HP PERFDAT - OpenVMS Performance Solution

OpenVMS Bootcamp 2015, Nashua NH

DI. Dr. Wolfgang Burger
HPE TS/Austria

© 2015 Hewlett-Packard Enterprise Ltd.

Agenda

- Basic Concepts & Components
- Special focus
  - Device statistics
    - Meaning
    - Most important stats to look at
  - How-To
    - Configure selective file filtering (Demo)
    - Create user defined (calculated) stats (Demo)
    - Online alerting (Demo)
    - Report automation (automatic graph creation)

Performance management

- OpenVMS performance management manual
  - …waiting until a problems cripples a system before addressing system mamangement is not performance management, rather it is crisis management …

    Performance management involves:
    - Systematically measuring the system
    - Gathering and analyzing the data
    - Evaluating trends
    - Archiving data to maintain a performance history

Performance management (contd.)

- In addition
  - Baseline your system
    - Every system is special
  - Communicate
    - Publish performance of your systems
      - Demonstrate that you care about the system
      - Demonstrate that you do a good job
      - In case of a performance issue you have to cooperate with other people to resolve it
        - A picture says more than thousend words
  - Get notified about exceptional system behaviour
    - Don‘t be triggered by end-users
    - Demonstrate that you care about the system
    - Demonstrate that you do a good job

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
  - 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.
  - Easy to manage and control
  - Automated data management without any system management intervention
    - Ability to manage huge amounts of data (> 1TByte)
    - Archive and housekeeping functionality
  - 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

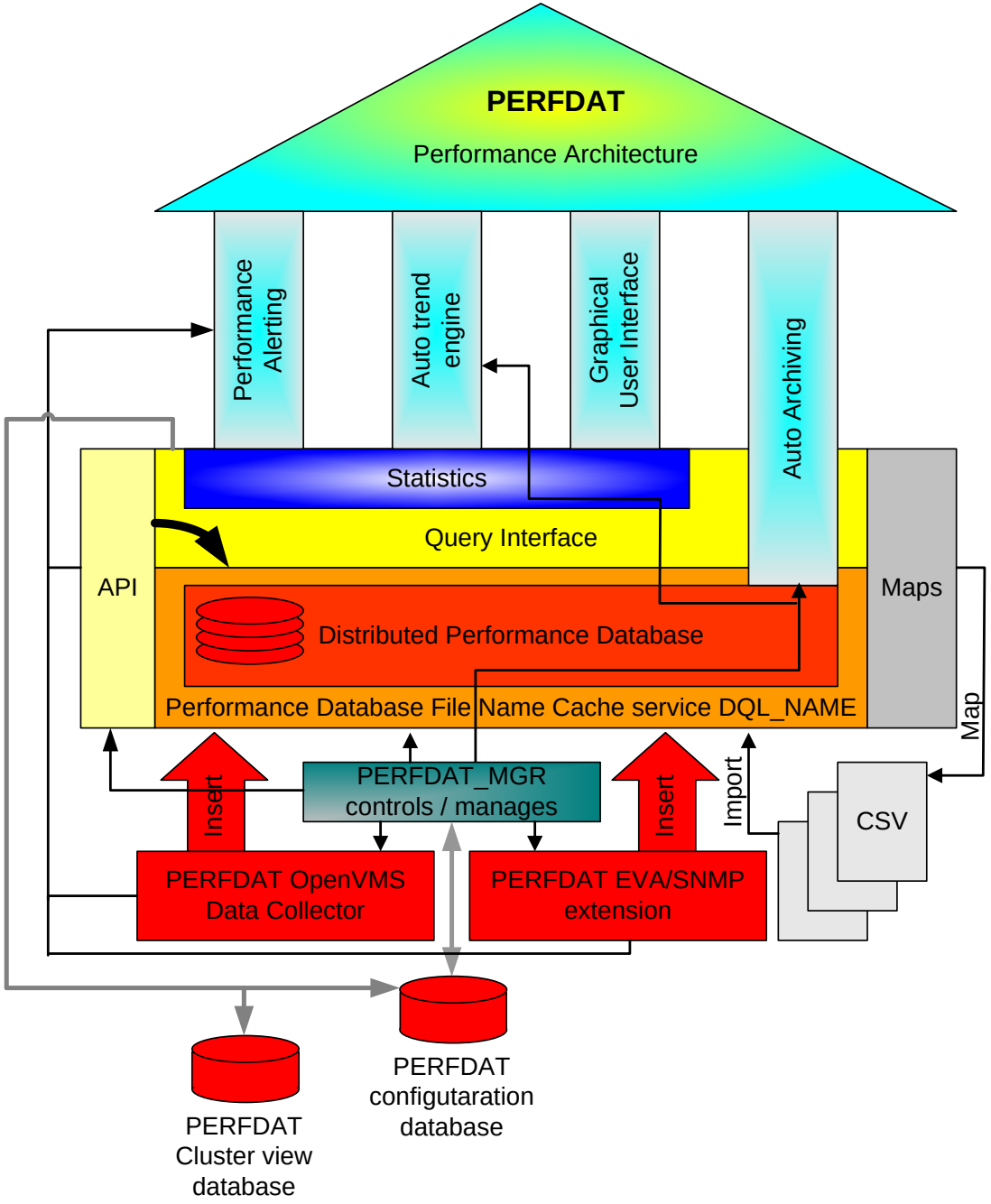
Requirements (contd.)

- 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
  - Performance depend on external, shared storage
  - Systems are coupled via shared storage
- Serious performance management sometimes requires one „to look over the rim of the tea cup“
  - Attached shared Storage
  - Systems accessing the same shared storage
    - Solaris
    - Linux
    - ...

