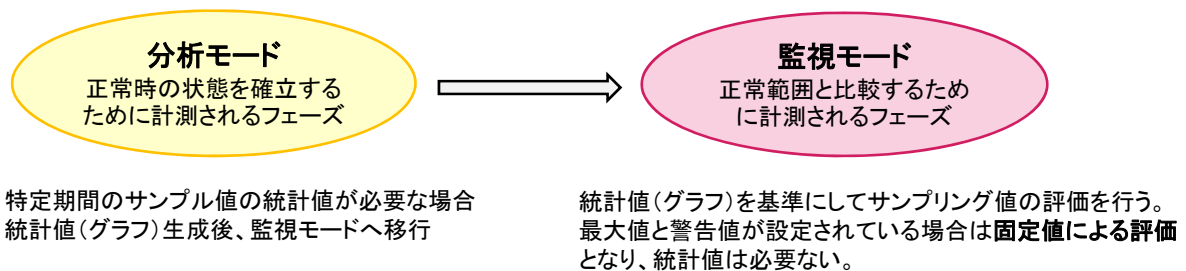
22

④ Caché モニタリングツール : サブスクライバクラス : ヘルスモニタ

正常範囲(通常値)とは？

- 異常な状態を見極めるためには、対象メトリック値が正常な範囲(通常値)であるかを知る必要がある。
- 1回のみのサンプリングでは決定できない。

ヘルスモニタは、2つのモードのいずれかで動作する



INTERSYSTEMS

23

You'll make breakthroughs

④ Caché モニタリングツール : サブスクライバクラス : ヘルスモニタ

分析モード: 期間単位の統計値(グラフ)を生成

- 期間単位の統計値の作成が可能、例えば...
 - 午前 9:00 ~ 午前 11:30 – ログイン多数(高負荷)
 - 午後 4:15 ~ 午後 6:00 – ログアウト多数(高負荷)
 - 午後 9:00 ~ 午後 11:59 – 稼働なし(低負荷)

既定のヘルス・モニタの期間

午前 0:15 ~ 午前 2:45	午前 3:00 ~ 午前 6:00	午前 6:15 ~ 午前 8:45
午前 9:00 ~ 午前 11:30	午前 11:45 ~ 午後 1:15	午後 1:30 ~ 午後 4:00
午後 4:15 ~ 午後 6:00	午後 6:15 ~ 午後 8:45	午後 9:00 ~ 午後 11:59

- 特別なスケジュールの設定が可能:
 - 週末、月末、年末
 - 四半期末
 - 休日

- 既定では、63 の週次の繰り返し期間が用意されている。
※分割期間(9) * 7日間(1週間)

ID	Description
1	Monday 00:15 - 02:45
2	Monday 03:00 - 06:00
3	Monday 06:15 - 08:45
4	Monday 09:00 - 11:30
5	Monday 11:45 - 13:15
6	Monday 13:30 - 16:00
7	Monday 16:15 - 18:00
8	Monday 18:15 - 20:45
9	Monday 21:00 - 23:59
10	Tuesday 00:15 - 02:45
11	Tuesday 03:00 - 06:00
12	Tuesday 06:15 - 08:45

全ての期間で同じ統計結果になるわけではない！

INTERSYSTEMS

24

You'll make breakthroughs

④ Caché モニタリングツール : サブスクライバクラス : ヘルスモニタ

分析モード: 統計値(グラフ)生成におけるルール

- 指定された週、月、四半期、または年ごとの期間に、30 秒の間隔でサンプリングを行う
- 統計値の信頼性を確保するために、分析モードでは最低 13 サンプルを取得する必要がある(1 回の期間で有効な 13 サンプルが取得されるまで、センサは分析モードとなる)

例: 最大値は以下のどちらかの式で算出

- ① センサオブジェクトに最大値がない場合、
(以下の 1 と 2 で大きい方) * 乗数
 1. 平均値 + 3σ(標準偏差)
 2. 統計値の最大単一値 + 1σ(標準偏差)
- ② 最大値も乗数もない場合、以下のうちの大きい方の値
 1. ベース値(又は平均値の大きい方) + 3σ(標準偏差)
 2. 統計値の最大単一値

