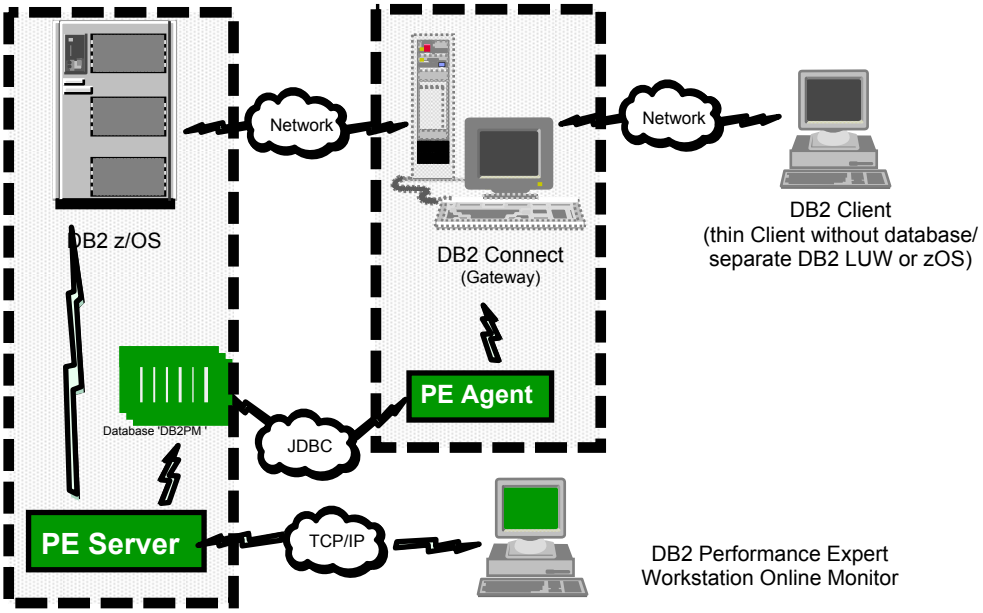
IBM DB2 Performance Expert for z/OS and Multiplatforms V2 Whitepaper



Gary Crupi
DB2 Technical Sales
gary.crupi@us.ibm.com
Milwaukee, WI

January 4, 2005

Introduction

DB2 Connect gateway monitoring, AKA e2e monitoring, is a new feature of DB2 Performance Monitor V8 for z/OS and DB2 Performance Expert Version 2 for z/OS and Multiplatforms (DB2 PE V2). This paper assumes you have licensed DB2 PE V2 for z/OS. If you only license DB2 PE V2 for Multiplatforms, the information you receive for e2e monitoring will be limited to a single screen – the DB2 Connect gateway statistics (page 10).

This paper is a collection of key information from various sources listed in the Resources section and my own observations. The goal is to make the reader aware of the benefits and resources available.

Components

There are 3 pieces of code that need to be installed in order to utilize e2e monitoring. The Performance Expert Client (workstation GUI), the Performance Expert Agent (on the DB2 Connect gateway) and the Performance Expert Server (on the DB2 for z/OS server) all work together to give a single view of the connection from the application to the DB2 for z/OS server.

The Performance Expert Agent (PE Agent) collects DCS snapshot information and sample SQL statement timings and sends it to the DB2 PE database on the host. The HISTORYDATA keyword must be updated with DB2CAPPLIC and DB2CSYSTEM on the DB2 PE Server. This enables the synchronization of performance data and history collection. The PE Agent is controlled by the e2e command interface. When starting the PE Agent on Unix or Linux, run the e2e –start command via “nohup ./e2e –start &” to avoid the process being killed when the administrative user logs off.

Data is viewed via:

1) DB2 Connect gateways
2) DB2 Thread information
3) DB2 Statistics information

The next section is based on a DB2 Connect Gateway db2inst1 on ccgtch50.ibmus2.ibm.com port 50000 in Chicago and DSNB on demomvs.demopkg.ibm.com port 6561 in Dallas. Each field on the following screens is described in detail in the product online help. In addition to a brief explanation, I have pasted the applicable help from the product below each screen shot.

Installation Tips

The Performance Expert Agent must be installed on each DB2 Connect workstation. The code is located in host MVS library SFPEWS01.

- FPEKAGNT for Windows
- FPEKAGAX on AIX
- FPEKAGHP on HP-UX
- FPEKAGLX on Linux
- FPEKAGL3 on zLinux
- FPEKAGSL on Solaris

Sample FTP session

```
/tmp> ftp boepmo1 Connected to boepmo1.boeblingen.de.ibm.com.
220-FTPD1 IBM FTP CS V1R2 at BOEPMO1.boeblingen.de.ibm.com, 08:53:44 on 2003-06-24
220 Connection will close if idle for more than 60 minutes.
User (boepmo1.boeblingen.de.ibm.com:(none)): wtn
331 Send password please.
Password:
230 WTN is logged on. Working directory is "WTN.".
ftp> cd..
250 "" is the working directory name prefix.
ftp> cd sys1.fpe.v210.sfpews01
250 The working directory "SYS1.FPE.V210.SFPEWS01" is a partioned data set
ftp> bin
200 Representation type is Image
ftp> get FPEKAGAX db2peage.install-on-aix
200 Port request OK.
125 Sending data set SYS1.FPE.V210.SFPEWS01(FPEKAGNT)
250 Transfer completed successfully. ftp: 51194200 bytes received in 32.08Seconds 1595.98Kbytes/sec.
ftp> quit
221 Quit command received. Goodbye.
```

Next, run the installation program.

On Windows

- Run **db2peagent.install-on-win.exe** (which is the downloaded FPEKAGENT) and follow instruction
- After installation you will be asked to configure the PE agent setup. You can do it now or later using e2e.exe command

On AIX

Installation via GUI: **<mount-point>/db2peage.install-on-aix**

Or

In console mode: **<mount-point>/db2peage.install-on-aix \is:javaconsole –console**

where db2peage.install.on.aix is the downloaded FPEKAGAX member.

Configuration

- Log on as root by using the command su –
- Change to the directory /opt/IBM/db2peage/V2.1/bin
- ./db2peage-config
- Type the name of the DB2 instance and Select **Add a host to your instance**

The e2e command is used to register and customize the agent.