Architecture



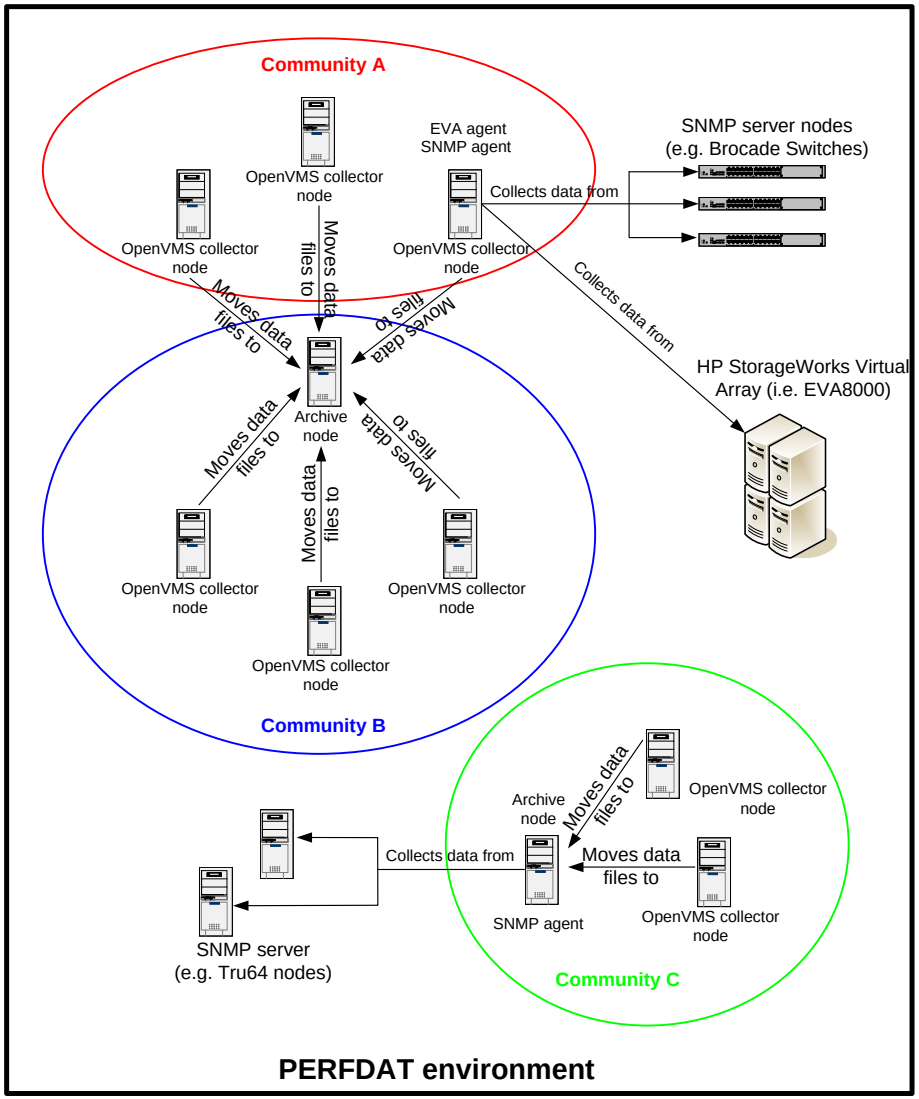
HP PERFDAT Components

- 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 environment

- The PERFDAT environment consists of so called communities. A community is a logical partition of the whole environment and defines the database view when accessing the data via any system within a community. All systems of particular interest can be configured within the context of a community. No rules exist that limit the configuration of such communities (such as cluster boundaries, location of the systems etc.). The number of possible communities ranges from one to the total number of systems within the whole environment.

PERFDAT environment (contd.)



HP 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 environment (contd.)

- The role of the systems within a community is defined by the SW-components running on the systems.
  - OpenVMS collector system
  - SNMP agent system (collects data from SNMP server systems)
  - EVA agent (collects HSV data)
  - Archive system
  - Access server
  - SNMP server system (provides performance data via SNMP)

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
  - 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 metrices

- 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

HP 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

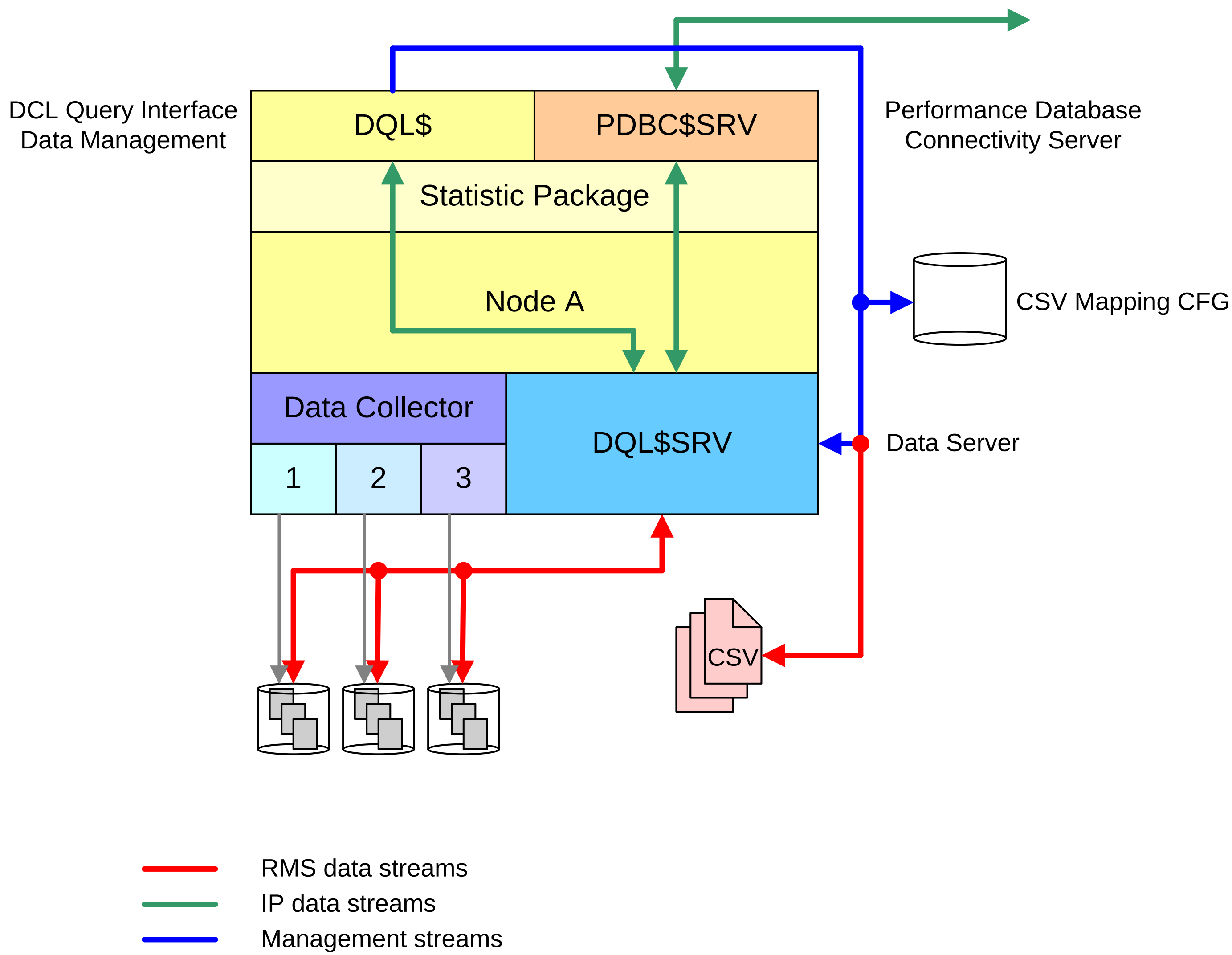
PERFDAT Query Interface (DQL)

- PERFDAT Query Interface architecture allows 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 analyzing 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

