

# Benchmark Report

## CONTENTS

<b>Preface</b>	<b>2</b>
Document History	2
<b>Chapter 1 / Introduction</b>	<b>2</b>
Executive Summary	2
Benchmark Summary	2
<b>Chapter 2 / Benchmark Setup</b>	<b>3</b>
Introduction	3
Benchmark configuration	3
Setup	4
Application Flow	5
Application configuration changes	6
– Placed small tables with scans to dedicated volumes	6
– Other optimizations	6
– Configuration of the Servlet engines in NSJSP	7
Running the Application	7
– Client systems running CentOS Linux	7
– Host system running NonStop OS release L18.02	7
<b>Chapter 3 / Benchmark results</b>	<b>8</b>
Results from Measure on the NonStop system	8
CPU Busy time	9
Transaction Information	10
Application Metrics	11
SQL Access Statistics	12
Conclusio	13

## Document History

### Version 1.0

This is the first version and includes the customer relevant parts of the HPE/ATC benchmark report from 19th Nov. 2018

## Chapter 1 / Introduction

---

### Executive Summary

The Payments Suite software from comforte has been ported to HPE NonStop servers. To confirm it will deliver good performance on this platform, a benchmark was executed on a dedicated system that was set up in the labs of the HPE Advanced Technology Center in Ft. Collins, USA. Prior to this benchmark, already some tests had been executed which resulted in recommendations to be implemented before running a large scale test.

The target of the benchmark was to show that the solution is able to execute/switch over 5,000 purchase/withdrawal (POS/ATM) transactions per second.

### Benchmark Summary

The application and database are configured to run on a fault-tolerant, scalable NonStop server. The configuration, i.e. 8 NonStop processors, with a total of 48 cores was capable of processing 5,086 purchase/withdrawal transactions per second with an average response time of less than 150 milliseconds. This time does not include time to authorize transactions remotely by issuing banks.

## Chapter 2 / Benchmark Setup

### Introduction

The Payments Suite software from comforte was benchmarked on an 8 processor 6-core NS7 X2 configuration that was configured in the HPE/ATC labs in Ft. Collins, USA. After initial tests on smaller configurations on-site at comforte AG in Germany, the purchase transaction was executed in increasing volumes until it exceeded the goal of 5,000 business transactions per second. The tests started early October 2018, and were executed until October 19.

### Benchmark configuration

The benchmark system configuration consists of the following components:

#### NonStop Server:

- o 8 processors with 6 cores each and 128GB per processor
- o Storage CLIMs: 4
- o Disk storage: 50 mirrored, 300GB SAS drives (HDD)
- o IP CLIMs: 2