Change directory to "/opt/IBM/db2peage/V2.1/bin" (AIX) or to "C:\Program Files\IBM\DB2 Performance Expert Agent\bin" (Windows). For example, on Windows:

```
>e2e --help
E2E agent initialization. Please refer to log file messages. Current operating system has been set to
'WINDOWS'.
JNI library 'E:\Program Files\IBM\DB2 Performance Expert Agent\bin\e2elib.dll' was successfully loaded.
Command line parameters:
--start[log]                  : Start collection process, [log snapshot buffer];
--help                               : Displays this help;
--level                              : Displays PE agent's release info;
--addhost                     : Registers a remote host to process;
                                       [host name/IP address]  [port number]  [DB name]  [DCS DB Y/N]
[login]   [password];
--listhosts                   : Displays a list of registered remote hosts;
--removehost[num]      : Removes and deregisters the host;
--uncatalog                   : Uncatalogs all E2E-registered DB2 entries;
--change                      : Changes user login/password for E2E host;    [num] [login] [password]
--update                      : Updates new release of the agent from the host;
--rollback                    : Restores previous release of the agent;
--test[num]                   : Tests connection to the registered remote host;

   User directory default folder : "..\instances\DB2\"
   Version of E2E agent/stream    : 3/3
   New version file name          : E:\Program Files\IBM\DB2 Performance Expert Agent\bin\newver\e2e.jar
E2E agent has been stopped.
```

The e2e command can be used after the initial setup. For example, on Windows:

```
E:\Program Files\IBM\DB2 Performance Expert Agent V2\bin>e2e --listhosts
IBM(c) Performance Expert Agent.
Current operating system is 'WINDOWS'.
The library 'E:\Program Files\IBM\DB2 Performance Expert Agent V2\bin\e2elib.dll' successfully loaded.
Servers registered in '..\instances\DB2\e2e.ini' file :
-------------------------------------------------------------------------------
Num) Host name                      Port  Database            Alias    M-Frame      UserLogin  Paused
-------------------------------------------------------------------------------
  1) 127.0.0.1                      50000 DB2PMLOC            D8708810 No           jen        No
  2) 9.152.87.192                   5090  SYSDSNI             D8770289 Yes          jen1       No
  3) 9.152.87.191                   5140  SYSDB2E             D8826650 Yes          jen1       No
  4) 9.152.87.222                   5721  PMO2D721            D0569923 Yes          jen1       No
-------------------------------------------------------------------------------
Registered DB2 entries :
-------------------------------------------------------------------------------
     Host name                      Port  Database  Alias     Node     M-Frame
-------------------------------------------------------------------------------
     9.152.87.222                   5721  D0569923  D0569923  N0569922 Yes
     9.152.87.192                   5090  D8770289  D8770289  N5690111 Yes
     9.152.87.191                   5140  D8826650  D8826650  N5691393 Yes
```

The e2e.ini file contains the information as well.

On Windows: C:\Program Files\IBM\DB2 Performance Expert Agent\instances\<DB2>\e2e.ini
On AIX: /var/db2pe/<DB2 instance>/e2e.ini

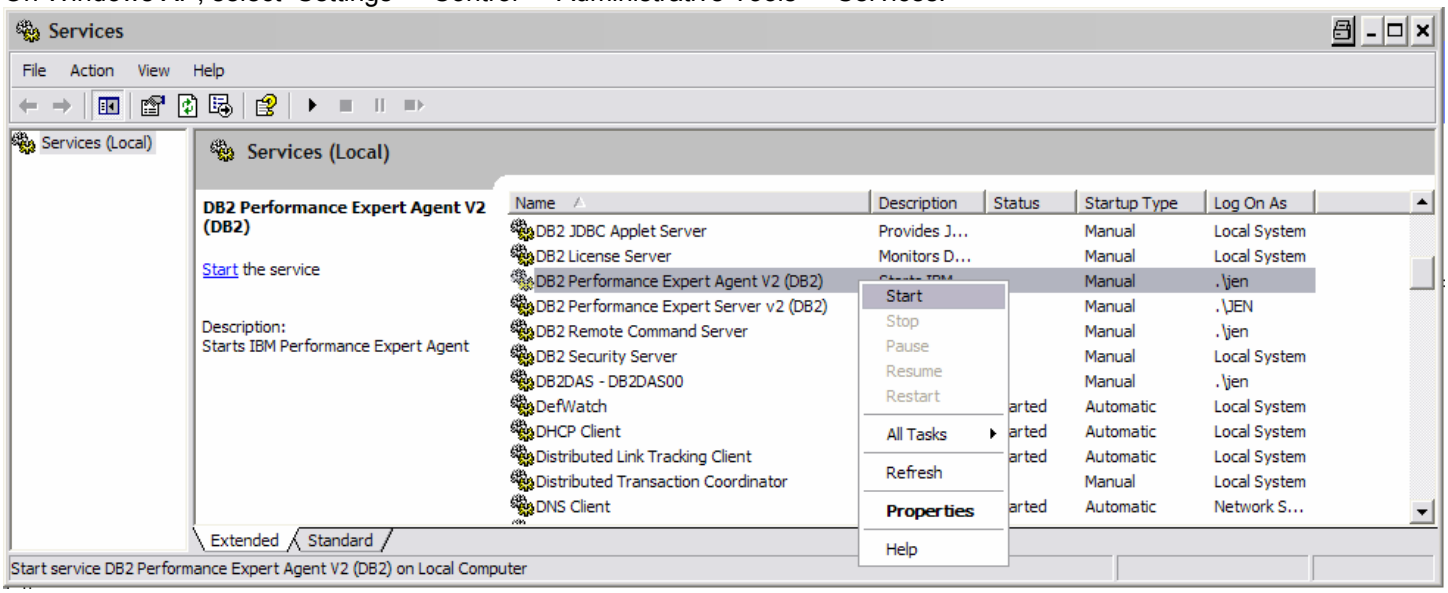
```
127.0.0.1    50000 D8708810 DB2PMLOC N ki1w liIq5UgYYOg= A
9.152.87.192 5090  D8770289 SYSDSNI  Y ki1w7Q== liIq5UgYYOg= A
9.152.87.191 5140  D8826650 SYSDB2E  Y ki1w7Q== liIq5UgYYOg= A
9.152.87.222 5721  D0569923 PMO2D721 Y ki1w7Q== liIq5UgYYOg= A
```

Before the PE agent is started, the PE Server parameters must be set up.

1. Activate the internal DB2PM DB2 tables to receive the PE agent data
   PERFORMANCEWAREHOUSE= YES (Abbr. PWH=YES)
2. Granting privileges
   To grant those privileges for the UIDs defined in E2E.INI, you can use the sample member *hlqdb2pe*.SFPEINS0(FPEZPDCA).
3. Setting of history parameter
   HISTORYDATA=(…,DB2CAPPLIC n,DB2CSYSTEM m) (Abbr. …,DB2CA,DB2CS) where n and m are multipliers to the standard history interval

Next Start the PE agent. Stopping the agent is a similar procedure.

