

Performance Data Collector for OpenVMS (TDC)

peter r. ranisch

email: openvms@chello.at

© 2003 Hewlett-Packard Development Company, L.P.
The information contained herein is shown for clarity without

Preparation

- <http://h71000.www7.hp.com/openvms/products/tdc/>
- \$ @SYS\$STARTUP:TDC\$STARTUP
- SYS\$COMMON:[TDC]TDC_HOW_TO_RUN.TXT
- \$ set command sys\$common:<tdc>tdc\$dcl

- CMKRNL, LOG_IO, NETMBX, PHY_IO, SYSLCK,
SYSPRV, WORLD

Development Goals

- Provide useful system-performance data...
 - Through an **OPEN**...
 - **FLEXIBLE**...
 - And **EXTENSIBLE** architectural **FRAMEWORK**
- Provide data to applications...
 - Through a **FILE**, ...
 - Or **LIVE**, ...
 - Or both
- Track OpenVMS development going forward...
 - On **Alpha** systems (OpenVMS V7.3-2+)
 - On **Industry Standard** platforms (OpenVMS V8.2+)

May 16, 2005

page 3

TDC is:

- A provider of system configuration and performance data for use by other software applications.
- A “bulk” data provider that allows applications to capture groups of performance-related system metrics.
- A tool to manage collection of system performance data.
- A tool to manage extraction of data from a file.
- A component to be integrated into applications.

May 16, 2005

page 4

TDC is **NOT**:

- A “system service” that offers programmers flexibility to request individual performance metrics of interest.
- A complete system performance analysis solution.
- A standalone application like MONITOR, ECP, T4, etc.

May 16, 2005

page 5

Target Audience for TDC

- TDC **IS** targeted at:
 - Software developers involved in analyzing or monitoring OpenVMS system performance on Alpha or Industry-Standard Server platforms...
 - HP customer, in-house
 - ISVs
 - HP/OpenVMS
- TDC is **NOT** targeted at:
 - End-users, including system managers

May 16, 2005

page 6

TDC Components (1)

- An **Engine** that:
 - **Manages** collection and extraction of performance metrics.
 - **Coordinates** activities of “Processor Modules” that collect and/or process system performance data.
 - Is **fully accessible** through a supported Application Programming Interface (**API**).

May 16, 2005

page 7

TDC Components (2)

- A set of **Processor Modules** that:
 - Perform data collection and/or extraction tasks, timing, file access, etc. using the Engine’s API.
 - Together, provide some **800** performance-related **metrics**.
 - **Can be extended...**
 - In the field
 - By ISVs, etc.
 - By customers

May 16, 2005

page 8

TDC Components (3)

- A DCL command-line interface **Application** that uses the Engine's API to:
 - **Collect** data and, **optionally**, store it in a file.
 - **Extract** data from a file.
 - **Convert** data to various formats.

May 16, 2005

page 9

TDC Components (4)

- A **Software Developer's Kit (SDK)** that provides:
 - A specification of the TDC API
 - A description of metrics provided "out-of-the-box" by TDC
 - A Programmer's Guide and technical overview
 - Header files
 - Code samples

May 16, 2005

page 10

Operational Characteristics (Data-1)

- A **COLLECTION** is a sequence of snapshots
- Data is periodically collected and aggregated into a **SNAPSHOT**
 - A snapshot represents the state of the system at a point in time
 - Two snapshots are maintained...
 - Current
 - Previous
 - Snapshots are the units in which data is made available

May 16, 2005

page 11

Operational Characteristics (Data-2)

- A snapshot contains **RECORD SETs**
 - A record set represents all data records of a particular type (e.g., CPU utilization) created for one snapshot
 - One record set per record type per snapshot
- Data **records can be linked**
 - Records in one record set can be linked to records in another record set within the same snapshot
 - Data records can be hierarchical within a snapshot
- Record sets can be **persistent**
 - Created once and made available as a part of each succeeding snapshot
 - Created as necessary and retained until updated

May 16, 2005

page 12

Operational Characteristics (Runtime)

- **Collecting Data**
 - Incorporates data **collection and utilization**
 - At each snapshot:
 - Data records are produced and aggregated into a snapshot
 - The snapshot is then immediately available for utilization
- **Extracting Data from a File**
 - Incorporates data **extraction and utilization**
 - A snapshot is extracted from a file:
 - Data records are recreated and aggregated into a snapshot
 - The snapshot is then immediately available for utilization

May 16, 2005

page 13

Processor Modules

- The TDC Engine delegates responsibility for producing and utilizing data records to Processor Modules.
- A Processor Module is a functional unit that performs one or more tasks related to collecting or processing system performance data.
- Processor Modules are packaged in shareable images and are loaded at runtime.
- A Processor Module presents itself to TDC, through a TDC-defined interface, as an autonomous entity.
 - PMs packaged within a single shareable image can interact among themselves in the background, but each must be capable of independently interacting with TDC.

