Migration to VSI OpenVMS
HPE PERFDAT OpenVMS Performance Solution
HPE OpenVMS ServiceControl
OpenVMS Connect Forum 2017, Vienna
DI. Dr. Wolfgang Burger
HPE GDS/TS Austria
Agenda

• Migration to VSI OpenVMS
  • HPE PERFDAT OpenVMS Performance Solution
  • HPE OpenVMS ServiceControl

• HPE PERFDAT OpenVMS Performance Solution
  • Concepts & architecture
  • How-To
    ▪ Configure selective file filtering (Demo)
    ▪ Create user defined (calculated) stats (Demo)
    ▪ Online alerting (Demo)
    ▪ Report automation (automatic graph creation)
Migration to VSI OpenVMS

What is HPE PERFDAT?

- OpenVMS Performance Solution
  - High resolution performance collector
  - Automatic data management
  - Highly automated collection, filtering, charting and trend analysis capabilities for effective performance lifecycle management
- Capable to monitor non-OpenVMS systems
  - Brocade switches
  - EVA storage arrays
  - Linux
  - ... Any system which provides performance data via SNMP
- No 3rd party and/or freeware/open source dependency
- Commercial product
HPE PERFDAT customer

Customers in 16 countries

- ~1500 OpenVMS productive system
- Monitors ~ 500 non-OpenVMS systems
Migration to VSI OpenVMS

What is HPE OpenVMS ServiceControl?

Failover

Switchover

- Freeware
- No 3rd party and/or freeware/open source dependency
- Support on request
HPE OpenVMS ServiceControl at a glance

Visualize Services

One-Stop Service Health Check

Service State and Event Monitoring

Service Management

Service Auto-Recovery

Service Exception Alerting
Migration to VSI OpenVMS

HPE PERFDAT & HPE OpenVMS ServiceControl characteristics

- **HPE PERFDAT**
  - C code
  - OpenVMS data collector
    - 70% kernel mode code
    - Calls OpenVMS kernel routines
    - C-RTL, LIB$, System services
  - All other components
    - 100% user mode code
    - C-RTL, LIB$, System services
  - Inter-process communication
    - MBX
    - TCP/IP
    - UDP (SNMP)

- **HPE OpenVMS ServiceControl**
  - C code
  - 100% user mode code
  - C-RTL, LIB$, System services
  - Inter-process communication
    - ICC
    - MBX
    - TCP/IP
Migration to VSI OpenVMS

• Challenges
  • Where do I get the kits?
  • Where do I get licences?
  • Applications binary compatible with VSI OpenVMS?
  • Re-compile & re-link required?
  • Code modifications required
    • Kernel mode routines
      • OpenVMS internal data structure modification that affect HPE PERFDAT?
    • Mixed cluster architecture
      • ICC has always been a challenge
Migration to VSI OpenVMS

Test environment

ES47 2P/1C
EV7 1GHz
2 GB Memory
HPE OpenVMS AXP
V7.3-2 – V8.4

rx2800 2P/16C
Itanium 9500 2.13 GHz
72 GB Memory
VSI OpenVMS IA64
V8.4 1H1 – V8.4 2H1

rx4640 1P/1C
Madison 1.3 GHz
2GB Memory
HPE OpenVMS IA64
V8.2 – V8.4
Migration to VSI OpenVMS

• Challenges
  • Where do I get the kits?
  • Where do I get licences?

• Could not find
  • link or hint on the official VSI web site
  • link or hint on the official HPE web site

• How to get them?
  • Contact VSI support Europe
    L1-support@vmssoftware.com
  • Response within 24 h with download link
  • All VSI kits, patches & layered products
Migration to VSI OpenVMS

• Challenges
  • Applications binary compatible with VSI OpenVMS?
    YES
  • Re-compile & re-link required?
    YES
  • Code modifications required
    NO
  • Any mixed cluster issues (HPE OpenVMS ServiceControl)
    NO
  • Only actions required
    • Modify Install/Startup/Shutdown/Control scripts due to:
      • Product re-branding
      • New version number
Questions
HPE PERFDAT OpenVMS Performance Solution
Requirements

• High resolution performance data collection for easy root-cause analysis

• Completeness of data

  The data collector has to provide sufficient performance information about all OpenVMS sub-systems including XFC, LAN and network protocol support.

• Online rule based performance alerting

  Online performance alerting has to support system management to detect performance anomalies even though their impact does not slow down the overall system performance significantly so that this remains transparent to the end-user.
Requirements (contd.)

• Easy to handle
  o Plug and play
    • Once the performance solution is installed data has to be collected and all performance management related tasks like trending and data archiving has to be performed automatically to maintain a performance history based on predefined profiles, unattended, and without any need of additional customization work.
  o Easy to manage and control
  o Automated data management without any system management intervention
    • Ability to manage huge amounts of data (> 1TByte)
    • Archive and housekeeping functionality
  o Easy data transfer for offline analysis
Requirements (contd.)