#### Driver servers

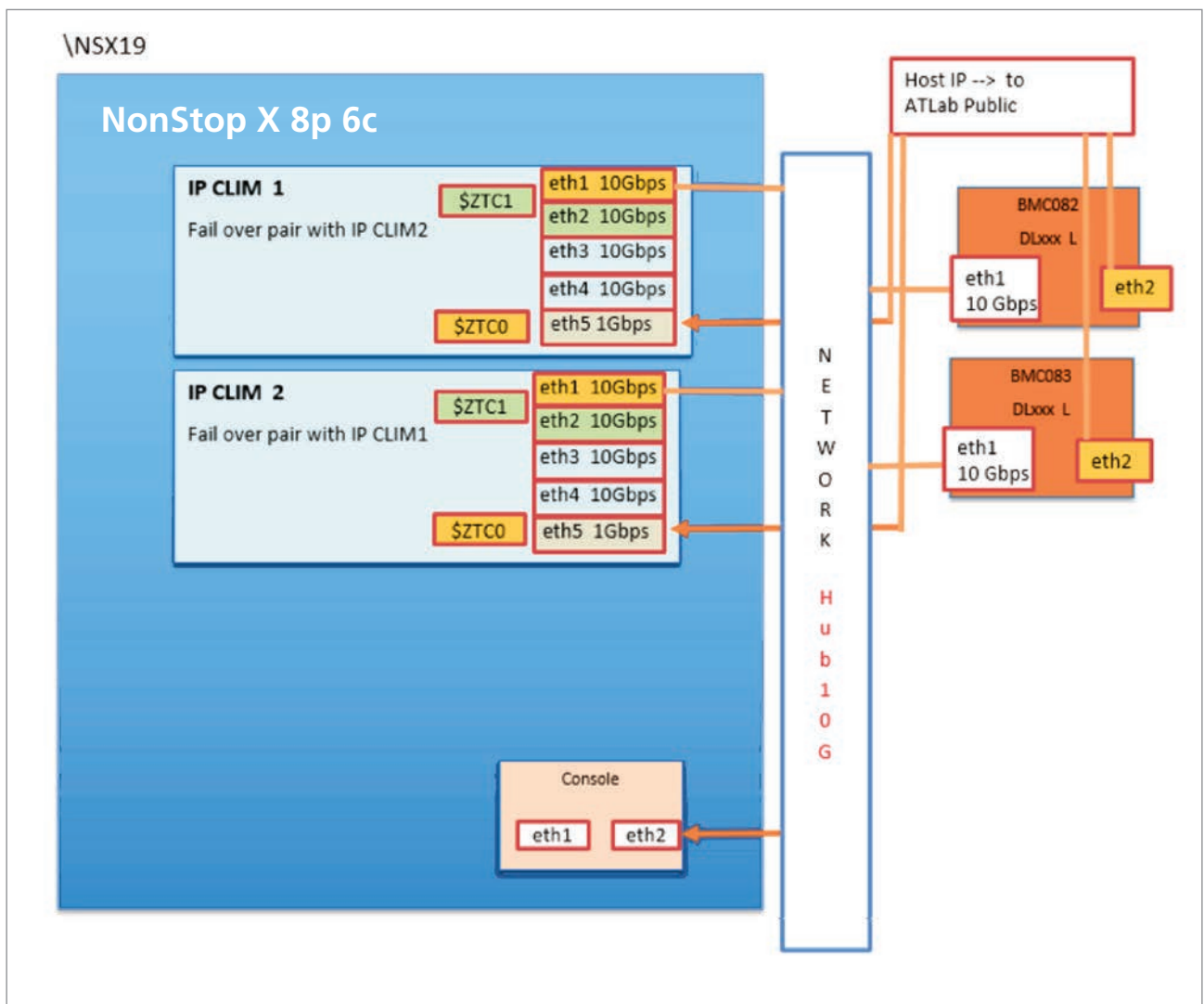
- o 2 DL360 Gen9, Xeon 2.6GHz 8 Core processors with 64 GB memory.
  - o CentOS 7.x operating system
- oconnected to the NonStop server via 10Gbit adapters.

#### Software

- o OS release: L18.02.00
- o Java T2866L80\_31JAN2018\_jdk180\_ADL\_05Jan2018
- o Java HotSpot(TM) 64-Bit Server VM (build "25.144-b01", mixed mode)
- o NSJSP T1222L71\_04MAY2017\_ABH\_V710
- o iTPWebserver T89777L01\_13OCT17\_AFO\_L104\_04
- o SQL.MX 3.6 T1051L36\_28FEB2018\_L36\_360\_0305
- o TMF was configured with two auxiliary audittrails, each handling half the before and after image data. WRITECACHE ENABLE (WCE) was turned ON for the audit trail disks. NB: WCE was enabled because the audittrail disks were normal HDD drives. Customers are typically using SSD drives and WCE allows close to the same write-speed for HDDs.
- o The data volumes were configured with 530 MB cache and 1GB of SQLMX Data Area (SDA).