On Windows XP, select Settings -> Control -> Administrative Tools -> Services:



On AIX,:

Change directory to: /opt/IBM/db2peage/V2.1/bin.
nohup ./e2e –start & respectively **./e2e --stop**

Messages are written to the e2elog.

```
Mon Dec 20 11:46:21 CET 2004 - Current agent version : 12
Mon Dec 20 11:46:21 CET 2004 - Current result data stream version : 3
Mon Dec 20 11:46:21 CET 2004 - Last compatible data stream version : 3
Mon Dec 20 11:46:21 CET 2004 - Level : java/com/ibm/db2pm/e2e/E2EVersion.java, jE2E, N210_FP2HOT, E073
Mon Dec 20 11:46:21 CET 2004 - Executing command : --start
Mon Dec 20 11:46:21 CET 2004 - CheckActive : lock [E:\Program Files\IBM\DB2 Performance Expert Agent
V2\instances\DB2\e2e.lock] does not exist.
Mon Dec 20 11:46:21 CET 2004 - Executing getHostList...
Mon Dec 20 11:46:21 CET 2004 - Executing getDatabaseAlias…
….
Mon Dec 20 11:46:21 CET 2004 - --------------------------------------------------------------------------
Mon Dec 20 11:46:21 CET 2004 - Tracing the content of e2e.ini file...
Mon Dec 20 11:46:21 CET 2004 - 1) 127.0.0.1 50000 D8708810 DB2PMLOC N ki1w liIq5UgYYOg= A
Mon Dec 20 11:46:21 CET 2004 - 2) 9.152.87.192 5090 D8770289 SYSDSNI Y ki1w7Q== liIq5UgYYOg= A
Mon Dec 20 11:46:21 CET 2004 - 3) 9.152.87.191 5140 D8826650 SYSDB2E Y ki1w7Q== liIq5UgYYOg= A
Mon Dec 20 11:46:21 CET 2004 - 4) 9.152.87.222 5721 D0569923 PMO2D721 Y ki1w7Q== liIq5UgYYOg= A
Mon Dec 20 11:46:21 CET 2004 - Refreshing host list from 'e2e.ini' file.
Mon Dec 20 11:46:21 CET 2004 - Servers from 'e2e.ini' file successfully retrieved.
Mon Dec 20 11:46:21 CET 2004 - [4] server(s) accepted.
Mon Dec 20 11:46:21 CET 2004 - Executing getHostList...
Mon Dec 20 11:46:21 CET 2004 - Executing getDatabaseAlias...
```

e2elog continued…

```
Mon Dec 20 11:46:22 CET 2004 - Testing connection to [jdbc:db2:D8770289; 9.152.87.192]...
Mon Dec 20 11:46:23 CET 2004 - Cannot establish connection to 'jdbc:db2:D8770289' :
COM.ibm.db2.jdbc.DB2Exception: [IBM][CLI Driver] SQL30081N  A communication error has been detected.
Communication protocol being used: "TCP/IP".  Communication API being used: "SOCKETS".  Location where the
error was detected: "".  Communication function detecting the error: "connect".  Protocol specific error
code(s): "10061", "*", "*".  SQLSTATE=08001
Mon Dec 20 11:46:23 CET 2004 - Testing connection to [jdbc:db2:D8826650; 9.152.87.191]...
Mon Dec 20 11:46:24 CET 2004 - Host OS from 'jdbc:db2:D8826650' successfully retrieved as 'z/OS'. Mainframe
type.
Mon Dec 20 11:46:24 CET 2004 - Checking status, 'D8826650' host...
Mon Dec 20 11:46:25 CET 2004 - This gateway is not registered with '192.168.2.100_B99FF913_DB2' on
'D8826650'alias. Registering...
Mon Dec 20 11:46:25 CET 2004 - Generating agent ID for [9.152.87.191] host...
Mon Dec 20 11:46:26 CET 2004 - Updating current timezone value with '3600'...
Mon Dec 20 11:46:26 CET 2004 - Agent ID for '9.152.87.191' generated/maintained as '8'.
Mon Dec 20 11:46:26 CET 2004 - IP address of network resource '9.152.87.191' retrieved as 9.152.87.191
Mon Dec 20 11:46:26 CET 2004 - <!> Server found '9.152.87.191'.
Mon Dec 20 11:46:22 CET 2004 - Testing connection to [jdbc:db2:D8770289; 9.152.87.222]...
….
Mon Dec 20 11:46:31 CET 2004 - ------------------------------------------------------------------------
Mon Dec 20 11:46:31 CET 2004 - Getting settings for each configured server...
Mon Dec 20 11:46:31 CET 2004 - ----- [9.152.87.191; D8826650] server -----
Mon Dec 20 11:46:31 CET 2004 - Timezone difference value [3600] found on the '9.152.87.191' host from
'VERSION.V_VALUE'.
Mon Dec 20 11:46:32 CET 2004 - History collection flag with value [Y] found on the host '9.152.87.191'.
Mon Dec 20 11:46:32 CET 2004 - History collection interval with value [15] found on the host '9.152.87.191'.
Mon Dec 20 11:46:32 CET 2004 - System collection multiplier with value [1] and system collection flag with
value [Y] found on the host '9.152.87.191'.
Mon Dec 20 11:46:32 CET 2004 - Application collection multiplier with value [1] and application collection
flag with value [Y] found on the host '9.152.87.191'.
Mon Dec 20 11:46:32 CET 2004 - Automatic agent update flag with value [N] found on the host '9.152.87.191'.
Mon Dec 20 11:46:33 CET 2004 - ----- [9.152.87.222; D0569923] server -----
Mon Dec 20 11:46:33 CET 2004 - Timezone difference value [3600] found on the '9.152.87.222' host from
'VERSION.V_VALUE'.
```

The PE agent then checks the setup parameters every 10 minutes.

```
Sat Dec 18 22:05:23 CET 2004 - Getting settings for each configured server...
Sat Dec 18 22:05:23 CET 2004 - ----- [9.152.87.191; D8826650] server -----
Sat Dec 18 22:05:23 CET 2004 - Snapshot successfuly retrieved.
Sat Dec 18 22:05:23 CET 2004 - Calculating 'Package statistics'...
Sat Dec 18 22:05:23 CET 2004 - Chains= null; STINBYTESRC counter= null; Result AVGRESPSZIN= null;
STOUTBYTESRC= null; Result AVGRESPSZOUT= null; STINBYTESSD= null; Result AVGREQSZIN= null; STOUTBYTESSD= null;
Result AVGREQSZOUT= null; HOSTRESPNCE= null; STMTEXECTIME= null; Result NETWORKTIME= null; Result AVGNETTIME=
null; TRANSMISSIONS_GROUP= null; Result STMTGROUP= null;
Sat Dec 18 22:05:23 CET 2004 - Collecting 'Task list' counters...
Sat Dec 18 22:05:23 CET 2004 - Windows detected.
Sat Dec 18 22:05:23 CET 2004 - Timezone difference value [3600] found on the '9.152.87.191' host from
'VERSION.V_VALUE'.
Sat Dec 18 22:05:23 CET 2004 - History collection flag with value [Y] found on the host '9.152.87.191'.
Sat Dec 18 22:05:23 CET 2004 - History collection interval with value [15] found on the host '9.152.87.191'.
Sat Dec 18 22:05:24 CET 2004 - System collection multiplier with value [1] and system collection flag with
value [Y] found on the host '9.152.87.191'.
Sat Dec 18 22:05:24 CET 2004 - [20] 'Task list' counters successfully collected.
Sat Dec 18 22:05:24 CET 2004 - System data/error codes retrieved.
Sat Dec 18 22:05:24 CET 2004 - Processing result stream...
Sat Dec 18 22:05:24 CET 2004 - Result telegram 'DCS_DB_Transmissions' with 1008 ID skipped. No data collected.
Sat Dec 18 22:05:24 CET 2004 - Result telegram 'RemoteDatabases' with 1006 ID skipped. No data collected.
Sat Dec 18 22:05:24 CET 2004 - Inserting data stream for '9.152.87.191' host...
Sat Dec 18 22:05:24 CET 2004 - Application collection multiplier with value [1] and application collection
flag with value [Y] found on the host '9.152.87.191'.
Sat Dec 18 22:05:24 CET 2004 - Automatic agent update flag with value [N] found on the host '9.152.87.191'.
Sat Dec 18 22:05:24 CET 2004 - ----- [9.152.87.222; D0569923] server -----
Sat Dec 18 22:05:25 CET 2004 - Timezone difference value [3600] found on the '9.152.87.222' host from
'VERSION.V_VALUE'.
Sat Dec 18 22:05:26 CET 2004 - History collection flag with value [Y] found on the host '9.152.87.222'.
Sat Dec 18 22:05:26 CET 2004 - System data/error codes successfully stored for '9.152.87.191' host.
Sat Dec 18 22:05:26 CET 2004 - System collection time for [9.152.87.191; D8826650] is [Fri Dec 17 21:40:41 CET
2004].
```