- Single point and transparent performance data access regardless of where the performance data is stored within the whole environment via a single common interface
- Data analysis without data pre-processing
- Data import/export capabilities to guarantee:
  - collaboration with other performance data collection utilities (import data from other sources)
  - collaboration with existing performance analysis utilities and charting tools
Requirements (contd.)

- Automatic trend and capacity reporting
- Up- and backward data compatibility
- Full cluster analysis capability
- No dependency on any layered product except those available on the OpenVMS installation media
- No dependency on any 3rd party product or any kind of shareware/freeware
Requirements (contd.)

- State of the art graphical GUI for data analysis
  - Easy to handle
  - Intuitive
  - Easy data navigation
  - Online descriptions for all statistics available
  - State of the art graphical features like
    - Stack/unstack functionality
    - Zoom in/out
    - Shift left /right
    - Data scanning
    - Ability to scale graphs separately
    - Auto, native and manual scaling capability
    - Data overlay capability (graphs of different time periods can be overlapped to allow visual comparison)
  - Correlation- and deviation analysis capability
  - Multi window support for multi screen systems
Requirements (contd.)

• We are not alone
  o Performance depend on external, shared storage
  o Systems are coupled via shared storage

• Serious performance management sometimes requires one „to look over the rim of the tea cup“
  o Attached shared Storage
  o Systems accessing the same shared storage
    ▪ Solaris
    ▪ Linux
    ▪ ...
Performance Architecture

**PERFDAT**

- Performance Architecture
- Distributed Performance Database
- Query Interface
- Performance Database File Name Cache service DQL_NAME
- Statistics
- Auto trend engine
- Graphical User Interface
- Auto Archiving
- Performance Alerting

**API**

- Insert
- Imports
- Insert

**PERFDAT**

- OpenVMS Data Collector
- EVA/SNMP extension

**Maps**

- CSV

**Cluster view database**

**Configuration database**
HPE PERFDAT Components

- OpenVMS Data Collector
- PERFDAT SNMP extension
- PERFDAT EVA extension
- Distributed performance database
- Application Programming Interface
- PERFDAT configuration database
- Performance database file name cache service DQL_NAME
- Data Query Interface (DQL)
- Online performance alerting
- Statistics package
- Auto trend engine
- Auto Archiving and housekeeping
- Management Interface (PERFDAT_MGR)
- Graphical User Interface
- Tools
HPE PERFDAT Components

- OpenVMS Data Collector
- PERFDAT SNMP extension
- PERFDAT EVA extension
- Distributed performance database
- Application Programming Interface
- PERFDAT configuration database
- Performance database file name cache service DQL_NAME
- Data Query Interface (DQL)
- Online performance alerting
- Statistics package
- Auto trend engine
- Auto Archiving and housekeeping
- Management Interface (PERFDAT_MGR)
- Graphical User Interface
- Tools
OpenVMS Data Collector – Features

- Up to 3 collections in parallel
- More than 700 statistics organized in 25 metrics
- Profile controlled – profiles reside in the PERFDAT configuration database and are managed via the PERFDAT_MGR utility
- Sample interval is freely definable (minimum = 1 second)
- Each of the metrics can be enabled/disabled independently
- For each of the metrics (except the system metrics), thresholds can be set to minimize the amount of data collected
OpenVMS Data Collector – Features

- Metrics can be restricted to single/multiple devices, processes, users, images and volumes
- Device metrics allows I/O resolution to single process, files and files per process (not only hot file statistic but also the originator of hot files can be identified)
- Files in the device- and XFC metrics not only resolve to file ID’s but also to their real file names
- Complete XFC integration
- Permits online monitoring
- Online performance alerting can be enabled dynamically
OpenVMS Data Collector – Features

• Dynamic resource trimming
  o In order to avoid performance problems due to running PERFDAT, the tool monitors its own resource consumption, and if CPU load and/or I/O load exceeds definable thresholds PERFDAT automatically increases collection sample intervals and/or dismisses metrics rules.

• Controlled by PERFDAT_MGR
OpenVMS performance metrics

- System
- CPU
- Process
- User
- Image
- Account
- Device
  - Device.IOSize
  - Device.IOTimeHist
  - Device.File
  - Device.Process
  - Device.Process.File
  - Device.Capacity
- Device.Path
- IOPathes
- XFCVolume
  - XFCVolume.IOSize
  - XFCVolume.File
  - XFCVolume.File.IOSize
  - LANAdapter
    - LANAdapter.Device
  - LANProtocol
  - SCSPort
  - SCSPort.VC
  - SCSPort.VC.Channel
HPE PERFDAT Components

- OpenVMS Data Collector
- PERFDAT SNMP extension
- PERFDAT EVA extension
- Distributed performance database
- Application Programming Interface
- PERFDAT configuration database
- Performance database file name cache service DQL_NAME
- Data Query Interface (DQL)
- Online performance alerting
- Statistics package
- Auto trend engine
- Auto Archiving and housekeeping
- Management Interface (PERFDAT_MGR)
- Graphical User Interface
- Tools
PERFDAT Query Interface (DQL)