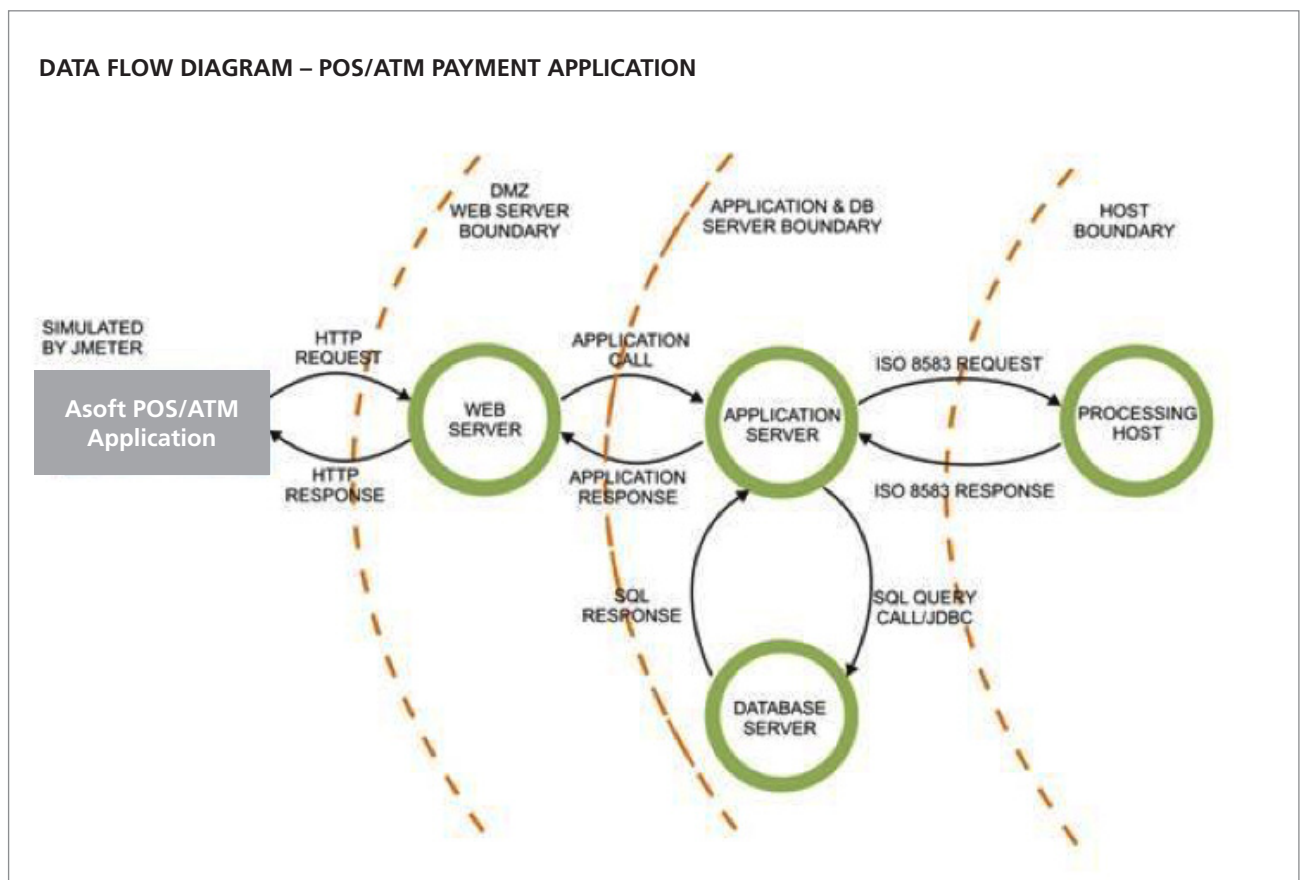
## Setup

The picture below shows the physical setup in the benchmark center.



## Application Flow

The application simulates a POS purchase or ATM transaction that comes in as an https request that is passed to the Java application. The Java application validates the request and simulates a short wait time of 10 ms (the ISO8583 request to the Processing Host (shown in the Application data flow diagram). It then finishes the transaction, logs the result and replies back to the POS or ATM device. A purchase Business Transaction comprises two TMF database transactions and accesses 12 SQL tables using 8 SQL statements, 6 SELECT statements, one UPDATE and one INSERT.



