GTUG 2012, Dresden, Germany

Bob Rakoczy, Vice President, Architecture and Design

Bank of America

Bank of America

NonStop, Base24 and a Cloud Burst

A Proof-Of-Concept with

- Persistent Cloud Services

- Application Clustering Services

Bank of America

Overview

- This document is intended to describe the components and procedures used in the Persistent Cloud Services (PCS) Proof of Concept (POC) performed during March 2012.
- Participants in the POC included staff from HP's NonStop Services division as well as a company representatives from a NonStop user.
- The POC took place using hardware, software and networking provided by HP in their Advanced Technologies Center (ATC).

What is Persistent Cloud Services?

- An implementation of techniques and software which extends the inherent linear scalability of NonStop systems to a multi-platform environment consisting of NonStop servers as well as commodity Linux/Unix/Windows (LUW) servers.
- NonStop in the Cloud
- NonStop *as* the Cloud
- Cloud In The Box
- Converged Infrastructure
- Peak Capacity versus Elastic Metering

What is Persistent Cloud Services?

- The primary software components of PCS are:
  - TS/MP 2.4 (aka PATHWAY/PATHMON)
    - TS/MP (Transaction Services / Massively Parallel), aka PATHWAY, is a middleware product for the HP NonStop and provides transaction services, monitoring and management.
  - Application Clustering Services (ACS)
    - ACS works in conjunction with PATHWAY-TS/MP to control communications between requestor and server processes
  - Gateway Server
    - C program provided by HP Consulting Services which runs under PATHWAY to enable PCS functionality
  - Persistent Cloud Services (PCS) library routines
    - Java jar (and C/C++ and COBOL library files) provided by HP Consulting used by applications to interface with the Gateway server.
  - Java application program
    - User written application code to process requests/transactions.

What is Persistent Cloud Services? – ACS

- Application Cluster Services (ACS) is a feature within TS/MP (PATHWAY) that allows the user to define/create a relationship between multiple PATHWAY environments to allow logical grouping.
- In the POC, this was used to define a PATHWAY SERVER named RAKTEST in three (3) separate Pathway environments ($RPMN, $RAK1, $RAKLX) as a locally executing process (in $RPMN), as a remote process on another NonStop (in $RAK1) and as a remote process on a Linux system (in $RAKLX).
- The PATHWAY version tested in the POC, TS/MP 2.4, permits the ACS domain to span a single HP NonStop system. In TS/MP 2.5, the ACS domain management will extend to multiple NonStop systems, which will allow a more direct relationship between relationship for message routing and load balancing.

What is Persistent Cloud Services? – PCS

- Persistent Cloud Services (PCS) is an architecture designed to enable the NonStop to be a controller of, and a participant with, a group of commodity servers or Cloud deployments. As such, it allows NonStop to be an important component of a Converged Infrastructure environment.

- In general terms, PCS enables the extension of Pathway functionality to commodity servers or virtual servers running in public or private clouds.

What is Persistent Cloud Services? – PCS Lib

- The Americas NonStop Solutions Engineering Group (ANSEG) has developed a handful of libraries that help support this extended capability (patents pending) .
- These Persistent Cloud Services (PCS) libraries provide the intersystem communications and message protocol for the PCS components.
- The libraries are provided for the following languages:
  - Java (used in this POC)
  - C
  - COBOL (to allow easy enhancement of legacy applications)

Proof of Concept

Bank of America

Proof Of Concept – Goals and design

- Goals
  - To determine the viability of Persistent Cloud Services in the Client's HP NonStop environment.
  - Specifically test PCS with ACI's Base24 application (Base24 is used for ATM and POS applications).
- Design
  - Java was selected as the programming language to ensure portability as the intention was to write/compile the program in the NonStop environment and distribute the "executable" to the other systems, including a LINUX server.
  - SQL/MX on the HP NonStop was selected as the database to highlight its use as well as spotlight its capabilities.
  - Base24 and PWR are products common to the HP NonStop sites.