- worldwide single point access
PERFDAT Query Interface (DQL)

- **Features**
  - Query interface (DQL) similar to SQL
  - Transparent single point access via network abstraction layer
  - Up- and downward data compatibility via data abstraction layer
  - Dynamic CSV file mapping capability for accessing and analysing data from different data sources
  - Multi file version support
  - CSV load capability
  - CSV file import capability (data is not only inserted but also normalized)
  - CSV export capability
  - Statistic package fully integrated in data query interface
Components

DCL Query Interface
Data Management

DQL$

PDBC$SRV

Statistic Package

Node A

Data Collector

1 2 3

CSV

DCL Query Interface
Data Management

Performance Database Connectivity Server

CSV Mapping CFG

Data Server

CSV Mapping CFG

RMS data streams

IP data streams

Management streams

OpenVMS Connect Forum 2017

HPE PERFDAT – OpenVMS Performance Solution
Query Interface - Community

- When accessing the performance database via a dedicated server the Community defines the database view

- Community
  - Defined via the logical PERFDAT$COMMUNITY
  - Defines the nodes of interest
  - Only data created by these nodes will be visible

- Independent of the Community definition, the local node and the archive node (if available) are always accessed
Query Interface - Data Flow

Data request from GUI

A
DQL$/PDBC$SRV
Community = A,B
DQL$/SRV

B
DQL$/PDBC$SRV
Community = A,C
DQL$/SRV

C
DQL$/PDBC$SRV
Community = C
DQL$/SRV

Archive
DQL$/PDBC$SRV
DQL$/SRV

Community = A,B

Community = A,C

Community = C
Query Interface - Data Flow

Data request from GUI

A
DQL$ / PDBC$SRV
Community = A,B
DQL$SRV

B
DQL$ / PDBC$SRV
Community = A,C
DQL$SRV

C
DQL$ / PDBC$SRV
Community = C
DQL$SRV

Archive
DQL$ / PDBC$SRV
DQL$SRV
Query Interface - Data Flow

A
DQL$/PDBC$SRV
Community = A,B
DQL$SRV

B
DQL$/PDBC$SRV
Community = A,C
DQL$SRV

C
DQL$/PDBC$SRV
Community = C
DQL$SRV

Archive
DQL$/PDBC$SRV

Data request from GUI
HPE PERFDAT Components

- OpenVMS Data Collector
- PERFDAT SNMP extension
- PERFDAT EVA extension
- Distributed performance database
- Application Programming Interface
- PERFDAT configuration database
- Performance database file name cache service DQL_NAME
- Data Query Interface (DQL)
- Online performance alerting
- Statistics package
- Auto trend engine
- Auto Archiving and housekeeping
- Management Interface (PERFDAT_MGR)
- Graphical User Interface
- Tools
Online Performance Alerting

• Provides real-time alerting capabilities
• Can be dynamically enabled for each active performance collection (OpenVMS & SNMP extension & EVA extension)
• Statistics to monitor, alert conditions and alert method defined by alert blocks
• Alert blocks are defined within an alert definition file
• An alert definition file is a text file – syntax comparable to PCM import files
• An alert definition file with valid alert blocks are a prerequisite to enable online performance alerting
• Max. number of elements tracked by a single alert block is 4096
HPE PERFDAT Components

- OpenVMS Data Collector
- PERFDAT SNMP extension
- PERFDAT EVA extension
- Distributed performance database
- Application Programming Interface
- PERFDAT configuration database
- Performance database file name cache service DQL_NAME
- Data Query Interface (DQL)
- Online performance alerting
- Statistics package
- Auto trend engine
- Auto Archiving and housekeeping
- Management Interface (PERFDAT_MGR)
- Graphical User Interface
- Tools
Auto trend engine

- Is triggered by the archiving process (if the archiving process is stopped the auto trend engine is stopped too)
- Only processes performance data created on the local node
- Time span of a trend report can be day, week, month, quarter or year.
- Trends are generated based on predefined report profiles
- Trend report profiles are defined via PERFDAT_MGR
### DEVICE statistics