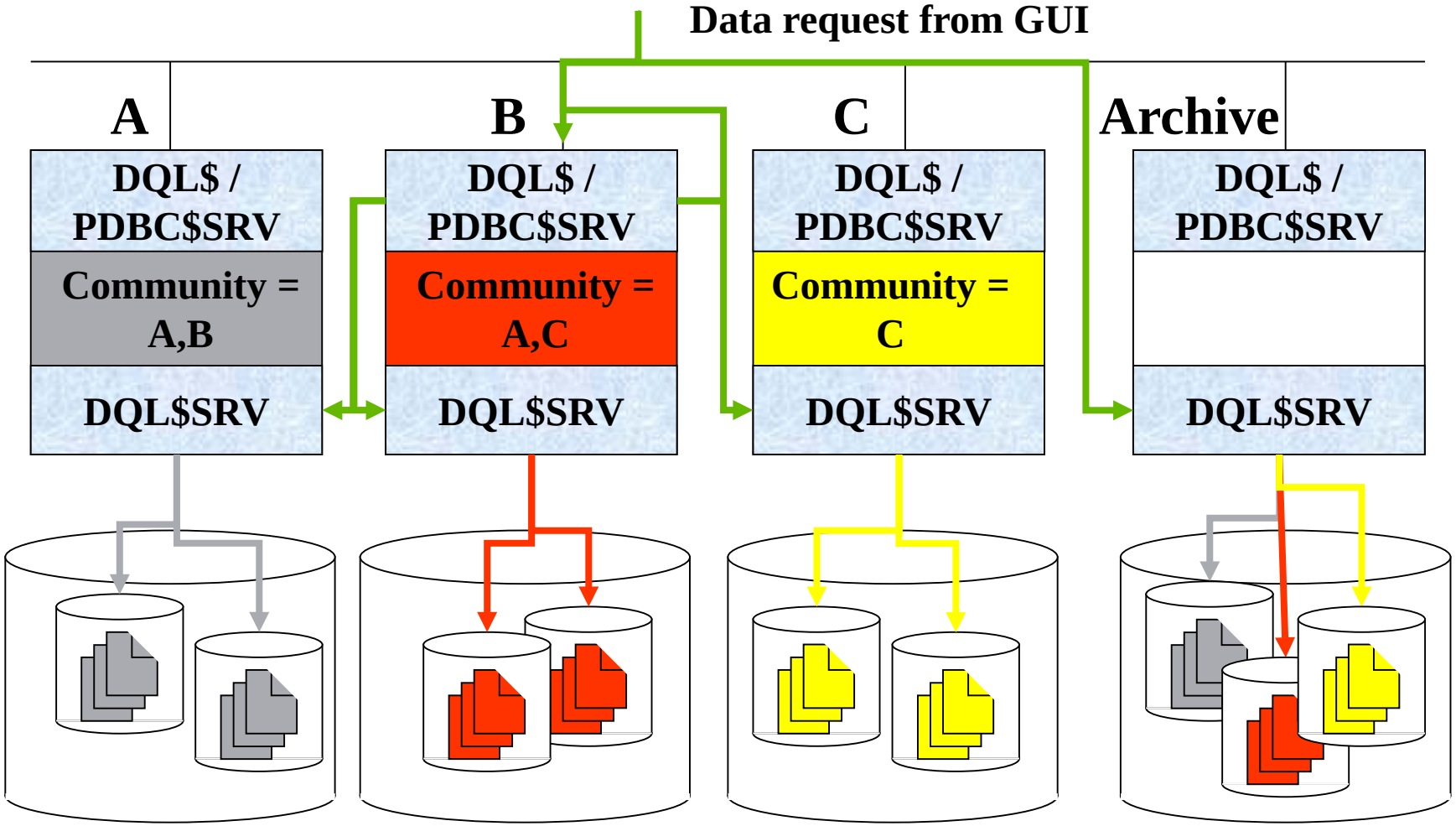


DCL Query Interface
Data Management
DQL$
PDBC$SRV
Performance Database
Connectivity Server
Statistic Package
Node A
CSV Mapping CFG
Data Collector
1
2
3
DQL$SRV
Data Server
CSV
RMS data streams
IP data streams
Management streams