The NonStop system was accessed by one or two client systems running Apache JMeter software to drive the application. The system load was determined by the number of driver threads that were executing the transactions (simulating ATM/POS terminals) without user think-time. 600 client threads were needed to reach 5,086 business TPS with an average response time of 114 – 131 ms. Two client systems were required to reach this number, each simulating 300 terminals in 300 threads.

## Application configuration changes

The benchmark system was configured with more than enough storage to host the database, so initially, all tables and indexes were configured as hash-partitioned with 24 partitions, regardless of the amount of data that would be contained in these partitions. This is clearly the easiest way to create a database. However, during the initial test runs, we found that queries on some tables perform sub-set or full table scans and while the number of rows in the table is very small, accessing the 24 partitions returning no results creates an unacceptable overhead.

There were some other optimizations done as listed below. At this time, the CPU path-length for a single business transaction was around 5.8 milliseconds. As we will see in further tests, this increases with the load of the JVMs and the total system.

### Placed small tables with scans to dedicated volumes

The following four tables were placed on separate volumes.

TABLE_NAME	STATS_ROWCOUNT
BINIZDAVALAC	8
BINKARTICA	6
BINPROCESOR	3
BIN_TABLE_ISSUER	19

This configuration change resulted in a reduction of the CPU cost of 50 percent, because SQL/MX has to scan every partition, of which most are empty. Each of these tables was assigned to a dedicated volume, because of the target transaction rate: 5,000 TPS may result in double that amount of requests to a single disk process.

### Other optimizations

Other optimization tasks included:

- optimized access using views (e.g. This CQD, VALIDATE\_VIEWS\_AT\_OPEN\_TIME, was set to 'ON' in the SYSTEM\_DEFAULTS table)
- Removed logging from Java application (one application was logging internal data)
- Removed HTTPD logging (The application sends data to the application servers who handle all the necessary logging, the iTP Webserver was configured to only log errors in the error.log in the webserver/logs directory)

### Configuration of the Servlet engines in NSJSP

The application runs on-platform as a Java application in NonStop Java server Pages (NSJSP) which in itself is a server class within the iTP webserver environment. This environment is a Transaction Services / MP (TSMP) middleware that manages a cluster of httpd processes and JVM processes that run the application. Incoming requests are directed to these processes in two ways

1. The TCP/IP stack routes the requests to httpd processes that are all listening on the same port
2. The TSMP middleware routes the httpd requests to the JVMs. Instead of a single JVM that handles all requests, NSJSP uses a configured number of separate JVMs each capable of running the same application, where each JVM will process a share of the total load. The share of the total load is determined by the total number of processes and the level of concurrency in a single process. Our tests had shown that a single JVM can handle up to 20 concurrent requests without becoming too busy to cause internal queuing.

The Java application uses the JDBC Type-2 driver to access a common NonStop SQL/MX database.

### Running the Application

The test run that reached the 5,086 TPS was executed as follows.

#### Client systems running CentOS Linux

Two driver machines running JMeter clients, each simulating 300 terminals with 300 threads;