センサ・オブジェクト	センサ・アイテム	ベース値	最大値	乗数	警告値	警告乗数
CPUUsage	job_type	50	90	—	80	—
CPUUsage		50	85	—	75	—
DBReads	database_server	1024	—	2	—	1.6

```

Cache TRM:12712 (CACHE)
ファイル(F) 編集(E) ヘルプ(H)
6) Exit

Option? 2
Sensor (*=All)? * => PhysicalBlockReadsPerMin
Item (*=All)? * =>
Period (*=All,?=Select)? * =>
Number of records to display? 20 =>
Num Period Sensor Mean Sigma
1 4 PhysicalBlockReadsPerMin 348.9 934.9
2 6 PhysicalBlockReadsPerMin 31.2 35.2
3 42 PhysicalBlockReadsPerMin 448.2 1289
Display detail of which record (0=Stop)? => 1

Period: Monday 09:00 - 11:30
Sensor: PhysicalBlockReadsPerMin Item: -
Mean: 348.9
Standard Dev: 934.9
Alert: 9285.8
Warn: 5932.8
Version: Cache for Windows (x86-64) 2014.1 (Build 608U) Tue
:23:33 EDT
Readings: 305, 145, 40, 20, 3, 89, 45, 317, 245, 39, 3708,
66, 96, 113, 3
Display detail of which record (0=Stop)? =>
  
```

INTERSYSTEMS

25

You'll make breakthroughs

Caché ヘルス・モニタのセンサ・オブジェクト

センサ・オブジェクト	センサ・アイテム	説明	ベース 値	最大 値	乗 数	警告 値	警告 乗数
CPUProc	job_type	リストされたジョブ・タイプのすべてのプロセスによる CPU 使用率 (%) の集計。	50	90	—	80	—
CPUUsage		システムの CPU 使用率 (%)。	50	85	—	75	—
CSPSessions	IP_address:port	リストされている CSP ゲートウェイ・サーバ上のアクティブな CSP セッションの 数。	100	—	2	—	1.6
CSPActivity	IP_address:port	リストされている CSP ゲートウェイ・サーバへの 1 分あたりの要求数。	100	—	2	—	1.6
CSPActualConnections	IP_address:port	リストされている CSP ゲートウェイ・サーバ上で作成された接続の数。	100	—	2	—	1.6
CSPInUseConnections	IP_address:port	リストされている CSP ゲートウェイ・サーバへの現在アクティブな接続の数。	100	—	2	—	1.6
CSPPrivateConnections	IP_address:port	リストされている CSP ゲートウェイ・サーバへのプライベート接続の数。	100	—	2	—	1.6
CSPUrlLatency	IP_address:port	IP_address:port/csp/sys/UtilHome.csp からの応答の取得に要した時間 (ミリ秒)。	1000	5000	—	3000	—
CSPGatewayLatency	IP_address:port	CSP センサ・オブジェクトによって表されるメトリックのフェッチ時に、リストされて いる CSP ゲートウェイ・サーバからの応答の取得に要した時間 (ミリ秒)。	1000	2000	—	1000	—
DBLatency	database_server	リストにあるマウントされたデータベースからのランダム読み取りが完了するまで に要した時間 (ミリ秒)。	1000	3000	—	1000	—
DBReads	database_server	リストにあるマウントされたデータベースからの 1 分あたりの読み取り数。	1024	—	2	—	1.6

INTERSYSTEMS

26

You'll make breakthroughs

④ Caché モニタリングツール : サブスクライバクラス : ヘルスモニタ

監視モード

- 必要なグラフが存在する場合やグラフが不要な場合、監視モードになる
- 3 分毎に各6つのサンプル値が平均化され、1つの読み取り値が生成 (= 評価対象の値)
- 評価に使用されるサブスクライバクラス

統計値なしの場合 (最大値と警告値が予め設定):

システムモニタの
サブスクライバクラス

統計値 (グラフ) ありの場合:

ヘルスモニタの
サブスクライバクラス

システム起動時に開始のシステムモニタにより呼び出される為、ヘルスモニタが有効でない場合にもアラートと警告は生成される

- 最近コンソールログによく見かける以下のメッセージは何？
→ グラフを必要としない (アラート・警告値を含む) ヘルスモニタが出しているメッセージ

```
05/18/14-19:52:50:886 (4612) 1 [SYSTEM MONITOR] DBLatency(c:\intersystems\symposia\mgr\cachetemp\)  
Warning: DBLatency = 1173 ( Warnvalue is 1000).  
05/20/14-14:04:03:977 (2548) 1 [SYSTEM MONITOR] CPUUsage Warning: CPUUsage = 78 ( Warnvalue is 75).
```

INTERSYSTEMS

27

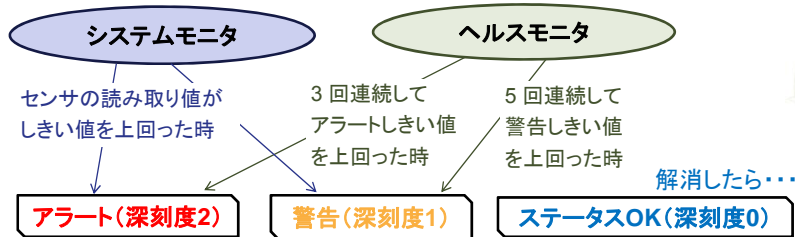
You'll make breakthroughs

④ Caché モニタリングツール：通知クラス

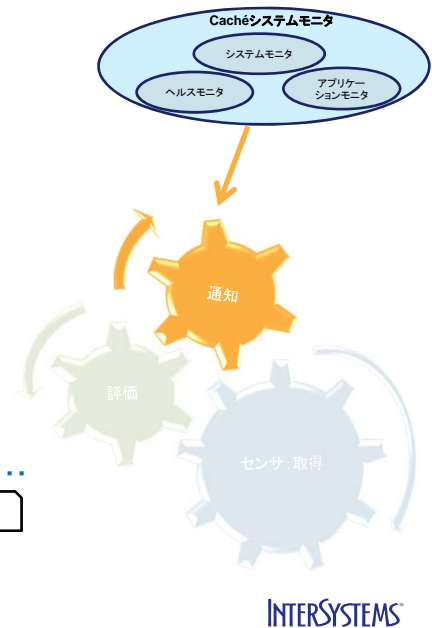
通知クラス * アラートの生成

- Cachéシステムモニタは、センサクラス及びサブスクライバクラスを呼び出した後に、通知クラスを呼び出す。
- 通知クラスは、サブスクライバクラスから通知を受け取ると、その通知をシステムモニタログ、コンソールログに書き込む。

※ C:\InterSystems¥Cache¥mgr¥SystemMonitor.log

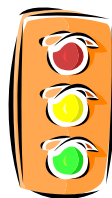


- システムモニタやヘルスマニタが出したエラーをメールで送信したい場合は、**^MONMGR**を使用する。



デモ 1

- 監視値の警告レベルを変えて、コンソールログへの出力メッセージを抑制する

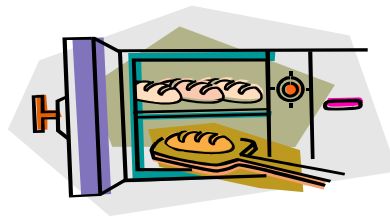


INTERSYSTEMS

28

デモ 2

- カスタムセンサクラスを作成する
- ベーカリーアプリケーション
- 現在のオーブンの温度をモニタリング



INTERSYSTEMS

30

④ Caché モニタリングツール : アプリケーションモニタ

Caché アプリケーションモニタ

- 主要なシステム定義メトリックとユーザ定義メトリックを収集してローカルネームスペースのグローバルに保存する。
- ユーザ定義のアラートで評価し、メールを送信もしくはクラスメソッドを実行する
- システムモニタ起動時、システムモニタで指定している実行開始ネームスペースで起動 (%SYSは常に起動)
- アプリケーションモニタで監視可能なメトリック群は、アプリケーションのセンサクラスに呼び出される
- ネームスペースごとに構成・実行可能(独自の評価式を構成可能)
- 通知オプション: なし、Email、クラスメソッドの実行