The PE agent also checks for a new version after startup and then every 10 minutes.

```
Sat Dec 18 21:29:48 CET 2004 - ------------------------------------------------------------------------------
Sat Dec 18 21:29:48 CET 2004 - Executing automatic version check...
Sat Dec 18 21:29:48 CET 2004 - New versions not found.
Sat Dec 18 21:29:48 CET 2004 - Version control execution time is [Sat Dec 18 21:29:38 CET 2004].
Sat Dec 18 21:29:48 CET 2004 - ------------------------------------------------------------------------------
```

Some fairly common error situations include:

```
Sat Dec 18 21:24:53 CET 2004 - Inserting data stream for '9.152.87.222' host...
Sat Dec 18 21:24:54 CET 2004 - <!> ERROR: Error inserting system data for the host '9.152.87.222' :
COM.ibm.db2.jdbc.DB2Exception: [IBM][CLI Driver][DB2] SQL0803N  One or more values in the INSERT statement,
UPDATE statement, or foreign key update caused by a DELETE statement are not valid because the primary key,
unique constraint or unique index identified by "" constrains table "" from having duplicate rows for those
columns.  SQLSTATE=23505
Sat Dec 18 21:24:54 CET 2004 - The server is stopped or collection interval changed. Waiting for the next
collection interval...
Sat Dec 18 21:24:54 CET 2004 - <!> ERROR: Error storing system information for '9.152.87.222' host.
Sat Dec 18 21:24:54 CET 2004 - Error(s) found in System information collection for host '9.152.87.222'.
Waiting for the next collection.
```

PE Agent tries to add new data into the DB2 tables but PE server does not read and store them into the snapshot history dataset.

PE server not running

DB2 PE server not correctly setup

PWH = YES not set

Privileges not granted

DB2CAPPLIC and DB2CSYSTEM (resp.DB2CA and DB2CS) not defined for parameter HISTORYDATA

Control interval was changed between two agents communications => Agent is waiting for next interval

No action is needed.

A successful execution may look like this:

```
Sat Dec 18 21:24:58 CET 2004 - Inserting data stream for '9.152.87.191' host...
Sat Dec 18 21:25:00 CET 2004 - System data/error codes successfully stored for '9.152.87.191' host.
Sat Dec 18 21:25:00 CET 2004 - System collection time for [9.152.87.191; D8826650] is [Fri Dec 17 20:49:11 CET
2004].
```

When fixpaks are applied to the PE agent files, the agent must be updated. This can be done 1) manually to test a new version of the agent or 2) automatically to apply the updates to all agents once the new version has been tested.

Manual

- Download and install the PTF / Fixpak
- Stop PE agent via Service panel
- Run “e2e --update” command
- Re-start via Service panel
- Test it
- Potentially recover (Stop service, “e2e --rollback”, restart service)

Automatic

- Set PE server startup parameter in the parameter file to AUTOMATICAGENTUPDATE=YES and reset it after automatic updates to ‘NO’

Or

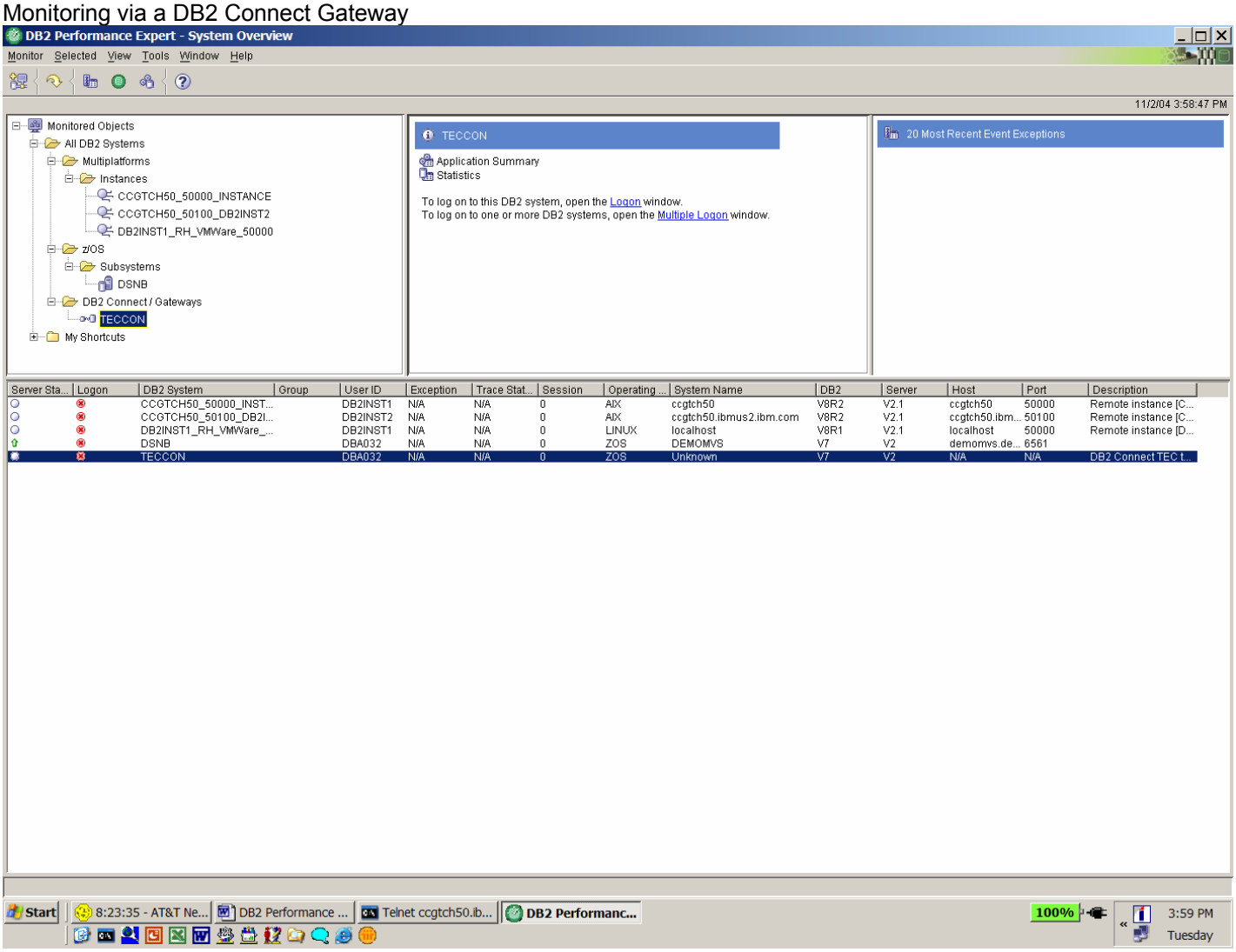
- Use MVS command
  /f xxxxxxxx,AUTOMATICAGENTUPDATE=YES whereas xxxxxxx is the PE server started task job or
  /f xxxxxxxx,AAU=YES
  After all PE agents are updated you may reset AAU=NO again.