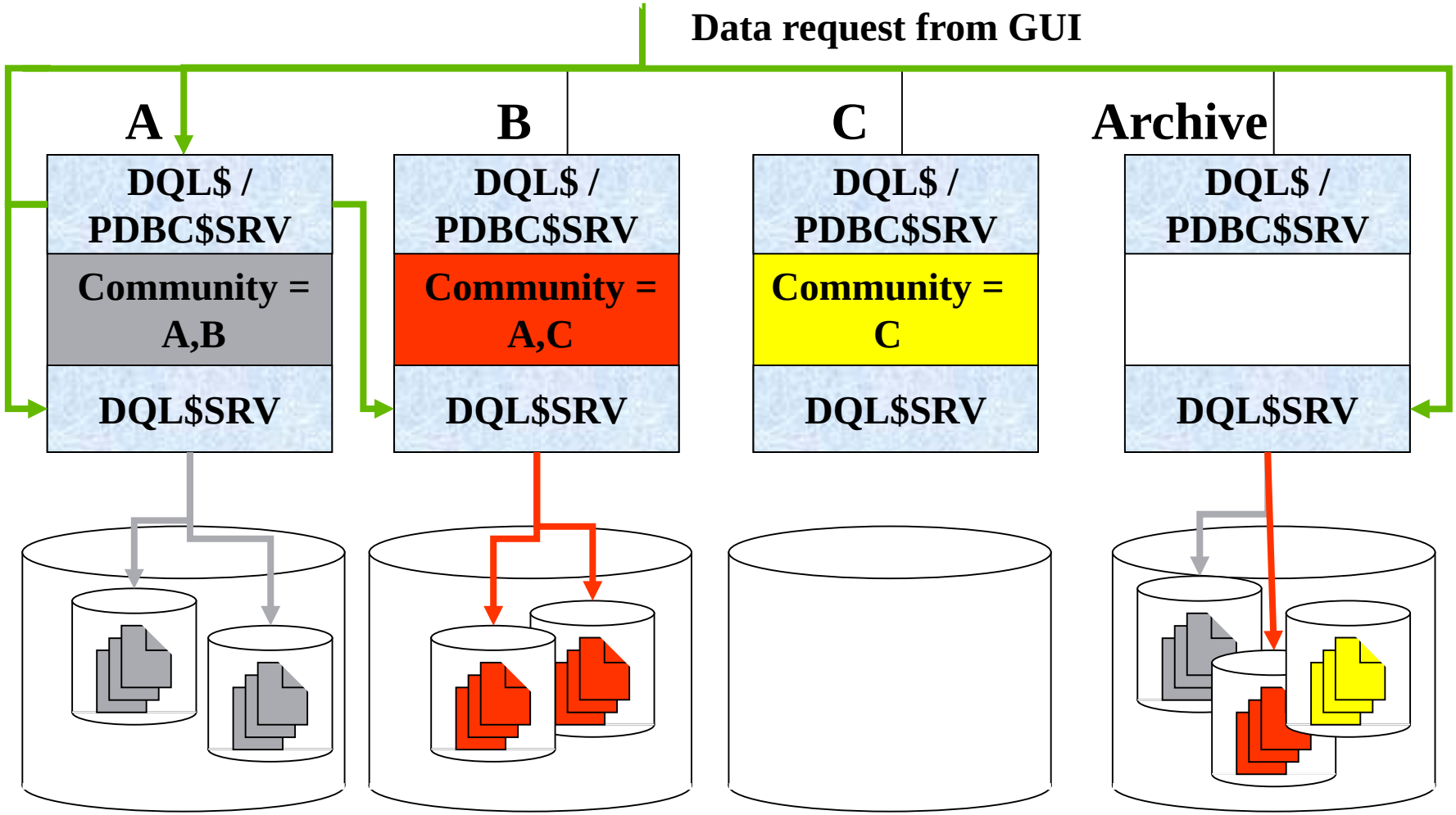
Query Interface - Community

- 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

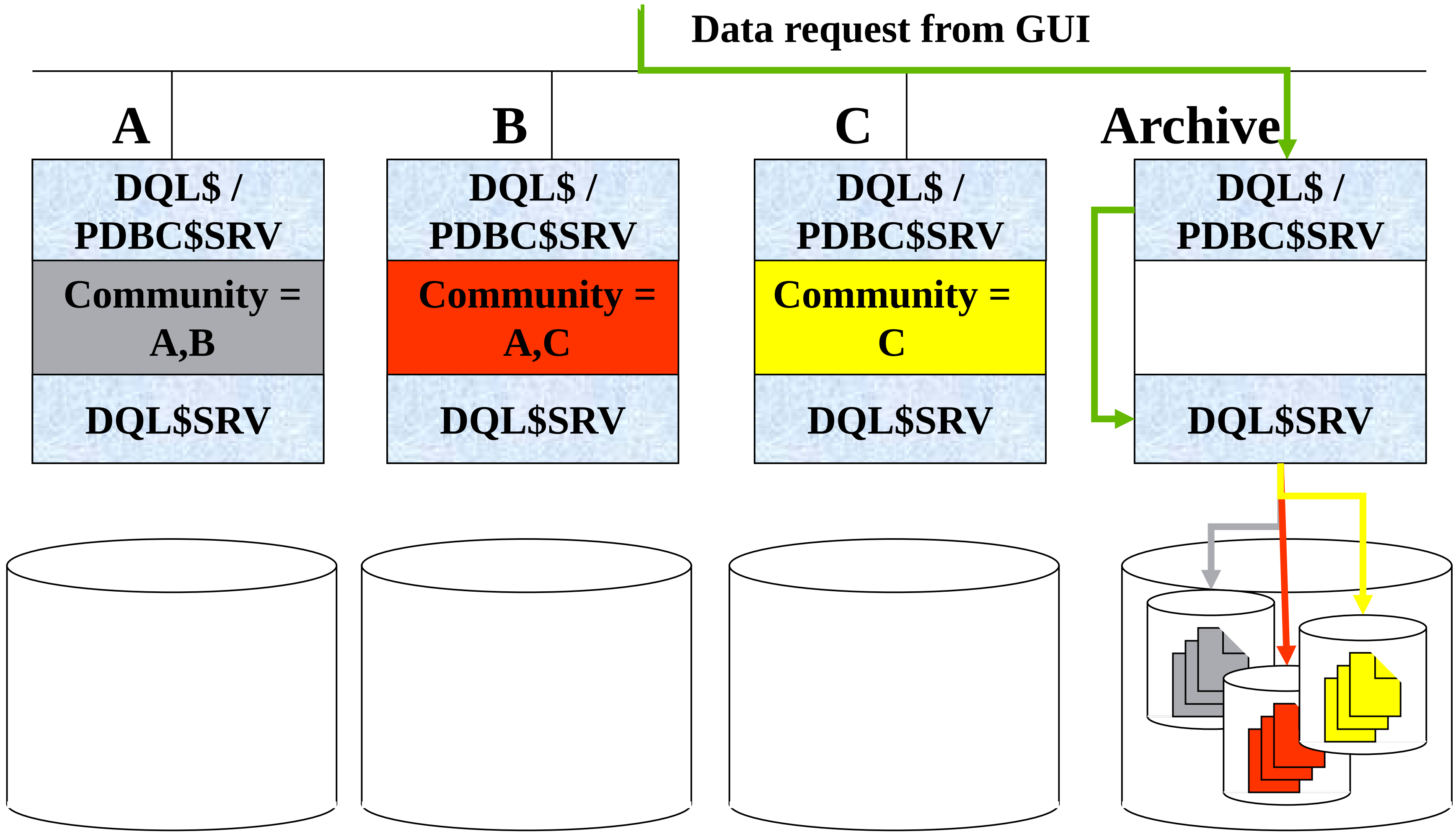
Query Interface - Data Flow



Query Interface - Data Flow



Query Interface - Data Flow



HP 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 compareable to PCM import files
- An alert definition file with valid alert blocks are a prerequiste to enable online performance alerting
- Max. number of elements tracked by a single alert block is 4096

HP 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
- Automatic selection and compression of performance statistics for trend- and capacity analysis.
- 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

Statistics	Description	Unit
iQIOs	Total QIO rate on device	[IO/s]
iRqs	Total device IO request rate	[IO/s]
iIOs	Total service IO (passing START_IO routine) rate	[IO/s]
iIOSp	Total split service IO (passing START_IO routine) rate	[IO/s]
iAbs	Total Aborted	[IO/s]
iMbs	Total Throughput	[MB/s]
iRQtime	IO Request time	[ms]
iRQtimeMax	MAX IO Request time during last sample interval	[ms]
iRQrespAcc	Accuracy of IO Request time	[+/-%]
iIOtime	IO Service time	[ms]
iIOtimeMax	MAX IO Service time during last sample interval	[ms]
iIOrspAcc	Accuracy of IO Service time	[+/-%]
iRdQIOs	Read QIO rate on device	[IO/s]
iRdRqs	Read device IO request rate	[IO/s]
iRdIOs	Read service IO (passing START_IO routine) rate	[IO/s]
iRdIOSp	Read split service IO (passing START_IO routine) rate	[IO/s]
iRdAbs	Read Aborted	[IO/s]
iRdMbs	Read Throughpu	[MB/s]
iRdRQtime	Read IO Request time	[ms]
iRdRQtimeMax	Read MAX IO Request time during last sample interval	[ms]
iRdRQrespAcc	Accuracy of Read IO Request time	[+/-%]
iRdIOtime	Read IO Service Time	[ms]
iRdIOtimeMax	Read MAX IO Service time during last sample interval	[ms]
iRdIOrspAcc	Accuracy of Read IO Service time	[+/-%]
iWrQIOs	Write QIO rate on device	[IO/s]
iWrRqs	Write device IO request rate	[IO/s]
iWrIOs	Write service IO (passing START_IO routine) rate	[IO/s]
iWrIOSp	Write split service IO (passing START_IO routine) rate	[IO/s]
iWrAbs	Write Aborted	[IO/s]
iWrMbs	Write Throughput	[MB/s]
iWrRQtime	Write IO Request time	[ms]
iWrRQtimeMax	Write MAX IO Request time during last sample interval	[ms]
iWrRQrespAcc	Accuracy of Write IO Request time	[+/-%]
iWrIOtime	Write IO Service Time	[ms]
iWrIOtimeMax	Write MAX IO Service time during last sample interval	[ms]
iWrIOrspAcc	Accuracy of Write IO Service time	[+/-%]
iCtlQIOs	Ctrl QIO rate on device	[IO/s]
iCtlRqs	Ctrl device IO request rate	[IO/s]
iCtlAbs	Ctrl Aborted	[IO/s]
iCtlRQtime	Ctrl IO Request time	[ms]
iCtlRQtimeMax	Ctrl MAX IO Request time during last sample interval	[ms]
iCtlRQrespAcc	Accuracy of Ctrl IO Request time	[+/-%]
iQlen	Device IO queue length	[#]
iIOthres	IO request threshold	[IO/s]
iElementCnt	Element count	[#]

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)

Hewlett Packard Enterprise

- 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

Statistics	Description	Unit
Σ $2msecPerc	Percentage of < 2ms I/Os	%
iIOs	Number of I/Os within last sample interval	[IO]
iIOtime	Avg I/O Service time during last sample interval	[ms]
iIOtimeMax	Max I/O Service time during last sample interval	[ms]
2ms	Number of I/Os completed within 2ms	[IO]
4ms	Number of I/Os completed between 2ms and 4ms	[IO]
6ms	Number of I/Os completed between 4ms and 6ms	[IO]
10ms	Number of I/Os completed between 6ms and 10ms	[IO]
20ms	Number of I/Os completed between 10ms and 20ms	[IO]
30ms	Number of I/Os completed between 20ms and 30ms	[IO]
40ms	Number of I/Os completed between 30ms and 40ms	[IO]
50ms	Number of I/Os completed between 40ms and 50ms	[IO]
100ms	Number of I/Os completed between 50ms and 100ms	[IO]
200ms	Number of I/Os completed between 100ms and 200ms	[IO]
300ms	Number of I/Os completed between 200ms and 300ms	[IO]
400ms	Number of I/Os completed between 300ms and 400ms	[IO]
500ms	Number of I/Os completed between 400ms and 500ms	[IO]
1000ms	Number of I/Os completed between 500ms and 1000ms	[IO]
iGT 1sec	Number of I/Os not completed within 1000ms	[IO]
iSampleTime	Sample Time	[sec]