<table>
<thead>
<tr>
<th>Statistics</th>
<th>Description</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>CQDs</td>
<td>Total QEO rate on device</td>
<td>[IO/s]</td>
</tr>
<tr>
<td>Rps</td>
<td>Total device IO request rate</td>
<td>[IO/s]</td>
</tr>
<tr>
<td>10s</td>
<td>Total service IO (passing START_IO routine) rate</td>
<td>[IO/s]</td>
</tr>
<tr>
<td>10Sp</td>
<td>Total split service IO (passing START_IO routine) rate</td>
<td>[IO/s]</td>
</tr>
<tr>
<td>Abs</td>
<td>Total Aborted</td>
<td>[IO/s]</td>
</tr>
<tr>
<td>1Ms</td>
<td>Total Throughput</td>
<td>[MB/s]</td>
</tr>
<tr>
<td>1Rqtime</td>
<td>IO Request time</td>
<td>[ms]</td>
</tr>
<tr>
<td>1RqtimeMax</td>
<td>MAX IO Request time during last sample interval</td>
<td>[ms]</td>
</tr>
<tr>
<td>1RqrespAcc</td>
<td>Accuracy of IO Request time</td>
<td>[%]</td>
</tr>
<tr>
<td>1S10s</td>
<td>IO Service time</td>
<td>[ms]</td>
</tr>
<tr>
<td>1S10sMax</td>
<td>MAX IO Service time during last sample interval</td>
<td>[ms]</td>
</tr>
<tr>
<td>1S10sAcc</td>
<td>Accuracy of IO Service time</td>
<td>[%]</td>
</tr>
<tr>
<td>RdQIOs</td>
<td>Read QIO rate on device</td>
<td>[IO/s]</td>
</tr>
<tr>
<td>RdRIOs</td>
<td>Read device IO request rate</td>
<td>[IO/s]</td>
</tr>
<tr>
<td>Rd10s</td>
<td>Read service IO (passing START_IO routine) rate</td>
<td>[IO/s]</td>
</tr>
<tr>
<td>Rd10sp</td>
<td>Read split service IO (passing START_IO routine) rate</td>
<td>[IO/s]</td>
</tr>
<tr>
<td>RdAbs</td>
<td>Read Aborted</td>
<td>[IO/s]</td>
</tr>
<tr>
<td>RdRbs</td>
<td>Read Throughput</td>
<td>[MB/s]</td>
</tr>
<tr>
<td>RdRqtime</td>
<td>Read IO Request time</td>
<td>[ms]</td>
</tr>
<tr>
<td>RdRqtimeMax</td>
<td>Read MAX IO Request time during last sample interval</td>
<td>[ms]</td>
</tr>
<tr>
<td>RdRqrespAcc</td>
<td>Accuracy of Read IO Request time</td>
<td>[%]</td>
</tr>
<tr>
<td>RdS10s</td>
<td>Read IO Service time</td>
<td>[ms]</td>
</tr>
<tr>
<td>RdS10sMax</td>
<td>MAX IO Service time during last sample interval</td>
<td>[ms]</td>
</tr>
<tr>
<td>RdS10sAcc</td>
<td>Accuracy of Read IO Service time</td>
<td>[%]</td>
</tr>
<tr>
<td>WrQIOs</td>
<td>Write QIO rate on device</td>
<td>[IO/s]</td>
</tr>
<tr>
<td>WrRIOs</td>
<td>Write device IO request rate</td>
<td>[IO/s]</td>
</tr>
<tr>
<td>Wr10s</td>
<td>Write service IO (passing START_IO routine) rate</td>
<td>[IO/s]</td>
</tr>
<tr>
<td>Wr10sp</td>
<td>Write split service IO (passing START_IO routine) rate</td>
<td>[IO/s]</td>
</tr>
<tr>
<td>WrAbs</td>
<td>Write Aborted</td>
<td>[IO/s]</td>
</tr>
<tr>
<td>WrRbs</td>
<td>Write Throughput</td>
<td>[MB/s]</td>
</tr>
<tr>
<td>WrRqtime</td>
<td>Write IO Request time</td>
<td>[ms]</td>
</tr>
<tr>
<td>WrRqtimeMax</td>
<td>Write MAX IO Request time during last sample interval</td>
<td>[ms]</td>
</tr>
<tr>
<td>WrRqrespAcc</td>
<td>Accuracy of Write IO Request time</td>
<td>[%]</td>
</tr>
<tr>
<td>WrS10s</td>
<td>Write IO Service time</td>
<td>[ms]</td>
</tr>
<tr>
<td>WrS10sMax</td>
<td>MAX IO Service time during last sample interval</td>
<td>[ms]</td>
</tr>
<tr>
<td>WrS10sAcc</td>
<td>Accuracy of Write IO Service time</td>
<td>[%]</td>
</tr>
<tr>
<td>CIQDs</td>
<td>Ctrl QIO rate on device</td>
<td>[IO/s]</td>
</tr>
<tr>
<td>CIRpus</td>
<td>Ctrl device IO request rate</td>
<td>[IO/s]</td>
</tr>
<tr>
<td>CIRAbs</td>
<td>Ctrl Aborted</td>
<td>[IO/s]</td>
</tr>
<tr>
<td>CIRqtime</td>
<td>Ctrl IO Request time</td>
<td>[ms]</td>
</tr>
<tr>
<td>CIIRqtimeMax</td>
<td>Ctrl MAX IO Request time during last sample interval</td>
<td>[ms]</td>
</tr>
<tr>
<td>CIIRqrespAcc</td>
<td>Accuracy of Ctrl IO Request time</td>
<td>[%]</td>
</tr>
<tr>
<td>Qlen</td>
<td>Device IO queue length</td>
<td>[?]</td>
</tr>
<tr>
<td>I0thresh</td>
<td>IO request threshold</td>
<td>[IO/s]</td>
</tr>
<tr>
<td>ElementCnt</td>
<td></td>
<td>[?]</td>
</tr>
</tbody>
</table>
Response-Time sensitive Application (serial I/Os)