May 16, 2005

page 14

Categories of Processor Modules

- **PRODUCER**: produces data records
- **CONSUMER**: processes data records provided by producers
- **TIMER**: controls collection and/or extraction pacing
- **EXTRACTOR**: provides data from a file
- **FILTER**: screens records provided by a producer or extractor
- **GROUPE**: alias for a set of processor modules
- **SUPPORTER**: provides support services for other processor modules
- A combination of the above

May 16, 2005

page 15

Out-of-the-Box Processor Modules (1)

Name	Description	Type	# Metrics
TDC_TIMR	Default collection timer	TIMER	N/A
TDC_PDSC	Manages processor module metadata	PRODUCER	N/A
TDC_TXTCVT	Converts data to text formats	CONSUMER	N/A
TDC_COLFIL	Manages data file access	CONSUMER, EXTRACTOR	N/A
TDC_COLHDR	Provides collection and system descriptions	PRODUCER	24
TDC_LWS	Support services	SUPPORTER	N/A

May 16, 2005

page 16

Out-of-the-Box Processor Modules (2)

Name	Description	Type	# Metrics
DEFAULT	Provides default set of processor modules	GROUPER	N/A
ALL	Provides all OOTB processor modules	GROUPER	N/A
CLUSTER	Provides cluster-related processor modules	GROUPER	N/A

May 16, 2005 page 17

Out-of-the-Box Processor Modules (3)

Name	Description	Type	# Metrics
ADP	Contents of ADP list	PRODUCER	14
CLU	VMSCUSTER configuration	PRODUCER	23
CPU	CPU utilization	PRODUCER	25
CPS	Cluster-wide Process Services	PRODUCER	10
CTL	I/O controllers/ports	PRODUCER	5
CVC	SCS traffic	PRODUCER	19

May 16, 2005 page 18

Out-of-the-Box Processor Modules (4)			
Name	Description	Type	# Metrics
DEV	Disks: fixed data	PRODUCER	29
DLM	Distributed Lock Manager	PRODUCER	50
DSK	Disks: variable data	PRODUCER	23
DTM	Distributed Transaction Manager	PRODUCER	37
FCP	FCP	PRODUCER	34
GLX	Galaxy resources	PRODUCER	11

May 16, 2005 page 19

Out-of-the-Box Processor Modules (5)			
Name	Description	Type	# Metrics
INET	Networking software	PRODUCER	146
MEM	Memory	PRODUCER	26
NTI	Ethernet, NIC	PRODUCER	67
PAR	SYSGEN parameters	PRODUCER	350
PRO	Processes	PRODUCER	88
SRV	MSCP, TMSCP	PRODUCER	58

May 16, 2005 page 20

Out-of-the-Box Processor Modules (6)

Name	Description	Type	# Metrics
SYS	Faulting, process states, etc	PRODUCER	73
XFC	File System Cache	PRODUCER	44
XVC	Volume caching	PRODUCER	37

May 16, 2005

page 21

The TDC Application (1)

```

$ TDC COLLECT -      ! Collect data
_ $ /INTERVAL=60 -   ! Snapshot every 60 seconds
_ $ /START=09:30 -   ! Beginning at 9:30 AM
_ $ /END=13:30 -     ! End at 1:30 PM
_ $ PRO,DSK, -       ! Process and Disk metrics
_ $ DISK:[DIR]MYSHARE/LIB/INCL=MYSTUFF -

```

Or

```

$ TDC
TDC> COLLECT/INTERVAL=60/COUNT=60 -
TDC> /COLLECTION=%N.TDC$COLLECTION -
TDC> PRO,DSK

```

May 16, 2005

page 22

The TDC Application (2)

```

$ TDC EXTRACT – ! Extract data
_ $ /START=10:30 – ! Beginning at 10:30 AM
_ $ /END=11:30 – ! End at 11:30 AM
_ $ /CONVERT=PRO – ! Process metrics only
_ $ /FORMAT=LIST ! Output format
_ $ /OUTPUT=PRO.LIS ! Output file

```

Or

```

$ TDC
TDC> EXTRACT /OUTPUT=TDC.OUT /COUNT=10 –
TDC> /COLLECTION=%N.TDC$DAT –
TDC> /CONVERT=PRO,DSK /FORMAT=LIST

```

May 16, 2005

page 23

Sample Extract

= * @ # # @ * = Collection Listing Begins = * @ # # @ * =

===== Snapshot Starts =====

Snapshot [0]:
 Created: 29-MAR-2005 21:12:16.67
 Completed: 29-MAR-2005 21:12:16.76
 Record Set count: 13

***** Record Set Starts *****

Sequence of 20 records of type TDC_PDSC [ID: 6]
 Record set collection start time: 29-MAR-2005 21:12:16.67
 Record set collection end time: 29-MAR-2005 21:12:16.67
 Prefix data for record set header:

```

# # # Record Prefix Begins # # #
prefix.TDC_REC_WU_RecNumber: 0 (Record number within set)
prefix.TDC_REC_LU_SnapshotNumber: 0 (Snapshot number)
prefix.TDC_REC_T_Timestamp: 29-MAR-2005 21:12:16.67 (Timestamp, record created)
prefix.TDC_REC_WU_TypeC: 0 [0] (Client-specific record type)
prefix.TDC_REC_WU_RecSize: 64 (Record size, bytes)
prefix.TDC_REC_A_TypeA.TDC_RecModifiers: 0XC4 (Record modifiers)

```

May 16, 2005

page 24

Sample Extract

The following characteristics are set:

TDC_REC_V_persistentRecord (persistent data)
 TDC_REC_V_writeNeeded (record not written since update or read)
 TDC_REC_V_setRecord (record is a record set header)
 prefix.TDC_REC_LU_Attributes: 1 (internal attributes)

Record Prefix Ends

Data records follow...