– 例: %Monitor.System.Licenseクラスの利用可能なライセンス数が10以下になったら Email 送信

Property: AvailableDist CurrentUsedDist

Expression: %1 - %2 < 10



件名: Cache Application Monitor Alert

内容: AlertLicense%Monitor.System.License: AvailableDist = 100 CurrentUsedDist = 98

INTERSYSTEMS

31

You'll make breakthroughs

アプリケーション・モニタ・システム・クラス

<code>%Monitor.System.Sample.AuditCount</code>	監査メトリック
<code>%Monitor.System.Sample.AuditEvents</code>	
<code>%Monitor.System.Sample.Clients</code>	クライアント・メトリック
<code>%Monitor.System.Sample.CSPGateway</code>	CSP ゲートウェイ・メトリック
<code>%Monitor.System.Sample.DiskSpace</code>	ディスク容量メトリック
<code>%Monitor.System.Sample.Freespace</code>	空き容量メトリック
<code>%Monitor.System.Sample.Globals</code>	グローバル・メトリック
<code>%Monitor.System.Sample.HistoryPerf</code>	履歴データベース・メトリック (このドキュメントの " Caché 履歴モニタ " の章を参照してください)
<code>%Monitor.System.Sample.HistorySys</code>	
<code>%Monitor.System.Sample.HistoryUser</code>	
<code>%Monitor.System.Sample.Journals</code>	ジャーナル・メトリック
<code>%Monitor.System.Sample.License</code>	ライセンス・メトリック
<code>%Monitor.System.Sample.LockTable</code>	ロック・テーブル・メトリック
<code>%Monitor.System.Sample.Routines</code>	ルーチン・メトリック
<code>%Monitor.System.Sample.Processes</code>	プロセス・メトリック
<code>%Monitor.System.Sample.Routines</code>	ルーチン・メトリック
<code>%Monitor.System.Sample.Servers</code>	サーバ・メトリック
<code>%Monitor.System.Sample.SystemMetrics</code>	システム活動メトリック

INTERSYSTEMS

32

You'll make breakthroughs

④ Caché モニタリングツール : アプリケーションモニタ

Caché アプリケーションモニタ : 履歴モニタの有効化

- %SYSネームスペースのみで実行可能である特別なアプリケーションモニタクラスがある
- アプリケーションモニタで提供されている以下のクラスは、有効化されると、システムの使用状況とパフォーマンスのメトリックの履歴データを作成および保持する
 - `%Monitor.System.HistorySys`
 - `%Monitor.System.HistoryPerf`
- 履歴データは、システムの使用状況とパフォーマンスの問題の経時的な分析に役立つ
 - **履歴モニタ**
- %SYS でのみ動作し、他のネームスペースに登録することはできない
- データベースに情報が格納できるので、DeepSeeを利用してカスタマイズが可能 (後ほど履歴モニタでデモ)
 - Cubeを含む環境はユーザが用意する必要がある

INTERSYSTEMS

33

④ Caché モニタリングツール : 履歴モニタ

Caché 履歴モニタ

- 一般的なパフォーマンスやシステムの使用率に関するデータをデータベースに保存し、パフォーマンスの問題(基準値からの逸脱)や長期的なシステム使用の分析に役立てる。
- 履歴モニタデータは、SYS.History パッケージで定義され、%SYS ネームスペースに保持される
- SQLまたは通常の永続オブジェクトアクセスで使用可能
- ベースラインと現在値を比較し、性能劣化などのシステムの異変を早急に検知
 - ベースラインとは、トラブルなく稼働しているシステムの状態や構造を示す管理データの標準値