- iRQTime
  - Avg. end-to-end response time (read&write) – process point of view

- iRQTimeMax
  - Max end-to-end response time (read&write) – process point of view
  - High value -> check DEVICE.IOTIMEHIST metric

- iIOTime
  - Avg. physical response time (read&write) – adapter point of view

- iIOTimeMax
  - Max physical response time (read&write) – adapter point of view

- iQlen
  - Device I/O queue length
Response-Time sensitive Application (serial I/Os)

- **iRdRQTime**
  - Avg. end-to-end read response time - process point of view

- **iRdRQTimeMax**
  - Max end-to-end read response time – process point of view
  - High value -> check DEVICE.IOTIMEHIST metric

- **iRdIOTime**
  - Avg. physical read response time – adapter point of view

- **iRdIOTimeMax**
  - Max physical read response time – adapter point of view
Response-Time sensitive Application (serial I/Os)

- iWrRQTime
  - Avg. end-to-end write response time - process point of view

- iWrRQTimeMax
  - Max end-to-end write response time – process point of view
  - High value -> check DEVICE.IOTIMEHIST metric

- iWrIOTime
  - Avg. physical write response time – adapter point of view

- iWrIOTimeMax
  - Max physical write response time – adapter point of view
Response-Time sensitive Application (serial I/Os)

- **iCtrlRQTime**
  - Avg. end-to-end non data transfer response time – process point of view

- **iCtrlRQTimeMax**
  - Max end-to-end non data transfer response time - process point of view
  - High value -> check DEVICE.IOTIMEHIST metric

- No physical I/O response time stats for non data transfers. Non data transfers are effectively reads or writes at this level.
DEVICE.IOTIMEHIST

- Big difference between avg. and max response time values
- Check this metric if this is due to a single event or not

![DEVICE.IOTIMEHIST Table]

<table>
<thead>
<tr>
<th>Statistics</th>
<th>Description</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>$2msecPerc</td>
<td>Percentage of &lt; 2ms I/Os</td>
<td>%</td>
</tr>
<tr>
<td>IOs</td>
<td>Number of I/Os within last sample interval</td>
<td>[IO]</td>
</tr>
<tr>
<td>IOTime</td>
<td>Avg I/O Service time during last sample interval</td>
<td>[ms]</td>
</tr>
<tr>
<td>IOTimeMax</td>
<td>Max I/O Service time during last sample interval</td>
<td>[ms]</td>
</tr>
<tr>
<td>2ms</td>
<td>Number of I/Os completed within 2ms</td>
<td>[IO]</td>
</tr>
<tr>
<td>4ms</td>
<td>Number of I/Os completed between 2ms and 4ms</td>
<td>[IO]</td>
</tr>
<tr>
<td>6ms</td>
<td>Number of I/Os completed between 4ms and 6ms</td>
<td>[IO]</td>
</tr>
<tr>
<td>10ms</td>
<td>Number of I/Os completed between 6ms and 10ms</td>
<td>[IO]</td>
</tr>
<tr>
<td>20ms</td>
<td>Number of I/Os completed between 10ms and 20ms</td>
<td>[IO]</td>
</tr>
<tr>
<td>30ms</td>
<td>Number of I/Os completed between 20ms and 30ms</td>
<td>[IO]</td>
</tr>
<tr>
<td>40ms</td>
<td>Number of I/Os completed between 30ms and 40ms</td>
<td>[IO]</td>
</tr>
<tr>
<td>50ms</td>
<td>Number of I/Os completed between 40ms and 50ms</td>
<td>[IO]</td>
</tr>
<tr>
<td>100ms</td>
<td>Number of I/Os completed between 50ms and 100ms</td>
<td>[IO]</td>
</tr>
<tr>
<td>200ms</td>
<td>Number of I/Os completed between 100ms and 200ms</td>
<td>[IO]</td>
</tr>
<tr>
<td>300ms</td>
<td>Number of I/Os completed between 200ms and 300ms</td>
<td>[IO]</td>
</tr>
<tr>
<td>400ms</td>
<td>Number of I/Os completed between 300ms and 400ms</td>
<td>[IO]</td>
</tr>
<tr>
<td>500ms</td>
<td>Number of I/Os completed between 400ms and 500ms</td>
<td>[IO]</td>
</tr>
<tr>
<td>1000ms</td>
<td>Number of I/Os completed between 500ms and 1000ms</td>
<td>[IO]</td>
</tr>
<tr>
<td>GT1sec</td>
<td>Number of I/Os not completed within 1000ms</td>
<td>[IO]</td>
</tr>
<tr>
<td>SampleTime</td>
<td>Sample Time</td>
<td>[sec]</td>
</tr>
</tbody>
</table>
How-To: Configure selective file filtering