E2ELOG.DB2 will show the new version after successful update:

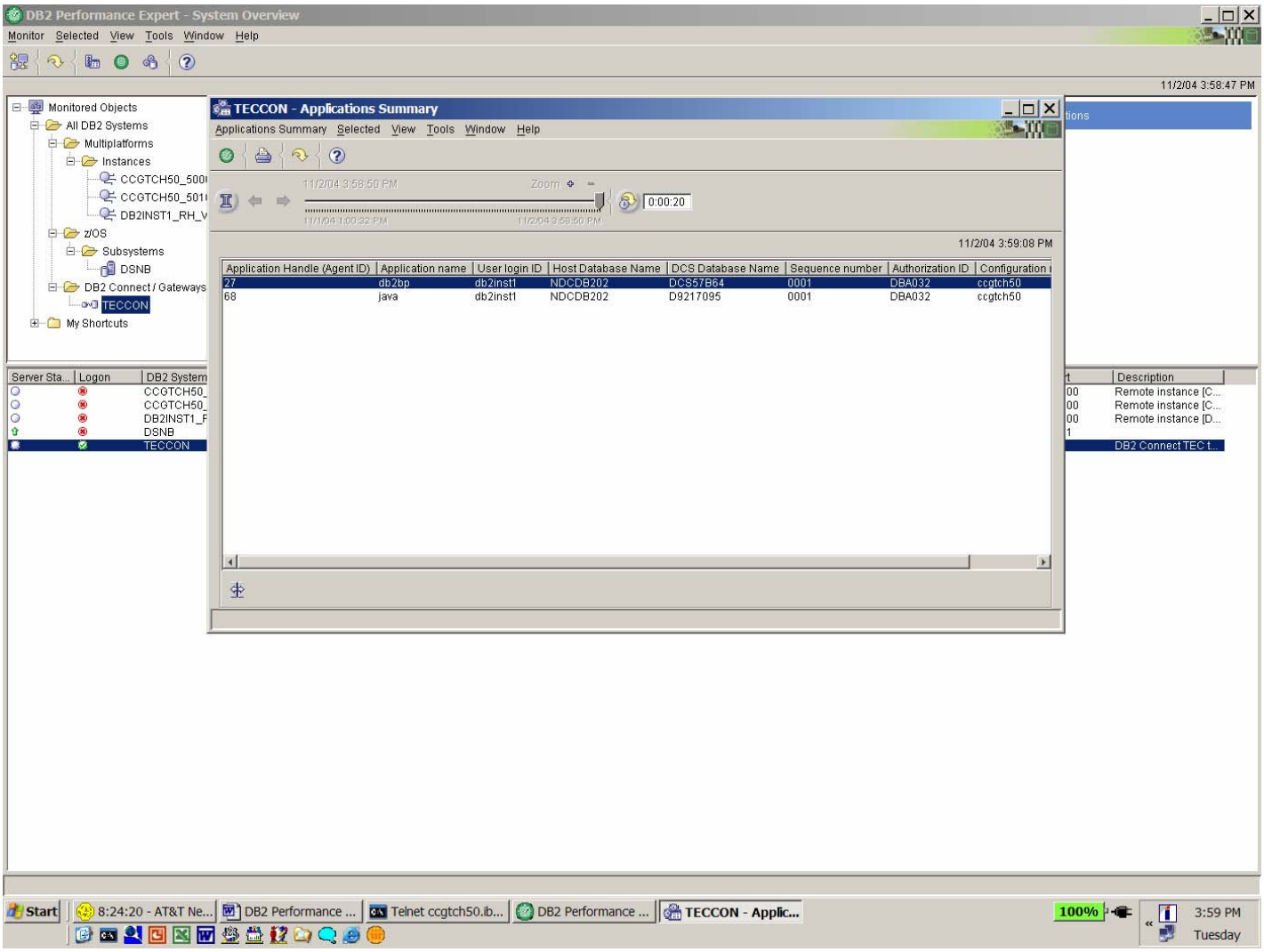
```
Mon Dec 20 11:46:21 CET 2004 - Last compatible data stream version : 3
Mon Dec 20 11:46:21 CET 2004 - Level : java/com/ibm/db2pm/e2e/E2EVersion.java, jE2E, N210_FP2HOT, E073
Mon Dec 20 11:46:21 CET 2004 - Executing command : --start
```

Features and Functions Detail

Monitoring via a DB2 Connect Gateway



DB2 Connect gateways are listed in the PE Client Monitored Objects list. In order to add an object, you must know the DB2 System Alias (eg DSNB). The list to select from is generated from the client node directory. This will already be populated if you set up the DB2 subsystem to be monitored. Next, push the “retrieve” button to generate a list of gateways that are sending information to the PE Server.



The Application Summary window shows the DCS applications that are currently connected to any DB2 subsystem. Any number of DB2 Connect gateways could be represented here. Only key information is shown for each application connection.

Application handle (agent ID)
Shows the system-wide unique ID for the application. On a single-partitioned database, this identifier consists of a 16-bit counter. On a multi-partitioned database, it consists of the coordinating partition number concatenated with a 16-bit counter. In addition, it is the same on every partition where the application might make a secondary connection.

The application handle can be used to uniquely identify an active application (application handle is synonymous with agent ID).

The behavior of the agent ID depends on your DB2 version. When taking snapshots from DB2 with version SQLM_DBMON_VERSION1 or SQLM_DBMON_VERSION2 to a DB2 Universal Database (Version 5 or greater) database, the agent ID returned cannot be used as an application identifier, rather it is the agent PID of the agent serving the application. In this case, an agent ID is returned for back-level compatibility, but internally the DB2 Universal Database server does not recognize the value as an agent ID.

The value for this counter can be used as input to GET SNAPSHOT commands that require an agent ID. When reading event traces, it can be used to match event records with a given application.

Application name
Shows the name of the application running at the client as known to the database manager or DB2 Connect. Together with the Application ID counter, it can be used to relate data items with your application.

In a client/server environment, this name is passed from the client to the server to establish the database connection. For DRDA-AS connections, this name is the DRDA external name. In situations where the client application code page is different from the code page under which the database system monitor is running, you can use the Code page used by application counter to translate this counter.