#### Host system running NonStop OS release L18.02

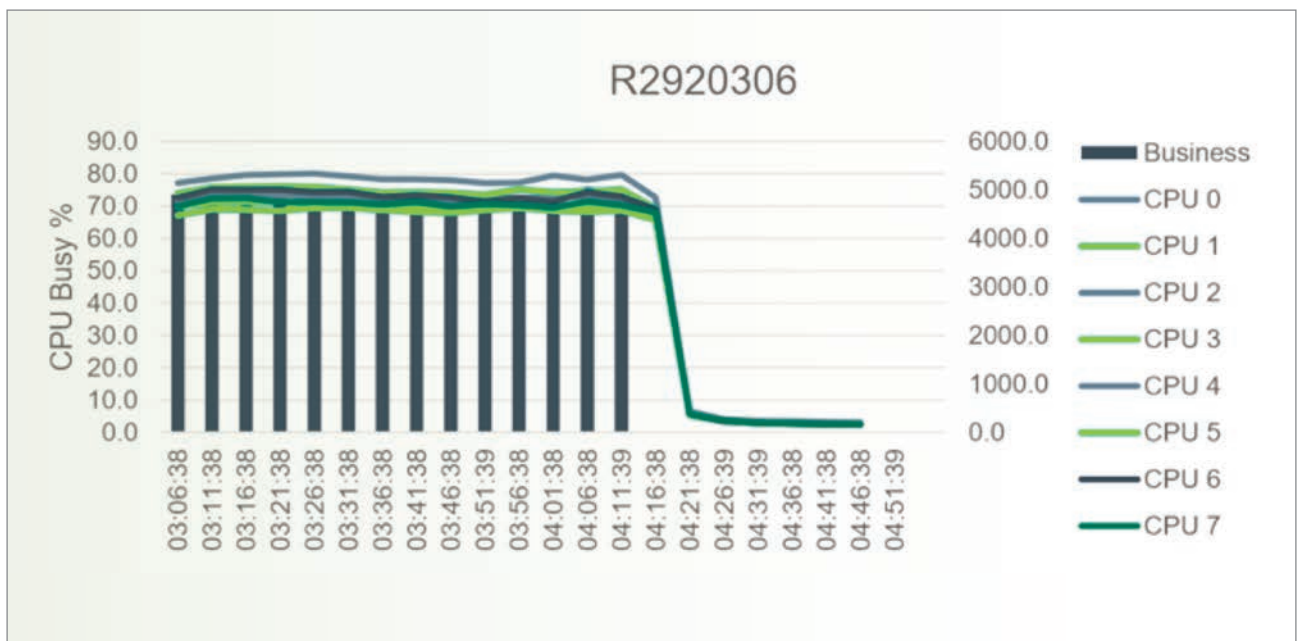
The definitions were:

- Default httpd setup; 40 processes, TANDEM\_RECEIVE\_DEPTH=50
- 40 JVMs (5 per logical processor) allowing a maximum of 24 concurrent requests (the value of the TANDEM\_RECEIVE\_DEPTH parameter);
- Value of LINKDEPTH = 3, value of MAXLINKS=24.

## Chapter 3 / Benchmark results

### Results from Measure on the NonStop system

Measure was run during this test on the host system1. The next chart shows the average CPU busy percentage for the 8 processors and the number of business transactions per second executed during the test period of one hour and 20 minutes. The distribution of transactions across the system is reasonably balanced with CPU busy rates between 79 and 80 percent. The transaction rate during this time varied from 4850 to 5071.