INTERSYSTEMS

34

④ Caché モニタリングツール : 履歴モニタ

基本メトリック

• 4つの基本メトリック

- SYS.History.Performance (パフォーマンス)
 - グローバル参照やルーチン呼び出しのような一般的なパフォーマンスメトリックを追跡
 - 例: Global Refs, Phys Reads, などなど...
- SYS.History.WriteDaemon (ライトデーモン)
 - ライトデーモンサイクルのパフォーマンスメトリックを追跡
 - 例: WD Cycle Time, # blocks, などなど...
- SYS.History.SystemUsage (システム使用)
 - システムの使用率を追跡
 - 例: License Usage, Processes, などなど...
- SYS.History.Database (データベース)
 - 各ローカルデータベースについて、データベースの増大、ファイル・サイズ、および空き容量を追跡
 - 例: Database Size, Freespace, などなど...

INTERSYSTEMS

35

You'll make breakthroughs

④Caché モニタリングツール：履歴モニタ

データの収集

• 2グループにより編成

- SYS.History.PerfData
埋め込みオブジェクトとしてパフォーマンス及びライトデーモンクラスを含む
- SYS.History.SysData
システム使用およびデータベースクラスを含む

30秒単位

5分単位

• データ収集手順

1. ^%SYSMONMGR ユーティリティを使用し、以下の両方またはいずれかを有効化
 - %Monitor.System.HistoryPerf クラス
 - %Monitor.System.HistorySys クラス
2. %SYS ネームスペースでシステムモニタを再起動

^mgstatと類似した
データを収集

Num	MetricsClassName	Activated
1)	%Monitor.System.HistoryPerf	Y
2)	%Monitor.System.HistorySys	Y

INTERSYSTEMS

36

You'll make breakthroughs

④Caché モニタリングツール：履歴モニタ

集計

- 詳細データ(PerfData、SysData)の他に、毎時および日次集計も作成
 - SYS.History.Hourly および SYS.History.Daily として定義
- 統計処理:
 - 平均値、最大値、標準偏差、最小値、中央値、または合計を計算
- 自動でデータを削除 ※手動で削除も可能
 - 詳細データは7日間
 - 毎時データは 60 日間 (日次データは自動削除されない)

パフォーマンスデータ(Performance&WD) :

30 秒間隔でサンプリング

7日過ぎたら削除

システム使用データ(System&Database) :

5分間隔でサンプリング

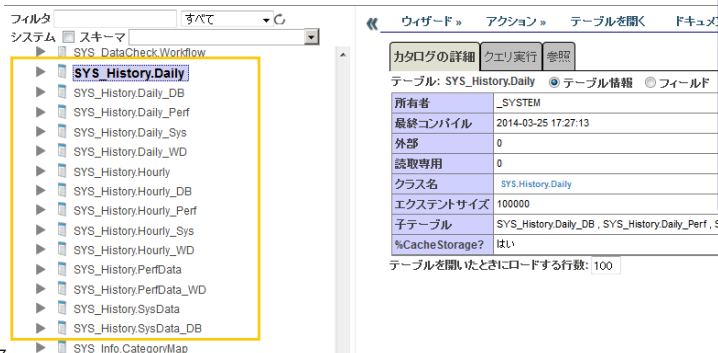
7日過ぎたら削除

時間データ:

60日過ぎたら削除

日次データ:

自動削除はされない。※手動で削除可能



INTERSYSTEMS

37

デモ 3

- 履歴モニタをDeepSeeのダッシュボードでしてみる



④ Caché モニタリングツール

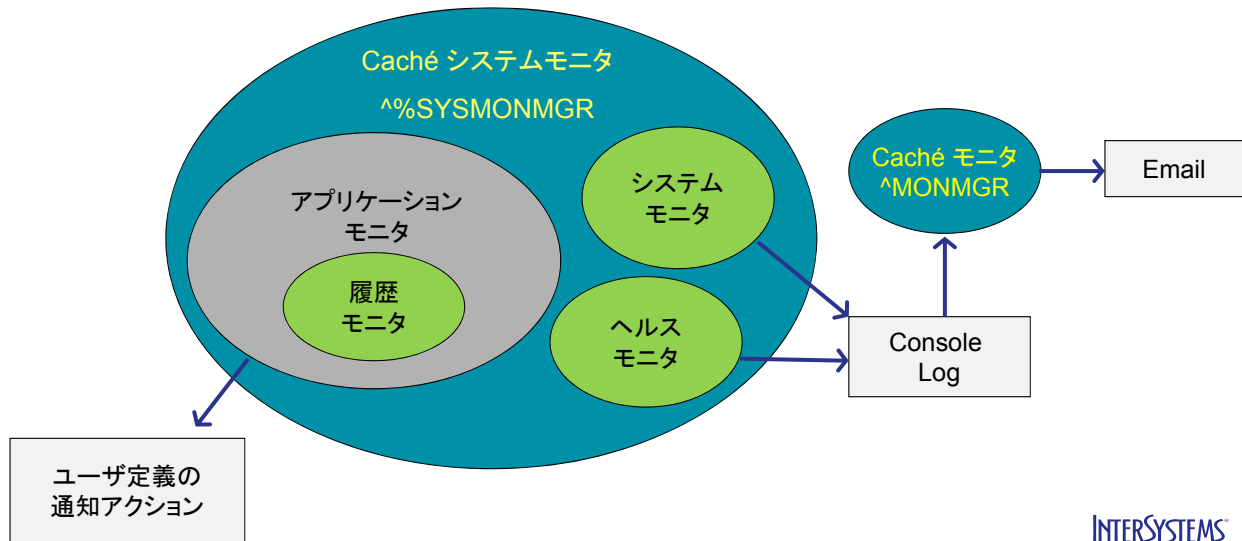
Cachéのモニタリングツール

Cachéモニタ ※	Cachéシステムモニタ		
	システムモニタ	ヘルスモニタ	アプリケーションモニタ
Do ^MONMGR	Do ^%SYSMONMGR	Do ^%SYSMONMGR ⇒ 6) Manage Health Monitor	Do ^%SYSMONMGR ⇒ 5) Manage Application Monitor
事前定義のシステムエラーを監視。 コンソール・ログを定期的にスキャンして構成された深刻度レベルのエントリを検出し、それぞれの通知を生成。	システム全体のステータスとリソースを監視。 システムおよびユーザ定義のメトリック（監視値）を評価し、固定のパラメータ及び閾値に基づき、通知を生成	システムおよびユーザ定義のメトリックをサンプリングし、ユーザ構成可能なパラメータおよび既定の通常値と比較して、通知を生成	主要なシステムメトリックをサンプリングし、評価後の収集値をローカルネームスペースのグローバルに格納。それを、ユーザ作成のアラート定義を使用して評価
・アラート・ログ ・電子メール生成・メソッドの実行（構成時のみ）	・コンソールログや、システムモニタログ ・電子メール生成・メソッドの実行（構成時のみ）	・コンソールログや、システムモニタログ ・電子メール生成・メソッドの実行（構成時のみ）	電子メール生成・メソッドの実行
デフォルトで開始（1インスタンスに1モニタ）	デフォルトで開始（登録ネームスペースのみ）※既定は%SYS	ユーザ自身が構成・開始（一部デフォルトで開始）	ユーザ自身が構成・開始
%SYSのみ	各ネームスペース固有の構成および設定で実行	%SYSのみ	各ネームスペース固有の構成および設定で実行
既に発生した状況の通知	障害を未然に防ぐ為、監視対象が閾値の範囲内であるかどうかを検知できる仕組みを提供	〃	〃