How-To
Configure selective file filtering

- Any HP 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

Hewlett Packard Enterprise

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*]:
     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]
```

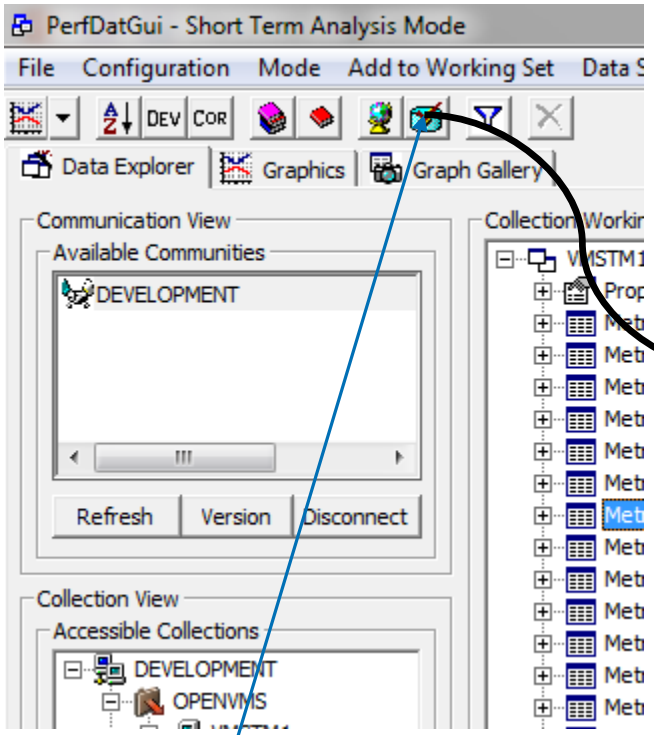
- Define the new/modified profile as the default profile in the auto-start table, if the default collection shall start with this profile whenever the HP PERFDAT OpenVMS data collector starts