Authorization ID
Shows the authorization ID of the user who invoked the application that is being monitored. On a DB2 Connect gateway node, this is the user's authorization ID on the host.

Configuration name of client
Shows the NNAME in the database manager configuration file at the client node. You can use this counter to identify the client node that is running the application.

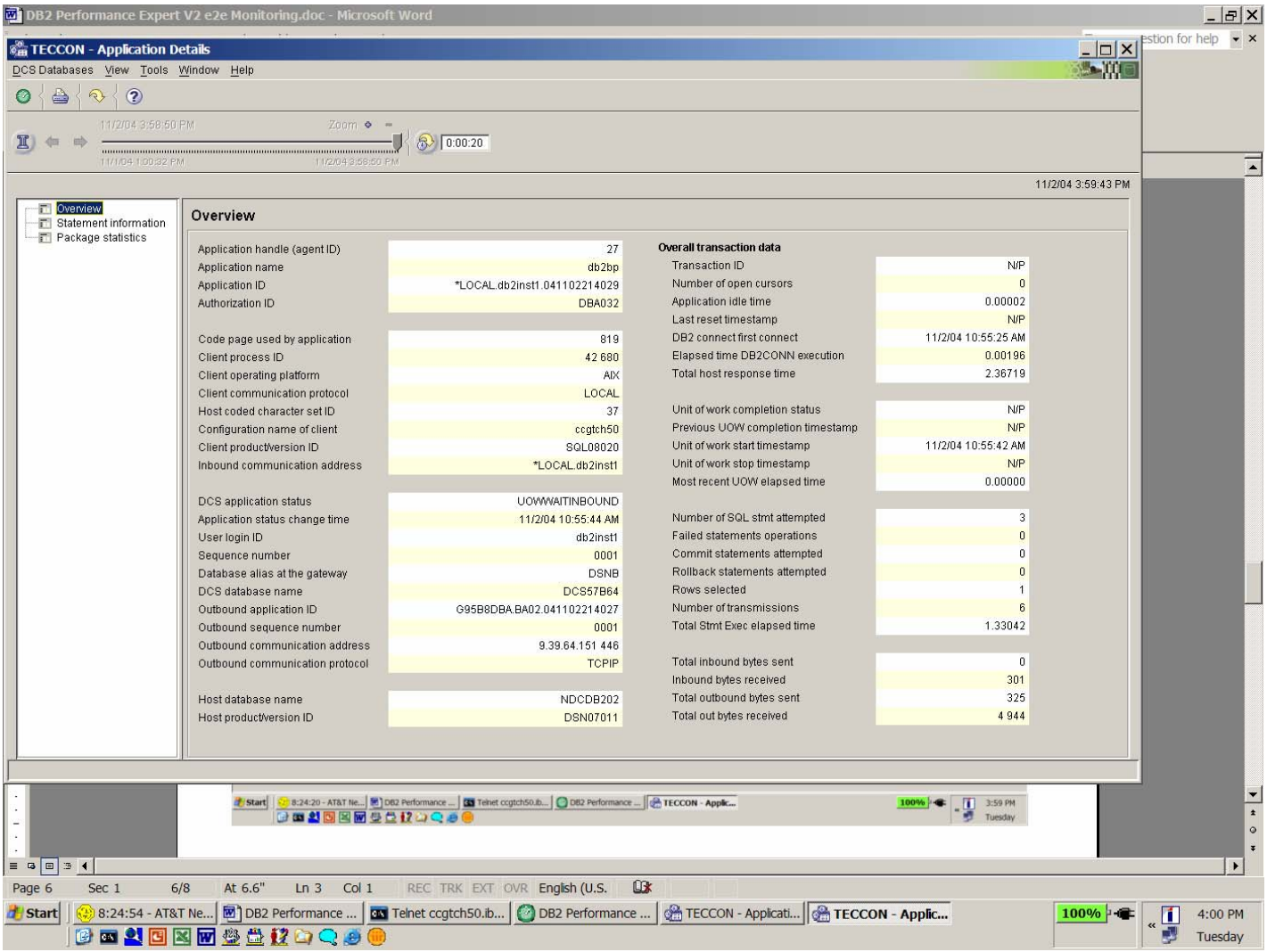
DCS database name
Shows the name of the remote database as cataloged in the DCS directory. Use this counter for problem determination on DCS applications.

Host database name
Shows the real name of the host database for which information is being collected or to which the application is connected. This is the name that was given to the database when it was created.

Sequence number
Is incremented whenever a unit of work ends, that is, when a COMMIT or ROLLBACK terminates a unit of work. Together with the Application ID counter, this counter uniquely identifies a transaction.

User login ID
Shows the ID that the user specified when logging in to the operating system. This ID differs from authorization ID, which the user specifies when connecting to the database.

You can use this counter to determine the operating system user ID of the individual running the application that you are monitoring.



Drilling down on a DCS application listed on the Application Summary screen yields more Overview details about the single, selected application including counters and status information.

Code page used by application
Shows the code page identifier.

For snapshot monitor data, this is the code page at the partition where the monitored application started. This identifier can be used for problem determination for remote applications. You can use this information to ensure that data conversion is supported between the application code page and the database code page or, for DRDA host databases, the host coded character set identifier (CCSID).

For event monitor data, this is the code page of the database for which event data is collected. You can use this counter to determine whether your event monitor application is running under a different code page from that used by the database. Data written by the event monitor uses the database code page. If your event monitor application uses a different code page, you might need to perform some character conversion to make the data readable.

Client process ID
Shows the process ID of the client application that made the connection to the database.

You can use this counter to correlate monitor information such as CPU and I/O time to your client application. In the case of a DRDA-AS connection, this counter is set to 0.

Client operating platform
Shows the operating system on which the client application is running. You can use this counter for problem determination on remote applications.

Client communication protocol
Shows the communication protocol that the client application is using to communicate with the server. You can use this counter for problem determination on remote applications.

Valid values for this counter are:
API Constant Communication Protocol
SQLM_PROT_UNKNOWN (Note 1)
SQLM_PROT_LOCAL none (Note 2)
SQLM_PROT_APPC APPC
SQLM_PROT_TCPIP TCP/IP
SQLM_PROT_IPXSPX IPX/SPX
SQLM_PROT_NETBIOS NETBIOS

Notes:
The client is communicating using an unknown protocol. This value is only returned if future clients connect with a down-level server.
The client is running on the same node as the server and no communications protocol is in use.

Host coded character set ID
Shows the coded character set identifier (CCSID) of the host database. Use this counter for problem determination on DCS applications.

Configuration name of client
Shows the NNAME in the database manager configuration file at the client node. You can use this counter to identify the client node that is running the application.

Client product/version ID
Shows the product and version that is running on the client.

You can use this counter to identify the product and code version of the database client. It is in the form pppvvrrm, where:
ppp identifies the product, which is “SQL” for the DB2 products.
vv identifies a 2-digit version number (with high-order 0 in the case of a 1-digit version).
rr identifies a 2-digit release number (with high-order 0 in the case of a 1-digit release).
m identifies a 1-digit modification level.

Inbound communication address
Shows the communication address of the client. For example, it could be an SNA net ID and LU partner name, or an IP address and port number for TCP/IP.