@@@@@ TDC_PDSC Data Record Starts @@@@@

Record Prefix Begins

prefix.TDC_REC_WU_RecNumber: 1 (Record number within set)
 prefix.TDC_REC_LU_SnapshotNumber: 0 (Snapshot number)
 prefix.TDC_REC_T_Stamp: 29-MAR-2005 21:12:16.67 (Timestamp, record created)
 prefix.TDC_REC_WU_TypeC: 0 [0] (Client-specific record type)
 prefix.TDC_REC_WU_RecSize: 121 (Record size, bytes)
 prefix.TDC_REC_A_TypeA.TDC_RecModifiers: 0X44 (Record modifiers)

The following characteristics are set:

TDC_REC_V_persistentRecord (persistent data)
 TDC_REC_V_writeNeeded (record not written since up

May 16, 2005

page 25

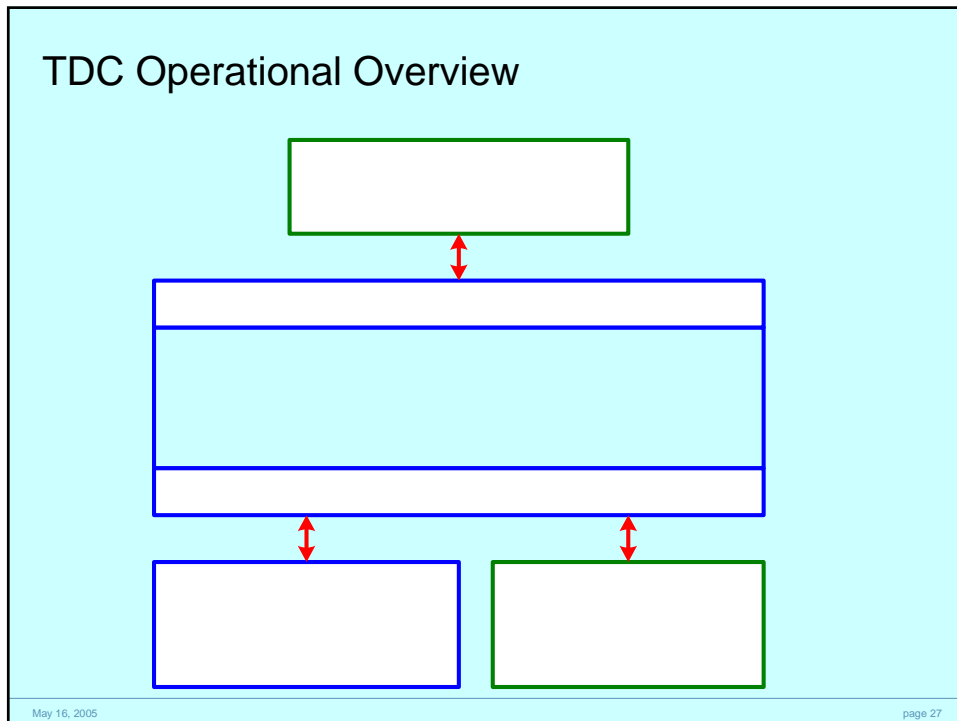
HELP

- Online help is available; type one of the following:
- **HELP OPERATIONS** - provides a brief overview of the operations available
- **HELP DATA** - provides a brief overview of data that can be collected
- **HELP PRIVILEGES** - lists the privileges required to collect data
- **HELP COLLECT** - describes the COLLECT command
- **HELP EXTRACT** - describes the EXTRACT command
- **HELP SHOW** - describes the SHOW command
- **HELP ANALYSIS** - provides information on data analysis and visualization tools

May 16, 2005

page 26

TDC Operational Overview



TDC Programming (Client Application)

```
#include "TDC_COMMON.H"

int main( void )
{
    // Collect "default" data every 5 minutes for 24 hours
    TDC_CTX_t    *ctx;
    TDC_Status_t status;

    ctx = malloc( sizeof *ctx );

    status = TDC_INIT( ctx );
    status = TDC_REGISTER( ctx, "DEFAULT", NULL, NULL );
    ctx->TDC_CTX_Operation = TDC_K_Collect;
    ctx->TDC_CTX_IntervalSize = 300;
    ctx->TDC_CTX_IntervalCount = 144;
    status = TDC_PREPARE( ctx );
    status = TDC_START( ctx );
    status = TDC_END( ctx );
    status = TDC_FINISH( ctx );
}
```