Performance

Bank of America

Performance

- The POC system was built with more of an eye toward functionality versus high speed performance.
- That said, load balancing is an important feature of TS/MP therefore it was monitored to determine if the distribution was performed as expected as well as which aspect(s) of the system had the highest utilization.
- One aspect of the configurations for the POC was to establish an environment where the workload was distributed between the three systems. The initial workload was serviced by processes on the HP NonStop system \A. As that system became busy, the workload was further distributed to HP NonStop \B. And, finally, all work in excess of the throttled levels for the HP NonStop was pushed to a Linux system named BMC.
- The function, known as Cloud Burst, worked as described.

Performance

- HP NonStop system \A was the location of the processes requesting/initiating the transaction, the XPNET process responsible for routing/queuing , and the SQL/MX database and as such had high messaging activity (as defined by the interrupt process TSMSGIP in the HP NonStop OS), hence the slightly elevated response time when compared to the HP NonStop system, \B, and the Linux BMC system.
- The chart on the following slide represents a 20 second interval of stable traffic flow in the middle of the test and shows the load was distributed across the three available systems with each HP NonStop system running approximately 200 Transactions Per Second (TPS) and the extra workload pushed off to the Linux BMC at 550+ TPS.

Performance

System	Dest	PPD	TPS	Response Time
\A	JRAKA	$GWR0	50.70	0.015
\A	JRAKA	$GWR1	41.50	0.017
\A	JRAKA	$GWR2	41.60	0.017
\A	JRAKA	$GWR3	48.80	0.015
\B	JRAK1	$GWA0	47.30	0.010
\B	JRAK1	$GWA1	57.00	0.010
\B	JRAK1	$GWA2	46.60	0.014
\B	JRAK1	$GWA3	65.10	0.008
BMC	JRAKL	$GWA0L	83.30	0.008
BMC	JRAKL	$GWA1L	78.50	0.009
BMC	JRAKL	$GWA2L	68.70	0.010
BMC	JRAKL	$GWA3L	73.20	0.009
BMC	JRAKL	$GWB0L	65.90	0.010
BMC	JRAKL	$GWB1L	65.40	0.010
BMC	JRAKL	$GWB2L	67.90	0.011
BMC	JRAKL	$G2B3L	63.60	0.011

Performance

- The four (4) CPUs on the primary HP NonStop system, \A, operated at an average of 42% with two processors running at 34% and two processors running at or near 52%. The difference is attributed to the location of the disk processes where the Java applications were alternately retrieving data from the SQL data or logging informational events.
- Additional CPU resources were used by \A because this system acted as the focal point for incoming transactions as well as database activity and messaging (XPNET).
- The four (4) CPUS on the second HP NonStop system, \B, operated at a much more modest 10% average. As system \B only ran the Java application, its activity was expected to be noticeably lower that \A.
- Performance data was not available for the Linux system, BMC.