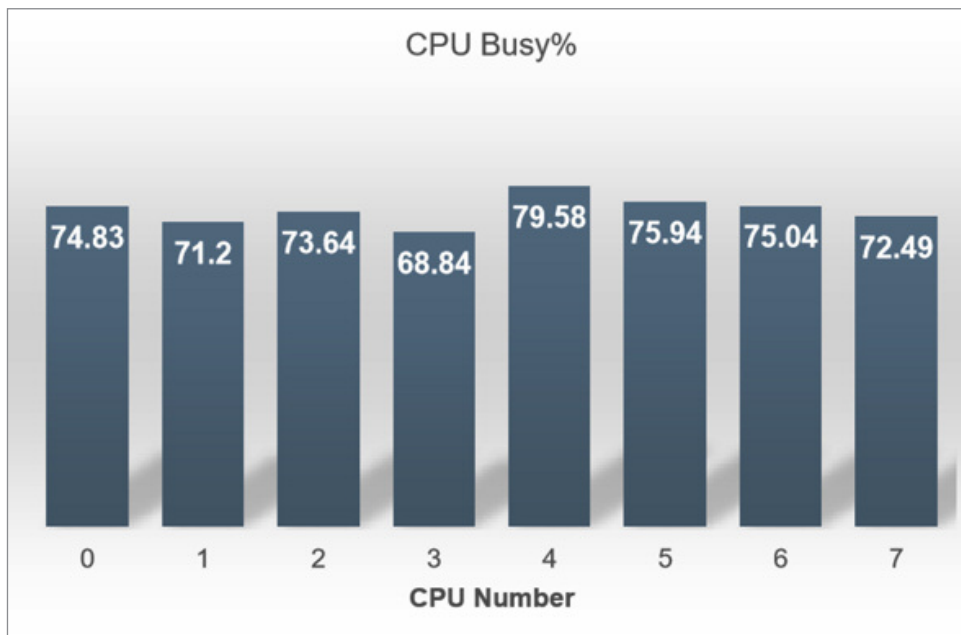


## CPU Busy time

In the interval of 5 minutes, the system with 8 processors was 590% busy (of 8\*100%, gives an average of 73.75%, which is a good busy percentage for response time and fault-tolerance.

CPU	Busy%	TMF	Info
<b>Cpu-Busy-Time</b>	<b>590.9</b>	<b>Home-Trans</b>	<b>10136. /s</b>
Ipu-Busy-Time	588.0	Home-Trans-Qtime	186.6. AQL
Ipu-Busy-Time	586.7	Home-Trans-ART	18.41 ms
Ipu-Busy-Time	593.7		
Ipu-Busy-Time	592.1		
Ipu-Busy-Time	590.3		
Ipu-Busy-Time	594.5		

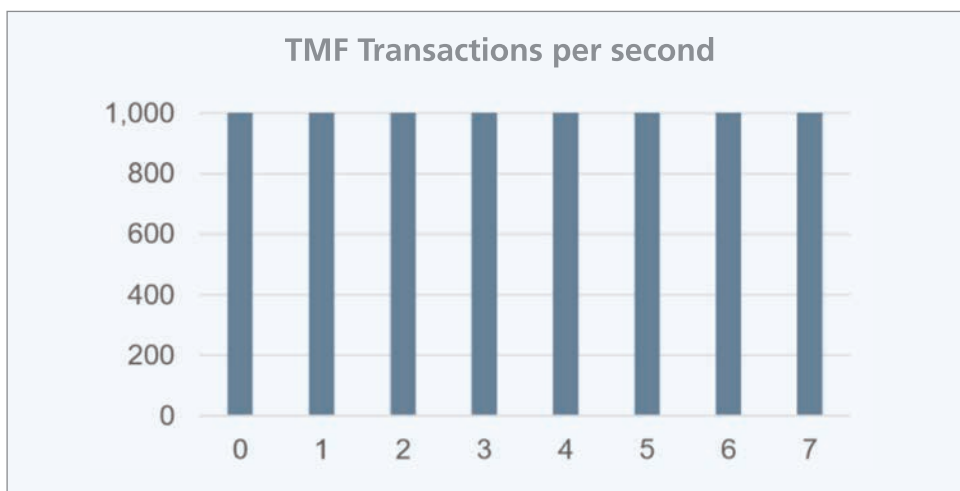
The IPU busy times are the busy times of all the cores (6 per processor).  
The next chart shows the distribution of the CPU times across the 8 processors.