DCS application status
Shows the current status of the application. It can help you diagnose potential application problems.

Application status change time
Shows the date and time the application entered its current status.

This counter allows you to determine how long an application has been in its current status. If it has been in the same status for a long period of time, this can indicate a problem.

User login ID
Shows the ID that the user specified when logging in to the operating system. This ID differs from authorization ID, which the user specifies when connecting to the database.

You can use this counter to determine the operating system user ID of the individual running the application that you are monitoring.

Sequence number
Is incremented whenever a unit of work ends, that is, when a COMMIT or ROLLBACK terminates a unit of work. Together with the Application ID counter, this counter uniquely identifies a transaction.

Database alias at the gateway
Shows the alias used at the DB2 Connect gateway to connect to the host database.

DCS database name
Shows the name of the remote database as cataloged in the DCS directory. Use this counter for problem determination on DCS applications.

Outbound application ID
Is generated when the application connects to the DRDA host database. It is used to connect the DB2 Connect gateway to the host, while the application ID is used to connect a client to the DB2 Connect gateway.

You can use this counter in conjunction with the Application ID counter to correlate the client and server parts of the application information. This identifier is unique across the network.

Format: Network.LU Name.Application instance
Example: CAIBMTOR.OSFDBM0.930131194520
Details: This application ID is the displayable format of an actual SNA LUWID (logical unit-of-work ID) that flows on the network when an APPC conversation is allocated. APPC-generated application IDs are made up by concatenating the network name, the LU name, and the LUWID instance number, which creates a unique label for the client/server application. The network name and LU name can each be a maximum of 8 characters. The application instance corresponds to the 12-decimal-character LUWID instance number.

Outbound sequence number
Is reserved for future use. In this release, its value will is always 0001. It can contain different values in future releases of the product.

Outbound communication address
Shows the communication address of the target database. For example, it could be an SNA net ID and LU partner name, or an IP address and port number for TCP/IP.

Use this counter for problem determination on DCS applications.

Outbound communication protocol
Shows the communication protocol used between the DB2 Connect gateway and the host.

Use this counter for problem determination on DCS applications. Valid values are:
SQLM_PROT_APPC
SQLM_PROT_TCPIP

Host database name
Shows the real name of the host database for which information is being collected or to which the application is connected. This is the name that was given to the database when it was created.

Host product/version ID
Shows the product and version that is running on the server.

This counter is used to identify the product and code version of the DRDA host database product. It is in the form ppvvrrm, where:
ppp identifies the host DRDA product:
ARI for DB2 for VSE & VM
DSN for DB2 for OS/390 and z/OS
QSQ for DB2 UDB for AS/400
SQL for other DB2 products
vv identifies a 2-digit version number (with high-order 0 in the case of a 1-digit version)
rr identifies a 2-digit release number (with high-order 0 in the case of a 1-digit release)
m identifies a 1-digit modification level

Transaction ID
Shows the unique transaction identifier across all databases generated by a transaction manager in a two-phase commit transaction.

You can use this identifier to correlate the transaction generated by the transaction manager with the transactions executed against multiple databases. In addition, it can help you diagnose transaction manager problems by tying database transactions that involve a two-phase commit protocol with the transactions that are originated by the transaction manager.

Number of open cursors
Shows the number of cursors currently open for an application.

Use this counter to assess how much memory is being allocated. The amount of memory allocated by the DB2 client, DB2 Connect, or the database agent on the target database is related to the number of cursors that are currently open. Knowing this information can help with capacity planning. For example, each open cursor that is blocking has a buffer size of RQRIOBLK. If DEFERRED_PREPARE is enabled, two buffers are allocated.

Application idle time
Shows the number of seconds since an application issued any requests to the server. This includes applications that have not terminated a transaction, for example, not issued a commit or rollback.

You can use this information to implement applications that force users that have been idle for a specified number of seconds.

Last reset timestamp
Shows the date and time that the monitor counters were reset for the application issuing the GET SNAPSHOT.

You can use this counter to determine the scope of information returned by the database system monitor. If the database manager counters have never been reset, this counter is zero. The database manager counters is only reset if you reset all active databases.

DB2 Connect first connect
Shows the date and time at which the first connection to the host database was initiated from the DB2 Connect gateway.

Elapsed time DB2CONN execution
Shows the time, in seconds and microseconds, at the DB2 Connect gateway to process an application request (since the connection was established), or to process a single statement.

Use this counter to determine what portion of the overall processing time is due to DB2 Connect gateway processing.

Total host response time
For a DCS statement, this is the elapsed time between the time that the statement was sent from the DB2 Connect gateway to the host for processing and the time when the result was received from the host.

For a DCS database or DCS application, it is the sum of the elapsed times for all the statements that were executed for a particular database or application.

For a data transmission, this is the sum of host response times for all the statements that used this many data transmissions.

Use this counter with the Overall transaction data - Total outbound bytes sent and Overall transaction data - Total out bytes received counters to calculate the outbound response time (transfer rate):
((Total outbound bytes sent) + (Total out bytes received)) / Total host response time

Unit of work completion status
Shows the status of the unit of work and how it stopped.

You can use this counter to determine if the unit of work ended due to a deadlock or abnormal termination. It can have been:
Committed due to a commit statement
Rolled back due to a rollback statement
Rolled back due to a deadlock
Rolled back due to an abnormal termination
Committed at normal application termination.
Unknown as a result of a FLUSH EVENT MONITOR command for which units of work were in progress

Note: API users should refer to the sqlmon.h header file containing definitions of database system monitor constants.

Previous UOW completion timestamp
Shows the time the unit of work completed.

You can use this counter with the Overall transaction data - Unit of work stop timestamp counter to calculate the total elapsed time between COMMIT or ROLLBACK points, and with the Overall transaction data - Unit of work start timestamp counter to calculate the time spent in the application between units of work:
For applications currently within a unit of work, this is the time at which the latest unit of work completed.
For applications not currently within a unit of work (the application has completed a unit of work, but not yet started a new one), this is the stop time of the last unit of work that completed prior to the one that just completed. The stop time of the one just completed is indicated by the Overall transaction data - Unit of work stop timestamp counter.
For applications within their first unit of work, this is the database connection request completion time.

Unit of work start timestamp
Shows the date and time at which the unit of work first required database resources.

This resource requirement occurs at the first SQL statement execution of that unit of work:
For the first unit of work, it is the time of the first database request (SQL statement execution) after connection completion
For subsequent units of work, it is the time of the first database request (SQL statement execution) after the previous COMMIT or ROLLBACK.

Note: The SQL Reference defines the boundaries of a unit of work as the COMMIT or ROLLBACK points.

Unit of work stop timestamp
Shows the date and time at which the most recent unit of work completed, which occurs when database changes are committed or rolled back.

You can use this counter with the Overall transaction data - Previous UOW completion timestamp counter to calculate the total elapsed time between COMMIT or ROLLBACK points, and with the Overall transaction data - Unit of work start timestamp counter to calculate the elapsed time of the latest unit of work. The timestamp contents are set as follows:
When the application has completed a unit of work and has not yet started a new one (as defined by the Overall transaction data - Unit of work start timestamp counter). this counter is a valid, nonzero timestamp
When the application is currently executing a unit of work, this counter contains zeros
When the application first connects to the database, this counter is set to the connection completion time.