```
$ MCR PERFDAT_MGR MODIFY AUTOSTART <node-name>
```

How-To
Create user defined stats

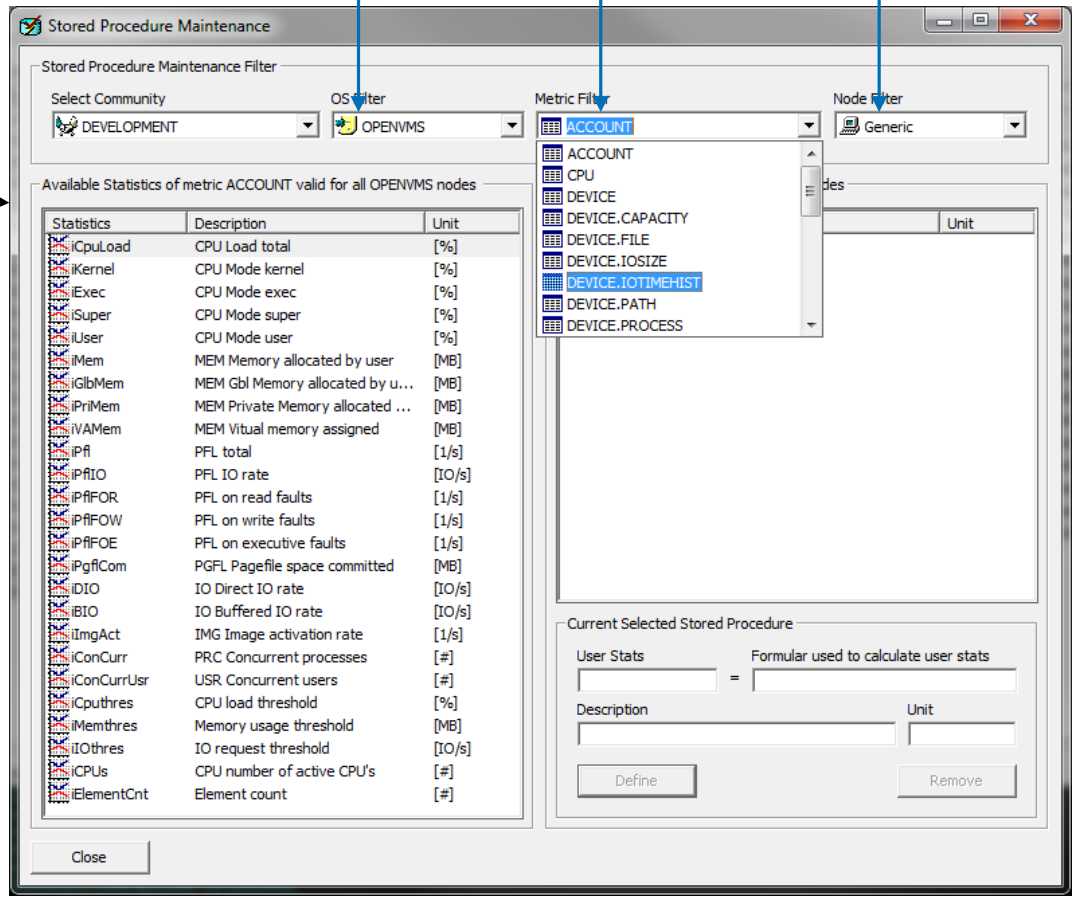
- User defined stats are:
  - Calculated statistics (measures)
  - Can be defined node specific
  - Can be accessed as if they are part of the data collection
- Usefull whenever you need a different view on the performance data
  - i.e. 2ms of DEVICE.IOTIMEHIST
    - provides I/Os per second which completed < 2ms
    - You need to know %
    - new user stat $2msecPerc = 2msec / iIOS * 100

How-To
Create user defined stat - GUI

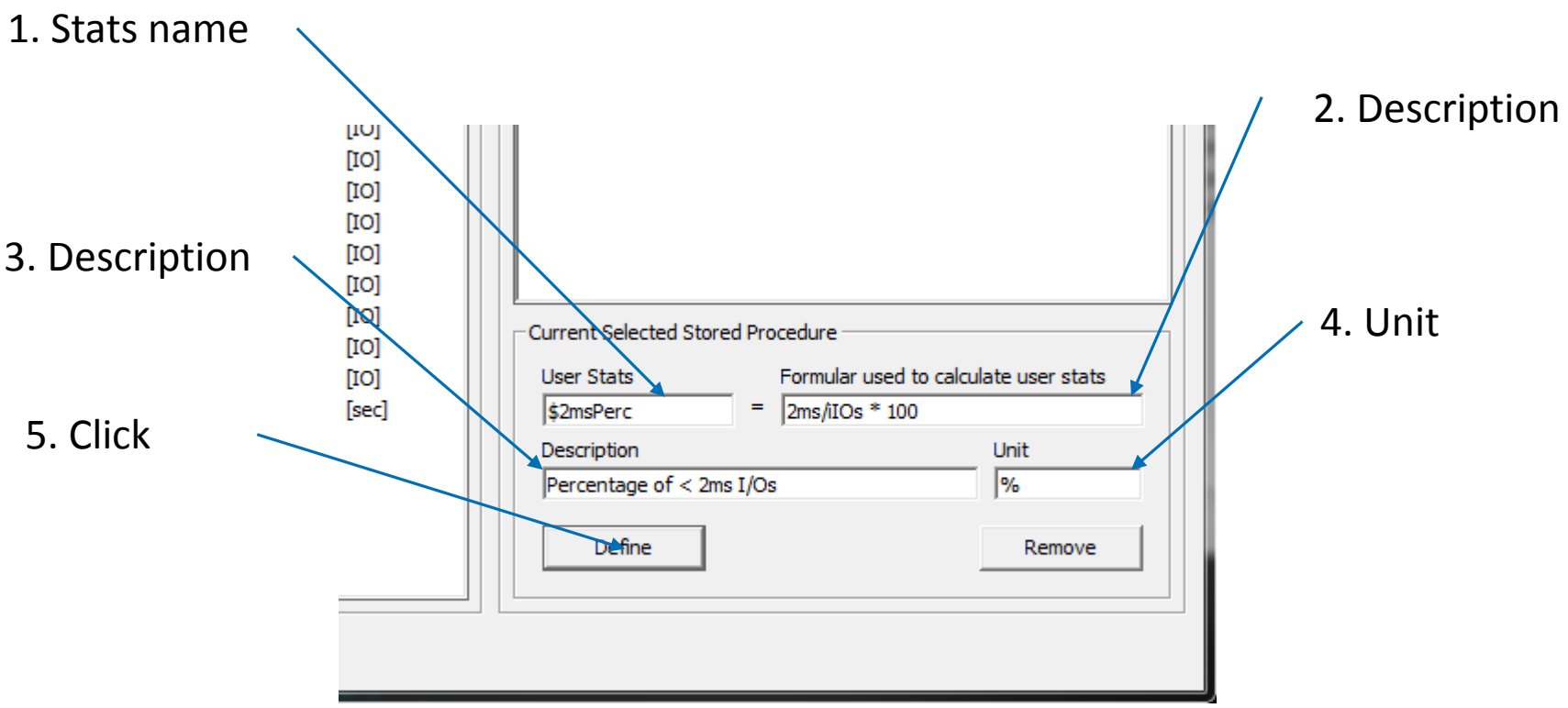


1. Select toolbar button

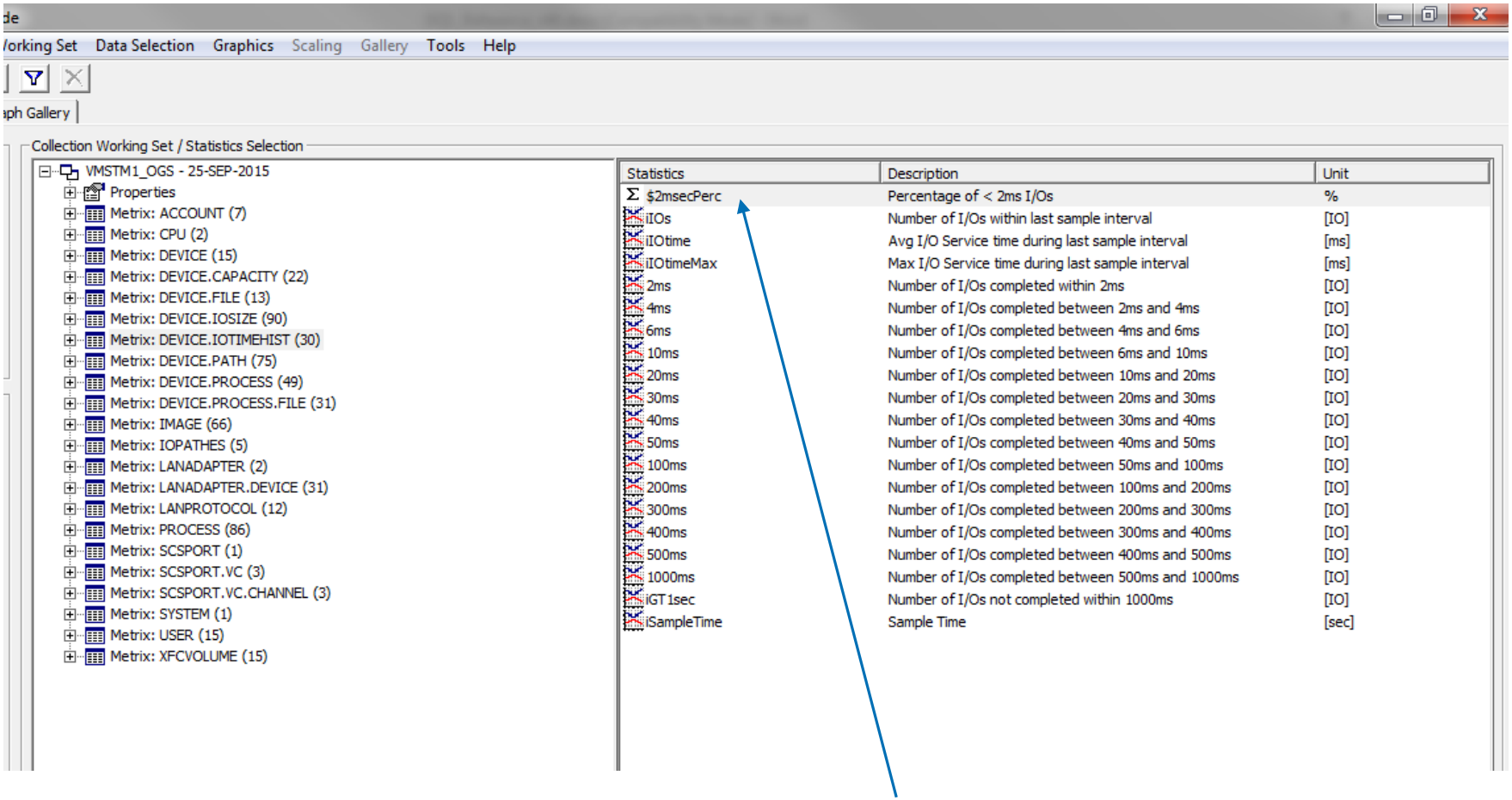
2. Select OS Filter/Metric Filter/Node Filter



How-To
Create user defined stat - GUI



How-To
Create user defined stat - GUI



User stat available in stats list

How-To
Create user defined stat – DQL$

Hewlett Packard Enterprise

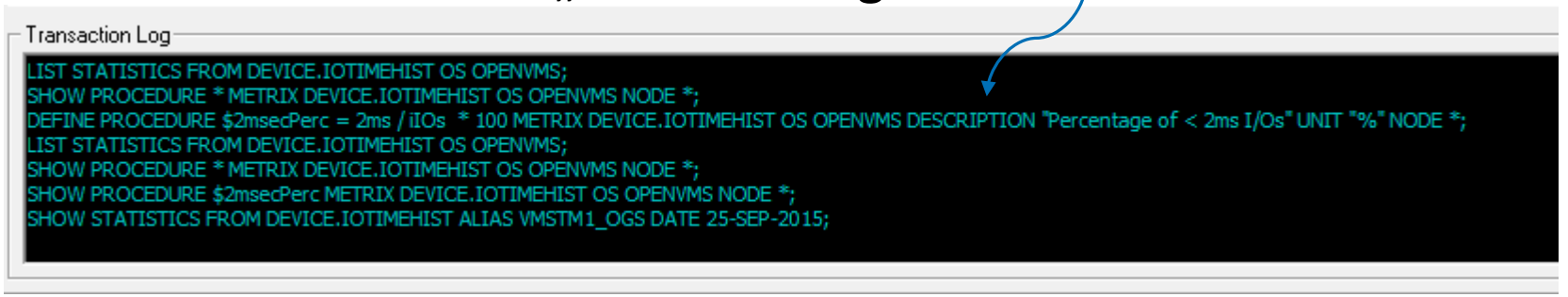
Syntax:

```
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 HP 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:

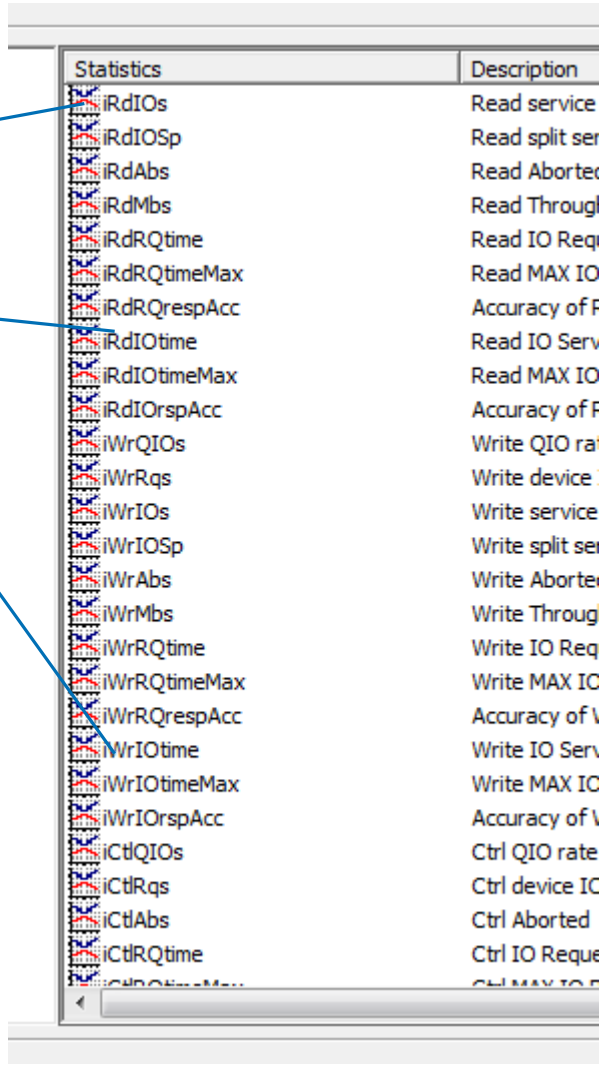
- Configuration keywords:
  - OSTYPE:
  - METRIX:
  - STATISTICS:
  - SCALED_BY:
  - OPERATOR
  - ELEMENTS
  - STACKED
  - STACKED_ELEM_NAME:
  - SAMPLE_COUNT:
  - WARNING_THRESHOLD:
  - WARNING_CMD:
  - CRITICAL_THRESHOLD:
  - CRITICAL_CMD:
  - CLEARALERTMSG:
  - CLEARALERT_CMD:
  - SUBMIT_QUEUE:

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 paramters

- Please refer to:
  - PERFDAT_MGR online help
    - $ MCR PERFDAT_MGR HELP ENABLE ALERT
  - HP 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=Claer, 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$CFG:PERFDAT_CSV2PNG.CFG)
- For detailed description of the layout parameters please refer to:
  - PERFDAT$CFG:PERFDAT_CSV2PNG.CFG
  - DQL$ online help
    - $ DQL$ HELP CREATE GRAPH Graph_Cfg_Parameters
  - HP PERFDAT – DQL$ reference manual
    - Section CREATE GRAPH
- Use the DEFINE GRAPH_CFG to define the layout for the adjecent CREATE GRAPH command
- DEFINE HEADER defines the caption of the graph