- Any HPE PERFDAT data collection is profile controlled
- Use the PERFDAT_MGR utility to add/modify/delete collection profiles
  - ADD PROFILE <name>/OS_TYPE=<OS-type|Application-Name>
  - MODIFY PROFILE <name>/OS_TYPE=<OS-type|Application-Name>
  - DELETE PROFILE <name>/OS_TYPE=<OS-type|Application-Name>
- Create a text file which contains the files to monitor
  - One file name per line
  - Wildcards supported for the file names as with the DIR command
How-To: Configure selective file filtering

File filter input file example:

$ type SHARKDB$ROOT:[CFG]OGS$DB_FILES.TXT
DSA400:[OGS.HP.TICKET$DB]HP$TICKET$DB.DBR
DSA400:[OGS.HP.TICKET$DB.EUML]EUML.DBS
DSA400:[OGS.HP.TICKET$DB.LOTTO]LOTTO.DBS
DSA400:[OGS.HP.TICKET$DB.QUITTUNG]QUITTUNGS_NBR.DBS
DSA400:[OGS.HP.TICKET$DB]HP$TICKET$DB.AIJ*
DSA410:[OGS.HP.UMSATZ$DB]HP$UMSATZ$DB.DBR
DSA410:[OGS.HP.TICKET$DB.TOTO]TOTO.DBS
DSA410:[OGS.HP.UMSATZ$DB]HP$UMSATZ$DB.AIJ*
DSA420:[OGS.HP.TICKET$DB.JOKER]JOKER.DBS
DSA420:[OGS.HP.RUNDEN$DB]HP$RUNDEN$DB.AIJ*
DSA430:[OGS.HP.UMSATZ$DB.TAG]UMSATZ_TAG.DBS
SHARKDB$RUJROOT:[RUJ]*.RUJ
$
How-To: Configure selective file filtering

Create/modify profile with selective file filtering

PerfDat_MGR> ADD PROFILE OGS/OS=OPENVMS

  WELCOME to OpenVMS collection profile wizard

Collection sample interval [600 sec]: 60
Enable SYSTEM metrix [Yes]:
...
Enable DEVICE metrix: [Yes]:
  On/devices (eq. DKA100, DG*, TN*) [*D*,*DS*]: Yes
  Enable io size metrix on selected FOD devices [No]: Yes
Enable FILE metrix on selected FOD devices [No]: Yes
  Top $QIO rate FILE statistics on selected FOD devices [No]:
...
  Filter List File name []: SHARKDB$ROOT:[CFG]OGS$DB_FILES.TXT
Do you want to enable per PROCESS collection on selected devices [No]: Yes
  Top $QIO rate PROCESS statistics on selected devices [No]:
    On Process [ALL]: HP*,SHARK*
Enable per FILE collection for each Process collection on FOD devices [No]: Yes
  Top $QIO rate FILE statistics for each Process on FOD devices [No]:
...
  Filter List File name []: SHARKDB$ROOT:[CFG]OGS$DB_FILES.TXT
...
Process to be excluded from file monitoring []: NONE
How-To: Configure selective file filtering

• Start performance collection with new/modified profile

  $ MCR PERFDAT_MGR START COLLECTION <profile>
  /OS_TYPE=OpenVMS [/SHARE]

• Define the new/modified profile as the default profile in the auto-start table, if the default collection shall start with this profile whenever the HPE PERFDAT OpenVMS data collector starts

  $ MCR PERFDAT_MGR MODIFY AUTOSTART <node-name>
How-To: Create user defined stats

• User defined stats are:
  o Calculated statistics (measures)
  o Can be defined node specific
  o Can be accessed as if they are part of the data collection

• Usefull whenever you need a different view on the performance data
  o i.e. 2ms of DEVICE.IOTIMEHIST
    ▪ provides I/Os per second which completed < 2ms
    ▪ You need to know %
    ▪ new user stat $2msecPerc = 2msec / iIOS * 100
How-To: Create user defined stat - GUI

1. Select toolbar button

2. Select OS Filter/Metric Filter/Node Filter
How-To: Create user defined stat - GUI

1. Stats name
2. Description
3. Description
4. Unit
5. Click
How-To: Create user defined stat - GUI

User stat available in stats list
How-To: Create user defined stat – DQL$

Syntax:

```
DEFINE PROCEDURE equation METRIX metric_name
    OSTYPE OS_name DESCRIPTION description_text
    UNIT unit_text [NODE node_name];
```

This example:

```
$ MCR DQL$
DQL$> DEFINE PROCEDURE $2msecPerc = 2ms / iIOs * 100
    METRIX DEVICE.IOTIMEHIST OS OPENVMS
    DESCRIPTION "Percentage of < 2ms I/Os" UNIT "%" NODE *;
```