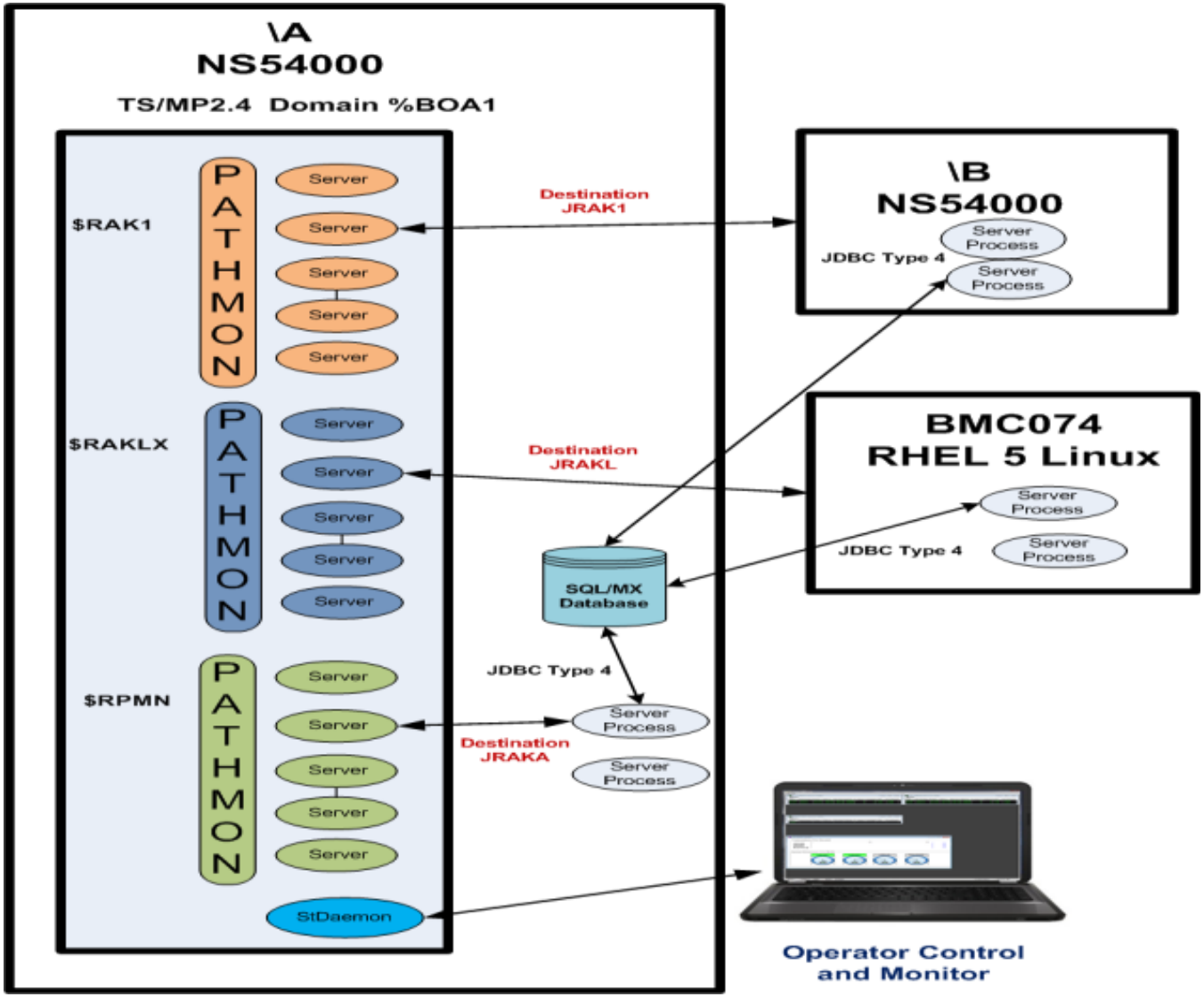
Configuration

Bank of America

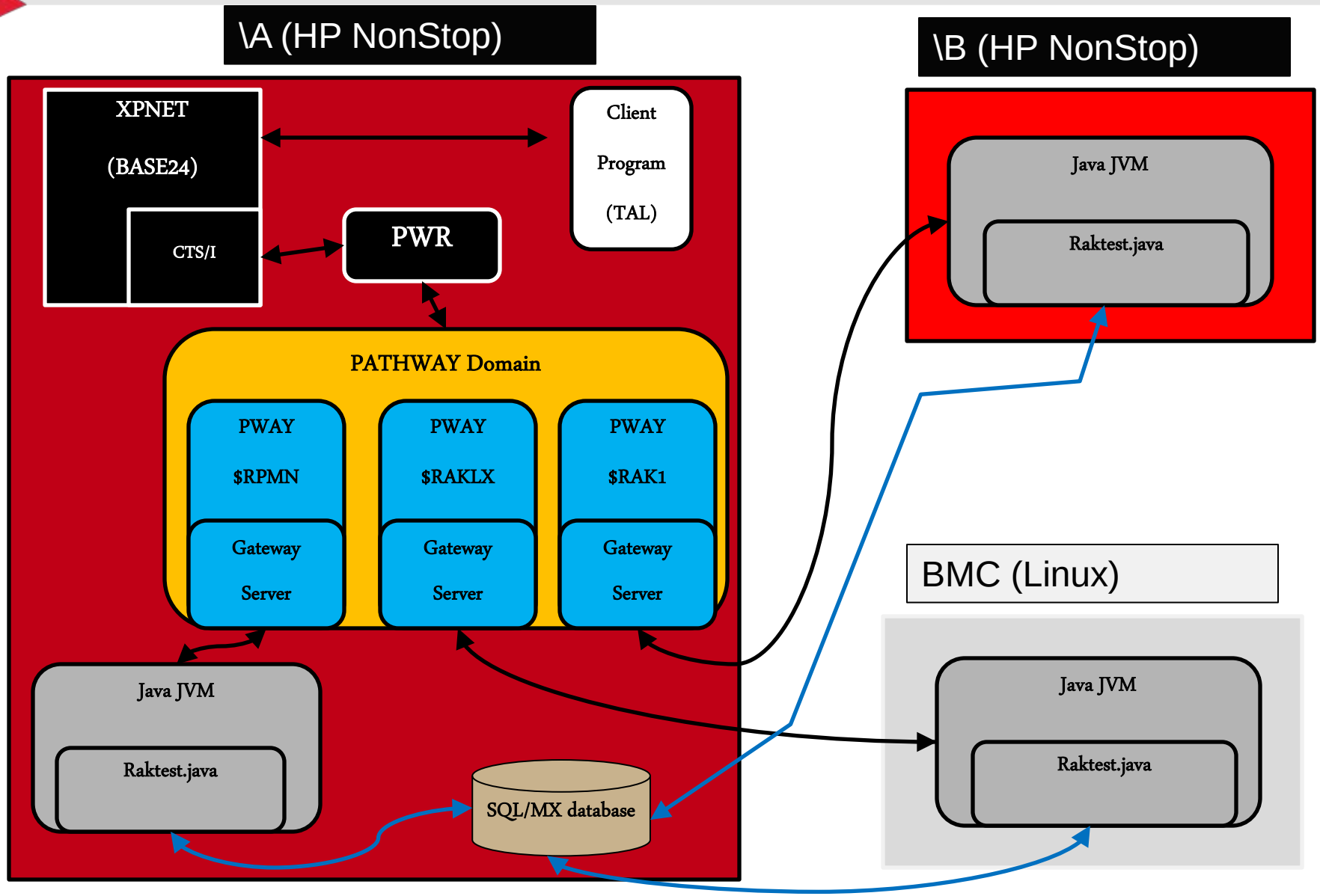
Configuration – Hardware/Software

- HP provided the hardware and software for the testing at their Advanced Technology Center (ATC).
- The hardware consisted of two (2) HP NonStop quad-core systems as well as a Linux server.
- The communications between the three systems was via a private LAN within HP's ATLAB data center.
- The HP NonStop systems were model NB54000c quad-core with the current version of the Guardian O/S, J06.13, and Pathway subsystem, TS/MP 2.4.
- The Linux system was an HP rack-mount server with an Intel Xeon processor running Red Hat Enterprise Linux (RHEL) version 5.