You'll make breakthroughs

④ Caché モニタリングツール

2013.1以降のモニタリング イメージ



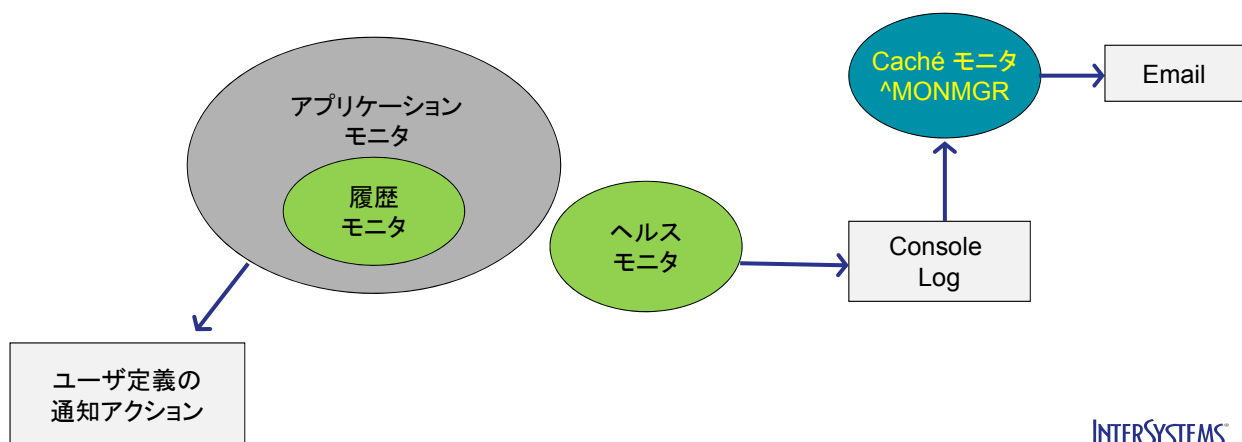
INTERSYSTEMS

40

You'll make breakthroughs

④ Caché モニタリングツール

(ご参考) 2012.2以前のモニタ



INTERSYSTEMS

41

④ Caché モニタリングツール

システムおよびメモリのステータスチェック

- 現在のシステム全体のステータスを取得可能

– `$SYSTEM.Monitor.State()`

ALERT (2)

• 30分間で5つ以上のアラート

WARNING (1)

• 1つのアラートか5つの警告

OK (0)

• アラートも警告もなし

INTERSYSTEMS

42

⑤ サードパーティーモニタリングツール

- BMC Patrol
- SNMP
- WMI
- WS-Management

INTERSYSTEMS

43

You'll make breakthroughs

⑤サードパーティーモニタリングツール

SNMP (Simple Network Management Protocol)

- SNMPとは？
 - ネットワークデバイスやコンピュータデバイスなど、TCP/IP ネットワーク全体を管理する手段として広く使用されている通信プロトコル
 - 多岐に渡る管理ツールに対して情報を管理、モニタする標準的な方法を提供
- CachéはAgentX (Agent Extensibility) プロトコルを使用してサブエージェントとして実装
 - Windowsのサービスを有効にしておく必要がある
- InterSystems は、Caché MIB (※)を提供
 - ※Caché SNMP インタフェースを介して取得できる全ての管理対象オブジェクトデータ
- SNMP を介した参照やクエリの対象とすることができる簡潔なアプリケーションメトリックを追加するためにMIBを拡張することが可能

INTERSYSTEMS

44

You'll make breakthroughs

⑤サードパーティーモニタリングツール

SNMP: サンプル

- 無償で提供されているSNMP ツール (今回使用したのは TWSNMP) のMIBブラウザを使用して、情報を収集すると、次のようになります。

The screenshot shows the MIB Browser application window. It has a top section with input fields for Node, IP Address, and SNMP Mode (set to SNMPv1). Below these are fields for Community/User (set to public) and Password (masked with asterisks). There are buttons for 'リトライ' (Retry), 'タイムアウト' (Timeout), '追加' (Add), and '削除' (Delete). A 'モード' (Mode) section has radio buttons for '通常' (Normal), 'テーブル' (Table), and 'Walk' (selected), and a checkbox for 'Bulkモード' (Bulk Mode). On the right, there are buttons for '開閉' (Open/Close), '閉じる(C)' (Close), '複写' (Copy), '保存' (Save), '順序確認' (Check Order), and 'ポーリング登録' (Register Polling). A status box shows: 取得数 = 855件, 時間 = 85.738秒, 速度 = 10.0件/秒. The main area is a table with two columns: 'オブジェクト名' (Object Name) and '値' (Value).