For detailed description of the syntax please refer to the documentation or use „GUI learning mode“
How To: Configure Online performance alerting

1) Baseline your system(s)
2) Define Alert definition files
   • Define the alert blocks
   • Define the alter methods
3) Enable online alerting on HPE PERFDAT collection
Alert definition file

Default alert definition files provided (use them as examples)

- OpenVMS:
  PERFDAT$CFG:PERFDAT_ALERT_OPENVMS.CFG
- EVA:
  PERFDAT$CFG:PERFDAT_ALERT_EVA.CFG
- Brocade:
  PERFDAT$CFG:PERFDAT_ALERT_BROCADE.CFG
- TRU64:
  PERFDAT$CFG:PERFDAT_ALERT_TRU64.CFG
Alert block

• Alert block starts with
  ADD ALERT:

• Alert block ends with
  END ALERT:

• Configuration keywords:
  • OSTYPE:
  • METRIEX:
  • STATISTICS:
  • SCALED_BY:
  • OPERATOR
  • ELEMENTS
  • STACKED
  • STACKED_ELEM_NAME:
  • SAMPLE_COUNT:
  • WARNING_THRESHOLD:
  • WARNING_CMD:
  • CRITICAL_THRESHOLD:
  • CRITICAL_CMD:
  • CLEARALERTMSG:
  • CLEARALERT_CMD:
  • SUBMIT_QUEUE:
Alert definition block (example)

ADD ALERT:

OSTYPE: OpenVMS
METRIX: DEVICE
STATISTICS: iRdIOs, iRdIOTime, iWrIOTime
SCALED_BY:
OPERATOR: GT, GT, GT
ELEMENTS: DSA*
STACKED: NO
STACKED_ELEM_NAME:
SAMPLE_COUNT: 2
WARNING_THRESHOLD: 200, 1, 1
WARNING_CMD:
CRITICAL_THRESHOLD: 400, 2, 2
CRITICAL_CMD:
CLEARALERTMSG: TRUE
CLEARALERT_CMD:
SUBMIT_QUEUE:

END ALERT:
Configuration parameters

Please refer to:

• PERFDAT_MGR online help
  $ MCR PERFDAT_MGR HELP ENABLE ALERT

• HPE PERFDAT – PERFDAT_MGR reference manual
  Section ENABLE ALERT
Enable/Disable online alerting

$ MCR PERFDAT_MGR

ENABLE ALERT collection_profile

/OS_TYPE=<os-type|application-name>
/ALERT_FILENAME=<alert-definition-file>
/NODE=<EVA or SNMP node name>

DISABLE ALERT collection_profile

/OS_TYPE=<os-type|application-name>
/NODE=<EVA or SNMP node name>
Alert methods

- OPCOM (default)
- Alert log file (default)
  - PERFDAT$ALERT:PERFDAT_ALERT_<node>.LOG_<date>
- User defined script (optional)
  - Alert block parameter:
    - CRITICAL_CMD
    - CRITICAL_CMD
    - CLEARALERT_CMD
    - SUBMIT_QUEUE
  - Data passed to user defined script:
    - P1 Node name
    - P2 Metric name
    - P3 Statistics
    - P4 Element name
    - P5 Average value of the statistics
    - P6 Critical/Warning threshold or both if this is a CLEAR event
    - P7 Numeric severity level (1=Clear, 2= Warning, 3=Error)
Report automation (automatic graph creation)

The CREATE GRAPH command selects data from collection databases and creates PNG formatted graphs that can be viewed directly with your WEB browser. This command facilitates automated WEB based graphing.

➢ Synatx:

```
CREATE GRAPH [STACKED] statistics_itemlist
   FROM metric_name
   ALIAS alias_name [DATE date]
      [ELEMENT element_name]
      [WHERE filter_list]
   [INTO directory]
   [NAME graph_name]
   [STACKED_OVERLAY | SINGLE_SCALED];
```
CREATE GRAPH

• Layout is defined by a graph layout file (default = PERFDAT$CFG:PERFDAT_CSV2PNG.CFG)

• For detailed description of the layout parameters please refer to:
  o PERFDAT$CFG:PERFDAT_CSV2PNG.CFG
  o DQL$ online help
    $ DQL$ HELP CREATE GRAPH Graph_Cfg_Parameters
  o HPE PERFDAT – DQL$ reference manual
    Section CREATE GRAPH

• Use the DEFINE GRAPH_CFG to define the layout for the adjacent CREATE GRAPH command

• DEFINE HEADER defines the caption of the graph
Create Graph

Enter commands interactively

• Example

$ MCR DQL$
DQL> ATTACH ALIAS DTIPM1_DEFAULT DATE 9-APR-2008;
DQL> DEFINE GRAPH_CFG PERFDAT$CFG:FILL_AREA.CFG;
DQL> DEFINE HEADER “My Stats to look at”;
DQL> CREATE GRAPH iCpuLoad, iDiskMB, iDiskIO FROM SYSTEM ALIAS DTIPM1_DEFAULT
cont> DATE 9-APR-2008 WHERE TIME >= 9-APR-2008 10:00:00, TIME < 9-APR-2008 11:45
cont> NAME SAME_SCALE;