Configuration – graphical view



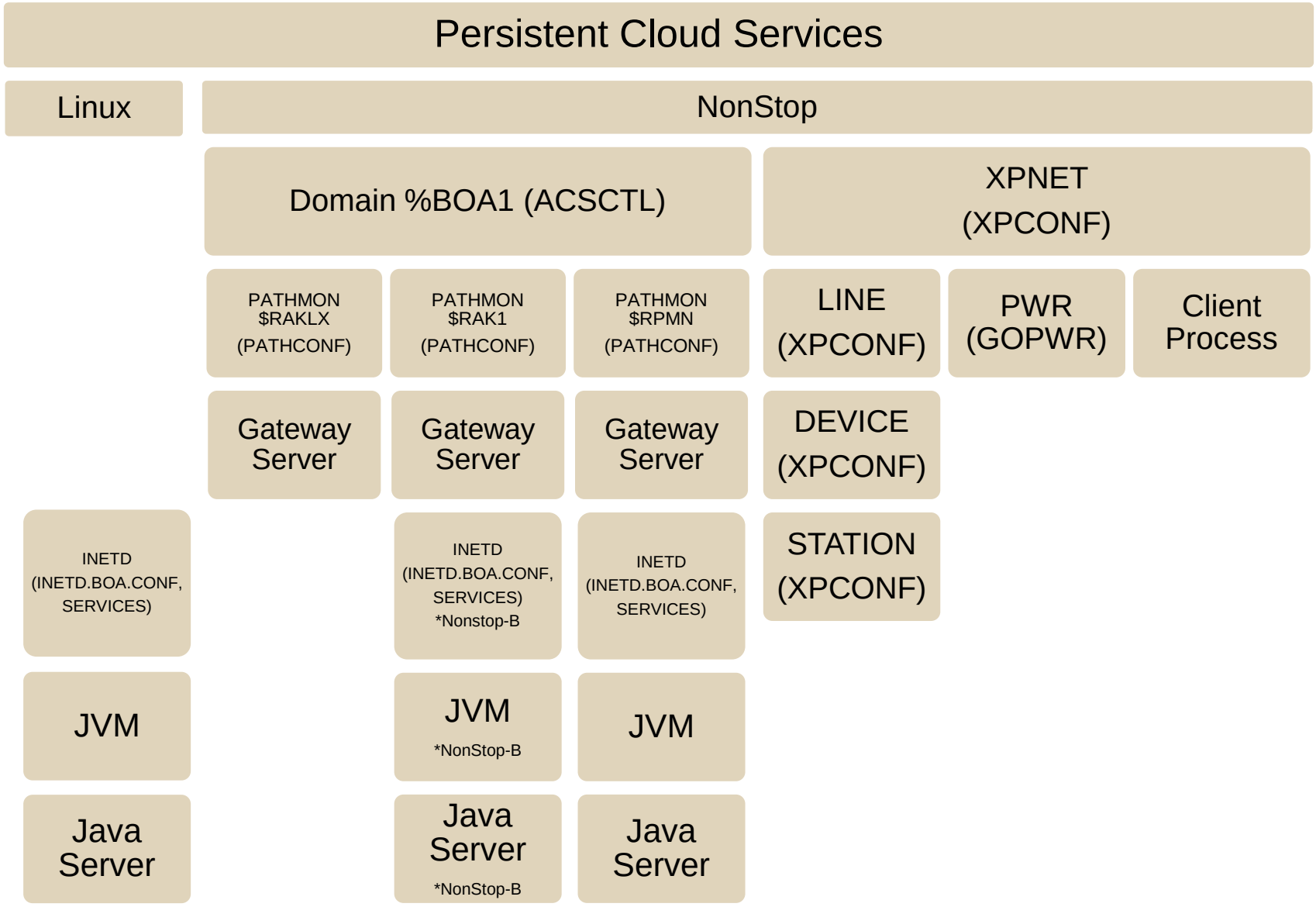
Configuration – graphical view



Message Flow

1. TAL test program initiates a message via XPNET
2. XPNET routes the message to PWR
3. PWR issues PATHSEND to the domain %BOA1 which encapsulates $RPMN, $RAK1, $RAKLX
4. Message is forwarded based upon workload to available PATHWAY SERVER (Gateway Server).
5. Java Server (RAKTEST) receives message using PCS library and performs SQL SELECT using message data (PAN – card number).
6. Java Server RAKTEST invokes JDBC library to retrieve record/row from SQL DB on system \A using PAN. SQL DB has two columns, PAN and SCORE.
7. RAKTEST returns response (PAN and SCORE) via PCS Library to PWR via PATHWAY
8. PWR sends response via XPNET to TAL test program