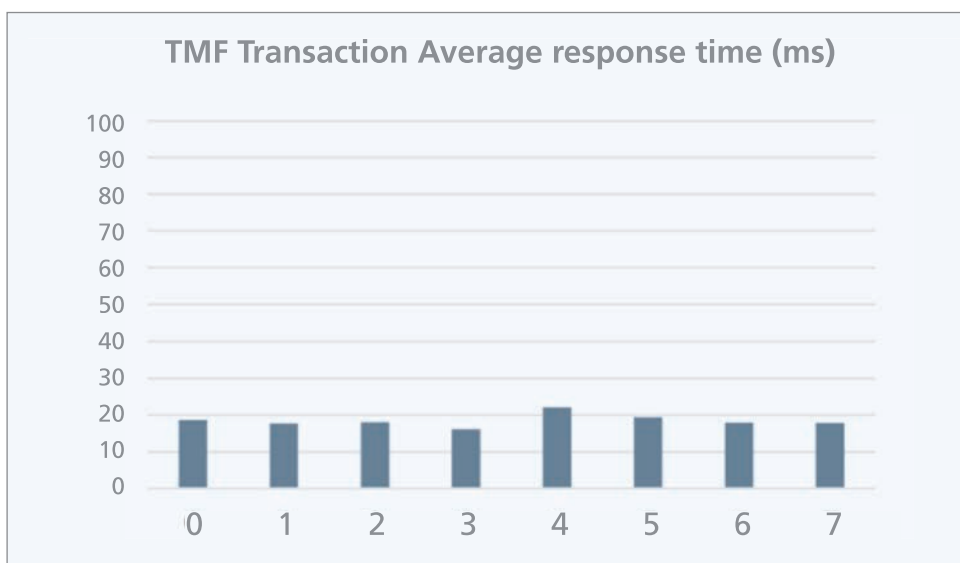
Processors 3 and 4 are the relative outliers, but in all, the system is well balanced.

## Transaction Information

The Measure subsystem records TMF transactions, not application business transactions. During the test in steady state, all transactions in the system are done by the application, except a few for system purposes. There are two TMF transactions for every business transaction. The following charts show the number of TMF transactions per logical processor as well as the average TMF transaction response time, which does not include any TCP/IP network time, but does include database IO time.



TMF TX rates during Measure interval



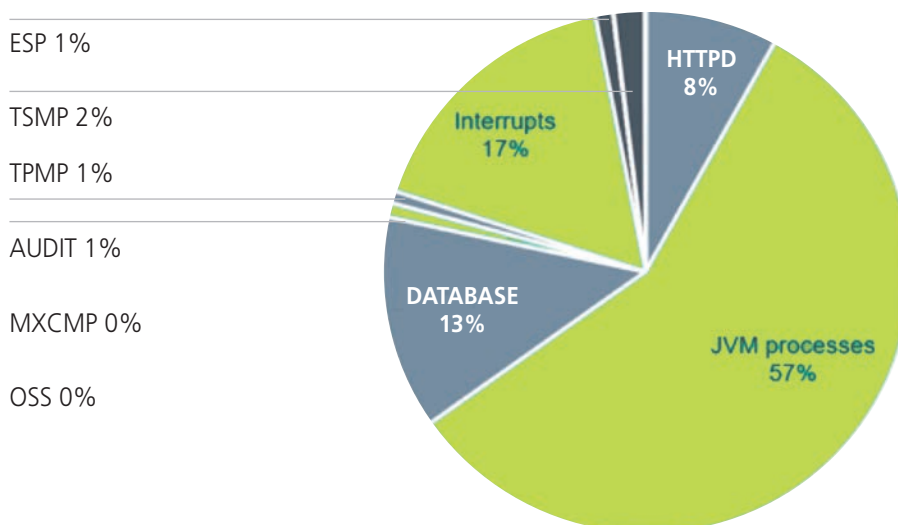
The response times in CPU three and four are related to the busy percentages of the processors.

## Application Metrics

The following information shows the breakdown of the system components (webserver, application server and database). It is not surprising that the JVMs that run the application take the lion share of the work. The table shows the total of CPU milliseconds spent by category of process. Note that due to the use of the JDBC T2 driver, a part of the database execution is included in the CPU cost of the JVM processes. Database cost is the total of the data disks (so not including the executor that runs inside the JVM processes).