オブジェクト名	値
cacheSysName.8.83.89.77.80.79.83.73.65	SYMPOsia
cacheSysFile.8.83.89.77.80.79.83.73.65	C:\InterSystems\Symposia\cache.cpf
cacheSysDir.8.83.89.77.80.79.83.73.65	C:\InterSystems\Symposia\mgr
cacheSysVersion.8.83.89.77.80.79.83.73.65	Cache for Windows (x86-64) 2014.1 (Build 608U) Tue Mar 25 2014 17:23:33
cacheSysCurUser.8.83.89.77.80.79.83.73.65	22
cacheSysRtnCache.8.83.89.77.80.79.83.73.65	0
cacheSysCnCache.8.83.89.77.80.79.83.73.65	256
cacheSysErrorMsg.8.83.89.77.80.79.83.73.65	
cacheSysLicenseUsed.8.83.89.77.80.79.83.73.65	2
cacheSysLicenseHigh.8.83.89.77.80.79.83.73.65	2
cacheSysState.8.83.89.77.80.79.83.73.65	OK
cachePerfLines.8.83.89.77.80.79.83.73.65	2694786
cachePerfRtnLoad.8.83.89.77.80.79.83.73.65	47170
cachePerfGloRef.8.83.89.77.80.79.83.73.65	391019
cachePerfGloSets.8.83.89.77.80.79.83.73.65	98969
cachePerfGloKills.8.83.89.77.80.79.83.73.65	0
cachePerfLogReads.8.83.89.77.80.79.83.73.65	285960
cachePerfRtnReads.8.83.89.77.80.79.83.73.65	3272
cachePerfRtnWrites.8.83.89.77.80.79.83.73.65	157
cachePerfRtnEntry.8.83.89.77.80.79.83.73.65	114
cachePerfLockCom.8.83.89.77.80.79.83.73.65	540
cachePerfLockSucc.8.83.89.77.80.79.83.73.65	394
cachePerfLockFail.8.83.89.77.80.79.83.73.65	5
cachePerfGThrottleCur.8.83.89.77.80.79.83.73.65	0
cachePerfGThrottleMax.8.83.89.77.80.79.83.73.65	0
cachePerfGThrottleCnt.8.83.89.77.80.79.83.73.65	0
cachePerfGUpLock.8.83.89.77.80.79.83.73.65	0
cachePerfGUpCnt.8.83.89.77.80.79.83.73.65	0
cachePerfGUpMin.8.83.89.77.80.79.83.73.65	0

INTERSYSTEMS

45

まとめ

- Cachéパフォーマンスツールは、システム全体や個別の問題の原因を特定するのに効果的
- Cachéモニタリングツールは、監視環境を適正に調整・設定することにより、問題を未然に防いだり、起こったとしても早期発見するのに効果的
- 履歴モニタにより、
 - パフォーマンス劣化の問題が起きた時、通常時のデータと比較し、違いを容易に探ることができる
 - 将来におけるシステム資産の見積もりができる
- Cachéモニタリングツールをうまく利用し、最適な環境で運用してきましょう
 - 本番環境でも通常時の情報を収集しておくことが、問題発生時の解決の近道となる

INTERSYSTEMS[®]

46

ご質問

INTERSYSTEMS[®]

47

You'll make breakthroughs

Appendix : SNMPのトラブルシューティング

Caché側:

1. %Service_Monitor が有効になっていることを確認
2. \$\$start^SNMP() で実行されるプロセスが起動されているかを確認
3. ^SYS("MONITOR","SNMP","DEBUG")=1をセットし、SNMP.logを確認する

Windows側:

1. Windows のサービス:SNMP Service が開始しているか確認
 - Cachéをインストールする前に、SNMPサービスがインストールされることが前提。
 - Caché が SNMP サービスよりも先にインストールされた場合
 - ① Windows SNMP サービスのインストール後、\$\$Register^SNMP() 関数を使用して登録し、SNMP サービスを再起動
または
 - ② Cachéの再インストール

INTERSYSTEMS®

48



You'll make breakthroughs



インターシステムズジャパン株式会社
カスタマーサポート部
掛地 めぐみ