Configuration Hierarchy



Configuration – Step 1a

Configure ACS via SCF (Subsystem Control Facility) on the NonStop

The ACSCTL file defines the DOMAIN and its component PATHWAY systems as well as a form of user permissions.

```
$SYSTEM ZACS 17> EDIT $SYSTEM.ZACS.ACSCTL
TEXT EDITOR - T9601H01 - (01MAY05)
CURRENT FILE IS $SYSTEM.ZACS.ACSCTL
*la
   1    [ACS Domain]
   2    # Name must match the name given in ADD ACS command
   3    # And the system name must match to the current system.
   4    A = \A
   5    [Pathway Domain]
   6    # Multiple PATHMONs on the same system
   7    %BOA1 = \A.$RPMN:33, \A.$RAK1:33, \A.$RAKLX
   8    [Owner Membership]
   9    *.* = %BOA1
*e
$SYSTEM ZACS 18>
```

Configuration – Step 1b

Implement the ACS configuration

```
$SYSTEM ZACS 18> SCF CONTROL ACS $ZACS, ACSCTL $SYSTEM.ZACS.ACSCTL

SCF - T9082H01 - (23JUN11) (02MAY11) - 03/22/2012 06:42:23 System \A
(C) 1986 Tandem (C) 2006 Hewlett Packard Development Company, L.P.

Total Errors = 0    Total Warnings = 0
$SYSTEM ZACS 19>
```

Configuration – Step 2a

Configure the RAKTEST server process in a PATHWAY

Three PATHWAY environments were configured for the POC: $RPMN, $RAK1 and $RAKLX, where $RPMN, in this case, is the “first among equals” in that its servers will be used first to process the workload.

The following commands show a configured RAKTEST server in the secondary PATHWAY environment $RAK1 although all the RAKTEST server configurations in all PATHMONs are essentially identical.

Design note – The Java servers running in the PATHWAY environment were coded to use the PCS libraries to exchange messages to ensure the same code was implemented on all environments. If the servers had been designed to run solely on the HP NonStop environment, the program would have implemented interprocess communication using $RECEIVE and the Java JToolkit JAR.

Configuration – Step 2b

Configure the RAKTEST server process in a PATHWAY

```
$SYSTEM ZACS 22> PATHCOM $RAK1;INFO SERVER RAKTEST
SERVER RAKTEST
  PROCESSTYPE OSS
  ARGLIST TEST
  AUTORESTART 1
  CPUS (2:3,3:2,1:0,0:1)
  CREATEDELAY 0 SECS
  CWD /home/BOA/robertr
  DEBUG OFF
  DELETEDELAY 1 MINS
  ENV MAXBUF=8000
  ENV PROJECT=BOA
  ENV SERVERCLASS=GATEWAYSRV
  ENV DESTINATION=JRAK1
  ENV AGENT=$RK1AG
  ENV LOGFILE=/usr/local/pcs/log/RAK1
  ENV CONFIGDIR=/usr/local/pcs/etc
  ENV TIMEOUT=30
  ENV DEBUGLVL=1
  ENV TRACELVL=0
  HIGHPIN ON
…
```

Configuration – Step 2c

Configure the RAKTEST server process in a PATHWAY (cont.)

```
  HOMETERM \*.$RVHS
  LINKDEPTH 1
  MAXLINKS 1
  MAXSERVERS 8
  NUMSTATIC 8
  OWNER \A.204,1
  PRI 50
  PROCESS $GWA0 (CPUS 0:1)
  PROCESS $GWA1 (CPUS 1:0)
  PROCESS $GWA2 (CPUS 2:3)
  PROCESS $GWA3 (CPUS 3:2)
  PROCESS $GWB0 (CPUS 0:1)
  PROCESS $GWB1 (CPUS 1:0)
  PROCESS $GWB2 (CPUS 2:3)
  PROCESS $GWB3 (CPUS 3:2)
  PROGRAM /usr/local/pcs/bin/GwServer.exe
  SECURITY "N"
  STDERR /usr/local/pcs/log/RAK1.stderr
  TIMEOUT 99 SECS
  TMF OFF
  VOLUME \A.$DATA01.RAK1PATH
$SYSTEM ZACS 23>
```