Create Graph

- Enter commds interactively
  - Example

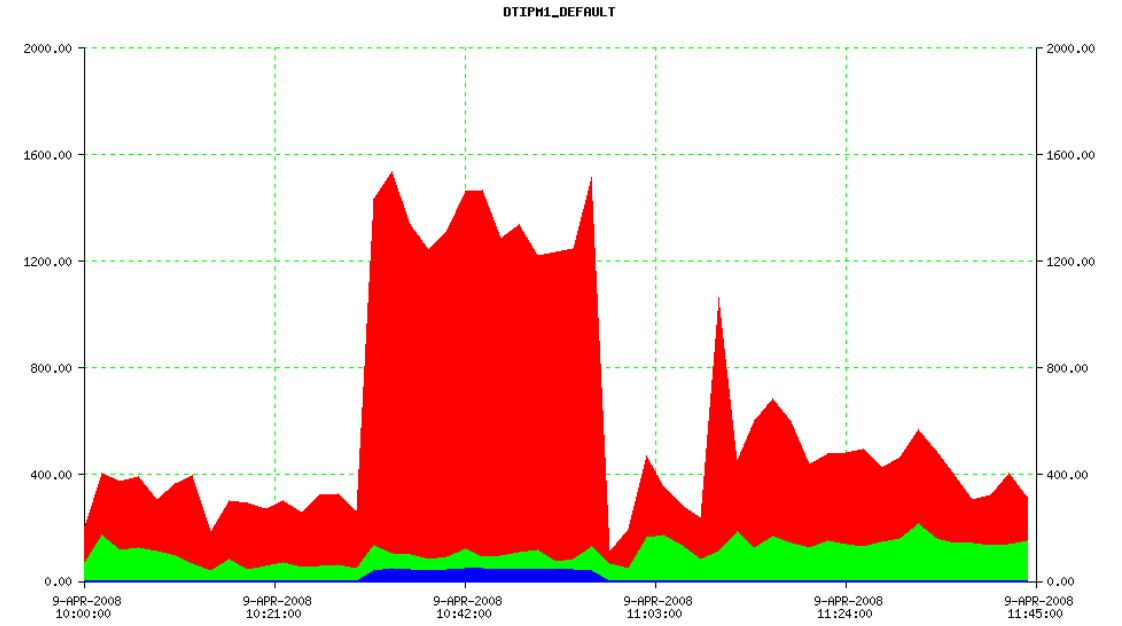
```
$ MCR DQL$
DQL> ATTACH ALIAS DTIPM1_DEFAULT DATE 9-APR-2008;
DQL> DEFINE GRAPH_CFG PERFDAT$CFG:FILL_AREA.CFG;
DQL> DEFINE HEADER “My Stats to look at”;
DQL> CREATE GRAPH iCpuLoad, iDiskMB, iDiskIO FROM SYSTEM ALIAS DTIPM1_DEFAULT
cont> DATE 9-APR-2008 WHERE TIME >= 9-APR-2008 10:00:00, TIME < 9-APR-2008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

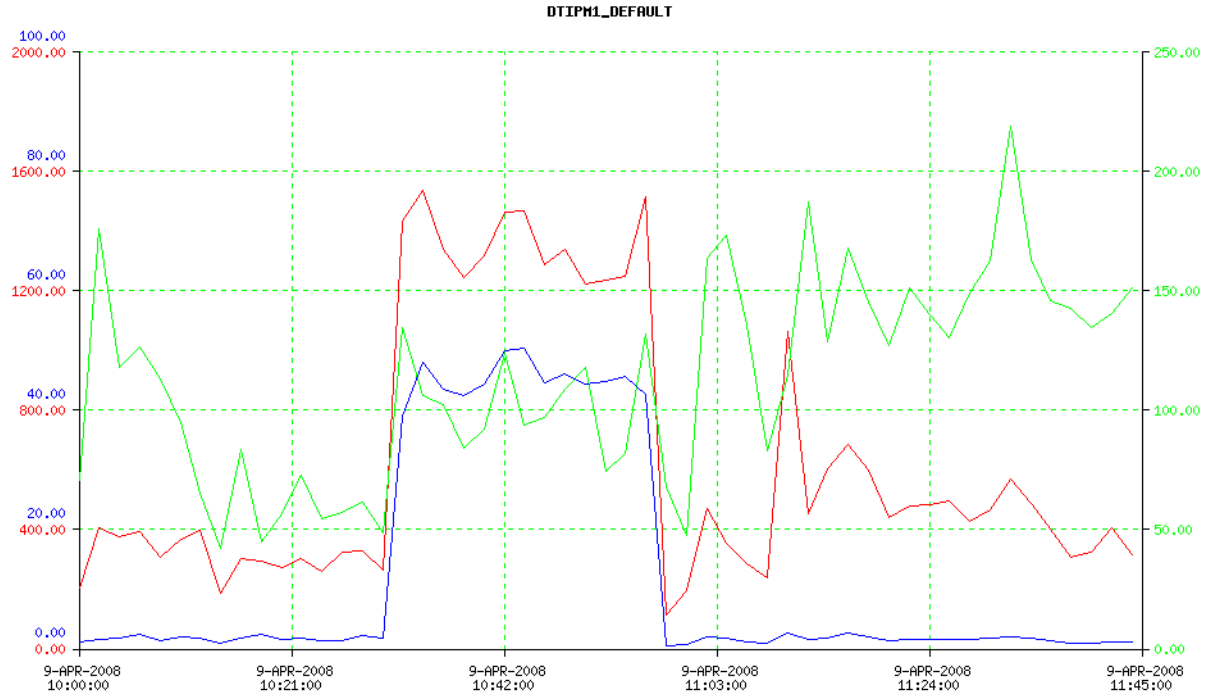
```
DQL> ATTACH ALIAS DTIPM1_DEFAULT DATE 9-APR-2008;
DQL> DEFINE GRAPH_CFG PERFDAT$CFG:FILL_AREA.CFG;
DQL> CREATE GRAPH iCpuLOad, iDiskMB, iDiskIO FROM SYSTEM ALIAS DTIPM1_DEFAULT
cont> DATE 9-APR-2008 WHERE TIME >= 9-APR-2008 10:00:00, TIME < 9-APR-2008 11:45
cont> NAME SAME_SCALE;
```



Avg.	Min.	Max.	Stats	Unit	Metrix	System	Element
627.70	111.77	1535.52	iDiskIO	[IO/s]	SYSTEM	DTIPM1	OPENVMS
113.20	42.35	218.74	iCpuLoad	[%]	SYSTEM	DTIPM1	OPENVMS
12.26	0.46	50.36	iDiskMB	[MB/s]	SYSTEM	DTIPM1	OPENVMS

CREATE GRAPH example
Stats individually scaled– line graph

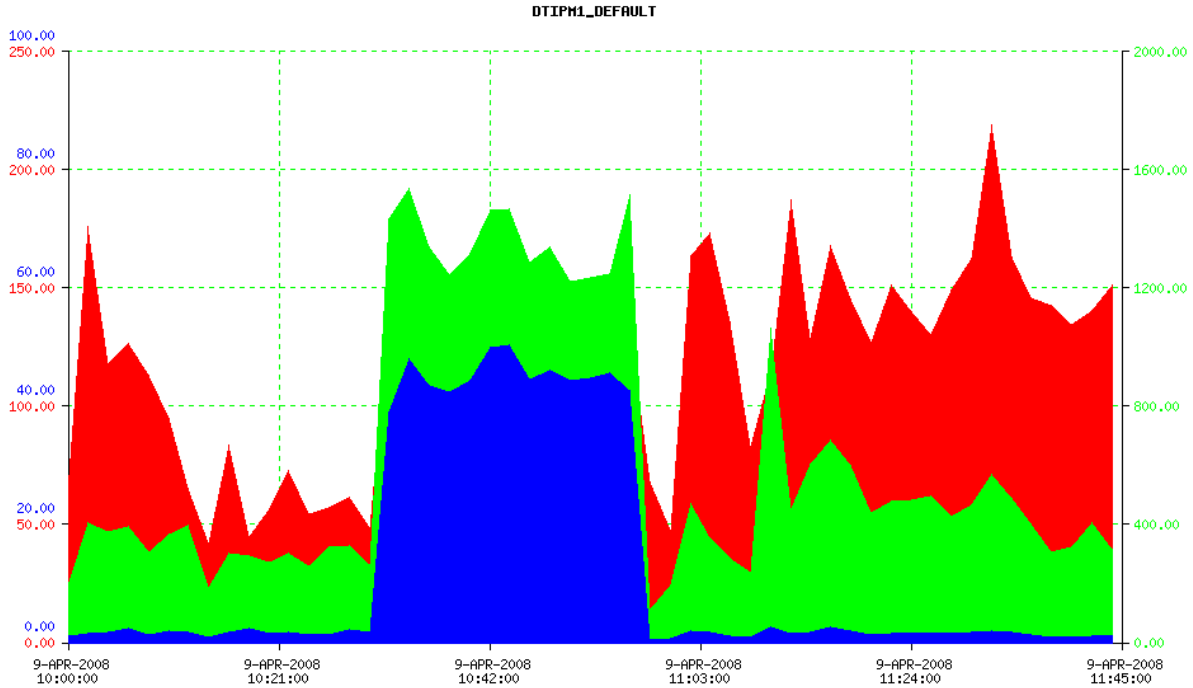
- DQL> ATTACH ALIAS DTIPM1_DEFAULT DATE 9-APR-2008;
- DQL> CREATE GRAPH iCpuLOad, iDiskMB, iDiskIO FROM SYSTEM ALIAS DTIPM1_DEFAULT
- cont> DATE 9-APR-2008 WHERE TIME >= 9-APR-2008 10:00:00, TIME < 9-APR-2008 11:45
- cont> NAME DIFF_SCALE SINGLE_SCALED;



Avg.	Min.	Max.	Stats	Unit	Metrix	System	Element
627.70	111.77	1535.52	iDiskIO	[IO/s]	SYSTEM	DTIPM1	OPENVMS
113.20	42.35	218.74	iCpuLoad	[%]	SYSTEM	DTIPM1	OPENVMS
12.26	0.46	50.36	iDiskMB	[MB/s]	SYSTEM	DTIPM1	OPENVMS

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;



Avg.	Min.	Max.	Stats	Unit	Metrix	System	Element
113.20	42.35	218.74	iCpuLoad	[%]	SYSTEM	DTIPM1	OPENVMS
627.70	111.77	1535.52	iDiskIO	[IO/s]	SYSTEM	DTIPM1	OPENVMS
12.26	0.46	50.36	iDiskMB	[MB/s]	SYSTEM	DTIPM1	OPENVMS

Report automation - Workflow

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

Supported Versions

- HP PERFDAT V4.6 is supported on:
  - OpenVMS AXP V7.3-2 – V8.4
  - OpenVMS IA64 V8.2 – V8.4
- HP PerfdatGUI V4.3.0:
  - Windows XP/7/8.1
- Upcoming releases (end Q4/2015):
  - HP PERFDAT V4.7
    - Bug-fixes
    - Supports OpenVMS V8.4 1H1
  - HP PerfdatGUI V4.3.1
    - Bug-fixes
    - Supports Windows 10

HP PERFDAT links

- For more information about HP PERFDAT please contact:
  - HP 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/


invent