Write commands into a file and execute this file

• Example

$ TYPE PERFDAT$CFG:DQL_TEST_SCRIPT.DQL
ATTACH ALIAS DTIPM1_DEFAULT DATE 9-APR-2008;
DEFINE GRAPH_CFG PERFDAT$CFG:FILL_AREA.CFG;
DEFINE HEADER “My Stats to look at”;
CREATE GRAPH iCpuLoad, iDiskMB, iDiskIO FROM SYSTEM ALIAS DTIPM1_DEFAULT DATE 9-APR-2008 WHERE TIME >= 9-APR-2008 10:00:00, TIME < 9-APR-2008 11:45 NAME SAME_SCALE;

$ MCR DQL$ @PERFDAT$CFG:DQL_TEST_SCRIPT.DQL
CREATE GRAPH example
Single scale for all stats – filled area graph

DQL> ATTACH ALIAS DTIPM1_DEFAULT DATE 9-APR-2008;
DQL> DEFINE GRAPH_CFG PERFDAT$CFG:FILL_AREA.CFG;
DQL> CREATE GRAPH iCpuLOad, iDiskMB, iDiskIO FROM SYSTEM ALIAS DTIPM1_DEFAULT
cont> DATE 9-APR-2008 WHERE TIME >= 9-APR-2008 10:00:00, TIME < 9-APR-2008 11:45
cont> NAME SAME_SCALE;
CREATE GRAPH example
Stats individually scaled—line graph

DQL> ATTACH ALIAS DTIPM1_DEFAULT DATE 9-APR-2008;
DQL> CREATE GRAPH iCpuLOad, iDiskMB, iDiskIO FROM SYSTEM ALIAS DTIPM1_DEFAULT
cont> DATE 9-APR-2008 WHERE TIME >= 9-APR-2008 10:00:00, TIME < 9-APR-2008 11:45
cont> NAME DIFF_SCALE SINGLE_SCALED;

![Graph Image]

<table>
<thead>
<tr>
<th>Min.</th>
<th>Max.</th>
<th>Stat</th>
<th>Unit</th>
<th>Matrix</th>
<th>System</th>
<th>Element</th>
</tr>
</thead>
<tbody>
<tr>
<td>821.70</td>
<td>1537.77</td>
<td>1509.52</td>
<td>iDiskIO</td>
<td>10%</td>
<td>SYSTEM</td>
<td>DTIPM1</td>
</tr>
<tr>
<td>133.00</td>
<td>242.95</td>
<td>219.74</td>
<td>iDiskLoad</td>
<td>10%</td>
<td>SYSTEM</td>
<td>DTIPM1</td>
</tr>
<tr>
<td>12.26</td>
<td>0.46</td>
<td>50.36</td>
<td>iDiskMB</td>
<td>Open</td>
<td>DTIPM1</td>
<td>OPENVMS</td>
</tr>
</tbody>
</table>
CREATE GRAPH example
Stats individually scaled—filled area graph

DQL> ATTACH ALIAS DTIPM1_DEFAULT DATE 9-APR-2008;
DQL> CREATE GRAPH iCpuLOad, iDiskMB, iDiskIO FROM SYSTEM ALIAS DTIPM1_DEFAULT
cont> DATE 9-APR-2008 WHERE TIME >= 9-APR-2008 10:00:00, TIME < 9-APR-2008 11:45
cont> NAME DIFF_SCALE SINGLE_SCALED;
Report automation - Workflow

• Create a DCL script that
  o Creates a valid DQL$ scripts (contains all required DQL commands to create the graphs as required)
  o Execute the DQL script from the DCL script
    ➢ $ MCR DQL$ @<DQL script>
  o Use DCL COPY commands to move the PNG files into the target directories
  o Reschedule the DCL script for execution

• Submit the the DCL script into a batch script
Supported Versions

• HPE PERFDAT V4.8 ECO 1 is supported on:
  OpenVMS AXP V7.3-2 – V8.4
  OpenVMS IA64 V8.2 – V8.4-2L1

• HPE PERFDATGUI V4.3.1:
  Windows XP/7/8.1/10

• Upcoming releases (Q1/2018):
  o HPE PERFDAT V4.9
    ✓ Bug-fixes
    ✓ Supports VSI OpenVMS AXP V8.4-2L2
HPE PERFDAT links

For more information about HPE PERFDAT please contact:

- HPE PERFDAT Support: perfdat@hpe.com
- Wolfgang Burger: wolfgang.burger@hpe.com
- Our partner: Compinia GmbH & Co. KG: perfdat@compinia.com

Download: http://www.perfdat.com/
Questions