Configuration – Step 3a

Configure the PCS Gateway Server

The GATEWAY server provides a conduit from PATHWAY to application server processes and implements the PCS Library for message exchange. It is located in the OSS space. The location defaults to the install directory but can be overridden using CONFIGDIR in the RAKTEST server configuration in PATHWAY.

```
 BOA.ROBERTR /usr/local/pcs/etc : cat JRAKA.conf
#
#  PCS Destination configuration
#
HOST      nsa
TCP       $ZTC0
PORT      9999
BUFSIZE   9000
TIMEOUT   8
TRACELVL  0
DEBUGLVL  0
BOA.ROBERTR /usr/local/pcs/etc : exit
exit
$DATA01 RAKPATH 10>
```

Configuration – Step 4a

Configure Internet connectivity

INETD is the Internet Daemon. It works in conjunction with the SERVICES configuration file to associate TCP/IP connection requests on specific ports to specific programs. Examples of this include the FTP client to FTP server program and Telnet client to Telnet service. The default INETD configuration file is INETD.CONF and is located in the OSS space on the HP NonStop systems. The SERVICES file is located in the Guardian space on the HP NonStop systems but is aliased via a “link” to provide exposure also via the OSS space.

In the following example, port 9999 is defined in SERVICES as being associated with JRAKSRV which is configured within INETD.BOA.CONF and therefore will invoke the JAVA program defined in that file.

Configuration – Step 4b

Configure Internet connectivity (cont.)

Abridged SERVICES file.

```
/home/BOA/robertr : cd /etc
/etc : cat services
#
# @(#)services 1.16 90/01/03 SMI
#
# Network services, Internet style
# This file is never consulted when the NIS are running
#
echo            7/tcp
ftp             21/tcp
telnet          23/tcp
#
# Host specific functions
#
stdaemon        18500/tcp                   # PCS stats daemon
jraksrv         9999/tcp                    # BOA POC java server
/etc :
```

Configuration – Step 4c

Configure Internet connectivity (cont.)

Abridged INETD.BOA.CONF file (see also inetd.conf).

```
/etc : cat inetd.BOA.conf
#
# Configuration file for inetd(8).  See inetd.conf(5).
# To re-configure the running inetd process, edit this file, then send
# send the inetd process a SIGHUP.
#
# Internet services syntax:
#  <service_name> <socket_type> <proto> <flags> <user> <server_pathname> <args>
#
echo    stream  tcp     nowait  super.super  internal
#
stdaemon stream tcp nowait BOA.robertr  /usr/local/pcs/bin/stdaemon.exe stdaemon
.exe /usr/local/pcs/etc/stdaemon/

jraksrv stream tcp nowait BOA.robertr /usr/tandem/java/bin/java java -classpath
/home/BOA/robertr/dmc:/home/BOA/robertr/dmc/lib/t4sqlmx.jar:/home/BOA/robertr/dm
c/jtestserver/:/home/BOA/robertr/dmc/common:/home/BOA/robertr/dmc/lib -Djava.lib
rary.path=/home/BOA/robertr/dmc/lib -Dt4sqlmx.properties=/home/BOA/robertr/dmc/j
testserver/t4sqlmx.properties JTestSrv -L /home/BOA/robertr/dmc/testlogs/ -v

/etc :
```

Configuration – Step 5a

ACI Configuration (XPNET, Base24)

The ACI product, XPNET, was installed on the ATC test system \A. XPNET is the core of the BASE24 application and provides message routing and queuing functions for the BASE24 applications such as ATM and POS processing.

The following LINE and STATION entities were replicated 16 times on the test system to provide sufficient throughput capability (Only a single DEVICE entity was required as it provides a template for the STATION)

Configuration – Step 5b

ACI Configuration (XPNET, Base24) (cont.)

$TECH1 RAKREMS 21> NCPCOM $RPMN
NCPCOM Version REL3^VER1^NCPC19^20081010
(c) Copyright 1995 Applied Communications, Inc.
1 > PATH $RPMN
2 > set more off
3 > info device *pwr*,obeyform

RESET DEVICE
SET DEVICE INPUTFORMAT TXT
SET DEVICE OUTPUTFORMAT TXT
SET DEVICE POLLFORMAT EOT P1 P2
SET DEVICE RECVLEN 32000
SET DEVICE SELECTFORMAT EOT S1 S2
SET DEVICE TYPE 20
SET DEVICE XMITLEN 32000
ADD DEVICE D1A^PWR^DEV^01, UNDER SYSNAME \A, UNDER NODE P1A^NODE

Configuration – Step 5c

ACI Configuration (XPNET, Base24) (cont.)