PROCESS	CPU ms	ms per TMF tx
HTTPD	855,840	0.28
JVM processes	5,974,000	1.96
Database	1,374,336	0.45
AUDIT	89,333	0.03
TMP	81,310	0.03
Interrupts	1,771,800	0.58
ESPs	116,550	0.04
MXCMP	1,870	0.00
TSMP	203,532	0.07
OSS	3,542	0.00
Measure	9	-
ms per TMF transaction		<b>3.44</b>
ms per business transaction		<b>6.88</b>

## CPU cost distribution



## SQL Access Statistics

Measure also collects SQL access-related data. The next table shows the data that was collected in a similar run as the final test run that executed 4,100 TPS. These transactions were done in an interval of 30 seconds. In the final test run not all measure entities were included and measure ran out of workspace due to the length of the test. This output however is conform to the typical results that were recorded in all the test runs and is shown to discuss how the application accesses the database. This table shows how many database access are done to database tables and indexes. Each insert into the TRANSFILE, one per business transaction, also inserts four rows in the indexes of this table. The "Unaccounted" row indicates some additional accesses to other tables than the one listed. This row is included for a sanity check and is the difference between the sum of the named tables and the all the IO to this database (NSX19.\$\*.zsdmts01). The updates are from a few sequence generators that update the sequences for the transaction reference number. The table also shows the scans on the four tables that were placed on separate volumes, the BIN% tables.

	Rows per BTX	Reads	Writes	Updates	Rows Accessed	RowsUsed	Lck- Wait	Messages
Total TMF transactions	<b>249,606</b>							
TRANSFILE	1.00		124,911		124,911	124,911		124,961
IX_TERMID_STANDAT	1.00		124,877		124,877	124,877		124,925
IX_TFDAT	1.00		124,878		124,878	124,878		124,925
TERMID_STANORIGINAL_INDEX	1.00		124,878		124,878	124,878		124,925
XI_TERMDAT	1.00		124,877		124,877	124,877		124,925
BANKA	1.00	124,995			124,995	124,995		125,000
BIN_ISSUER	19.03	125,020			2,375,380	125,020		125,025
BINIZDAVALAC	8.01	999,560			999,560	999,560		124,948
BINKARTICA	12.01	249,897			1,499,382	249,897		249,907
BINPROCESOR	3.00	374,826			374,826	374,826		124,947
LOKACIJA	1.00	124,999			124,999	124,999		125,003
POS_WORK_MODE_PROFILE	1.00	124,994			124,994	124,994		124,998
TERMINAL	1.00	125,007			125,007	125,007		125,102
TERMINAL_STATE	2.00	124,980		124,975	249,955	249,955		250,053
TRGOVAC	1.00	125,002			125,002	125,002		125,004
\\NSX19.\$*.zsdmts01.*	54.09	2,500,774	624,421	124,989	6,750,029	3,250,184		2,125,672
Totals		2,499,280	624,421	124,975	6,748,521	3,248,676		2,124,648
Unaccounted		1,494	0	14	1,508	1,508		
Per TMF transaction		10.0	2.5	0.5	27.0	13.0		8.5
Per business transaction		20.03	5.00	1.00	54.07	26.03		17.02

## Conclusio

The purpose of this benchmark was to show that a NonStop server is able to process 5,000 purchase transactions per second using the comforte TMSSWITCH software. This volume has been reached multiple times and the data in this report was taken from one of these tests. During the preparation process that led to run this benchmark, several application improvements have been made which all contributed to this result.

The participants took some time during this benchmark to study several configuration options, which will help Comforte AG sizing future customer's systems.

This report was written by:

comforte AG  
Abraham-Lincoln-Str. 22  
65189 Wiesbaden  
Germany  
[www.comforte.com](http://www.comforte.com)

Copyright © 2018 comforte AG. All rights reserved.

*This document is the property of comforte AG and the information contained herein is confidential. This document, either in whole or in part, must not be reproduced or used for purposes other than that for which it has been supplied, without prior written permission or, if any part hereof is furnished by virtue of a contract with a third party, as expressly authorized under that contract.*

### *Trademark Acknowledgements*

*Hewlett Packard Enterprise, HPE, NonStop and other trademarks used by Hewlett Packard Enterprise ("HPE") are registered trademarks of Hewlett Packard Enterprise Development LP and/or its affiliates.*

*All other trademarks and registered trademarks are acknowledged and are the property of their respective companies.*