…

```
RESET LINE
SET LINE DUPLEX HALFLCL
SET LINE PORT \A.$RAKP
SET LINE FMM STATUS
SET LINE PROTOCOL CTS
ADD LINE L1A^PWR^01, UNDER SYSNAME \A UNDER NODE P1A^NODE
5 > info station *pwr*, obeyform

RESET STATION
SET STATION DESTINATION P1A^PWR^01
SET STATION LADDR %BOA1
SET STATION RADDR RAKTEST
SET STATION CLASS PWR
SET STATION DEVICE D1A^PWR^DEV^01
SET STATION LINENAME \A.P1A^NODE.L1A^PWR^01
SET STATION FMMBROADCAST OFF
ADD STATION S1A^PWR^01, UNDER SYSNAME \A, UNDER NODE P1A^NODE
6 > exit
$TECH1 RAKREMS 22>
```

Configuration – Step 6a

Configure the PWR ACI component

PWR is used to provide a communications pipeline from the ACI XPNET process to PATHWAY servers (i.e. it permits XPNET to act as a PATHWAY requester and communicate with PATHWAY servers).

```
 $TECH RAK1CONF 27> edit gopwr r;la;e
TEXT EDITOR - T9601H01 - (01MAY05)
CURRENT FILE IS $TECH.RAK1CONF.GOPWR
    1     comment **
    2     comment **  This is an example of an obey file to run PWR.
    3     comment **  You need to change the parameters, PPD name, and the
    4     comment **  EMS collector name to your needs.
    5     comment **  TACL parameters and run-line parameters are not both
    6     comment **  necessary.  When both are present, run-line params will
    7     comment **  override.
    8     comment **  To run with no backup, remove TACL 'PARAM' and 'RUN'
    9     comment **  values for the backup CPU.
   10      comment **
   11
…
```

Configuration – Step 6b

Configure the PWR ACI component (cont.)

```
  12    clear all
  13    param maximum-threads 2048
  14    comment      param backup-cpu 0
  15    comment ** Timeout in seconds:
  16    param timeout 600
  17    param buffsize 32000
  18    param tmf-protect off
  19
  20    run $D2D008.XPNET.pwr /name $rakp , nowait, TERM $RVHS, cpu 2/&
  21    threads 64, &
  22    timeout 600, &
  23    buff 4096, &
  24    ems $R0, alt-ems $0
$TECH RAK1CONF 28>
```

PDMCOM – Pathway Domain Manager

Using PDMCOM to manage multiple PATHWAYs

```
$SYSTEM ZACS 6> PDMCOM
PDMI - T0845H09 - (01FEB2012) - 20 March 2012, 09:21:32
(C)2008-2012 Hewlett Packard Development Company, L.P.
PDMI 1>> open %BOA1
 PATHMON : \A.$RAKLX
 PATHMON : \A.$RPMN
 PATHMON : \A.$RAK1
 PDMI 2>>
```

PDMCOM – Pathway Domain Manager

Using PDMCOM to control multiple PATHWAYs

```
PDMI 2>> STATUS SERVER *

PATHMON : \A.$RAKLX
SERVER           #RUNNING   ERROR  INFO
AGENT                1
RAKTEST              2

PATHMON : \A.$RPMN
SERVER           #RUNNING   ERROR  INFO
AGENT                1
RAKTEST              8
SERVER-NCP           1
SERVER-NCPI-1A       1
SERVER-NCPI-1B       1

PATHMON : \A.$RAK1
SERVER           #RUNNING   ERROR  INFO
AGENT                1
RAKTEST              8

PDMI 3>>
```

Persistent Cloud Services

Question and Answer (maybe)

Bank of America