As a new unit of work is started, the contents of this counter are moved to the Overall transaction data - Previous UOW completion timestamp counter.

Most recent UOW elapsed time
Shows the elapsed execution time of the most recently completed unit of work.

Use this counter as an indicator of the time it takes for units of work to complete.

Number of SQL stmt attempted
Shows the number of SQL statements that have been attempted since the latter of: application startup, database activation, or last reset.

For a data transmission, this is the number of SQL statements that have been attempted against this DCS database or in this DCS application since the database was activated, the connection to it was established by the application, or RESET MONITOR was issued against the database, and that used this number of data transmissions between the DB2 Connect gateway and the host during statement processing.

Use this counter to measure the database activity for a database or application. To calculate the SQL statement throughput for a given period, you can divide this counter by the elapsed time between two snapshots.

For a data transmission, use this counter to get statistics on how many statements used two, three, four, etc. data transmissions during their processing. At least two data transmissions are necessary to process a statement: a send and a receive. These statistics can give you a better idea of the database or application activity and network traffic for a database or an application.

Rows selected
Shows the number of rows that have been selected and returned to the application.

You can use this counter to gain insight into the current level of activity within the database. This counter does not include a count of rows read for actions, such as COUNT(*) or joins.

For a federated system;, you can calculate the average time to return a row to the federated server from the data source:
average time = rows returned / aggregate query response time

You can use these results to modify CPU speed or communication speed parameters in SYSCAT.SERVERS. Modifying these parameters can impact whether the optimizer does or does not send requests to the data source.

Note: This counter is collected at the DCS database and DCS application if the gateway being monitored is at DB2 Version 7.2 or lower.

Number of transmissions
Shows the number of data transmissions between the DB2 Connect gateway and the host that was used to process this DCS statement. (One data transmission consists of one send or one receive.)

Use this counter to get a better understanding of the reasons why a particular statement took longer to execute. For example, a query returning a large result set might need many data transmissions to complete.

Total stmt exec elapsed time
For a DCS statement, this is the elapsed time spent processing an SQL request on a host database server. This value is reported by this server. In contrast to the Overall transaction data - Total host response time counter, this counter does not include the network elapsed time between DB2 Connect and the host database server.

At other levels, this value represents the sum of the host execution times for all the statements that were executed for a particular database or application, or for those statements that used a given number of data transmissions.

Use this counter, along with other elapsed time monitor elements, to evaluate the database server's processing of SQL requests and to help isolate performance issues.

Subtract the value for this counter from the value for the Overall transaction data - Total host response time counter to calculate the network elapsed time between DB2 Connect and the host database server.

Total inbound bytes sent
Shows the number of bytes sent by the DB2 Connect gateway to the client, excluding communication protocol overhead, for example, TCP/IP or SNA headers.

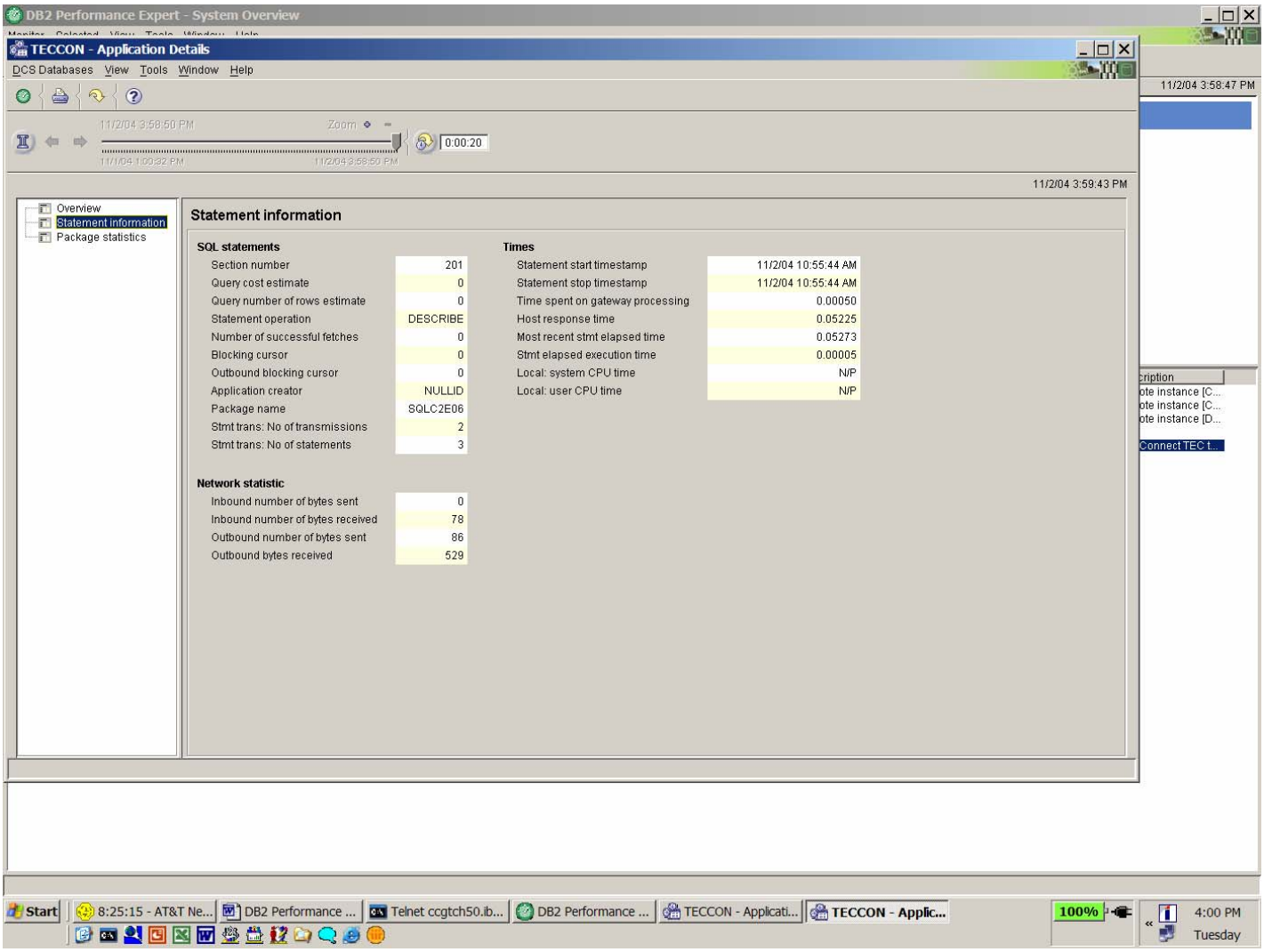
Inbound bytes received
Shows the number of bytes received by the DB2 Connect gateway from the client, excluding communication protocol overhead (for example, TCP/IP or SNA headers).

Total outbound bytes sent
Shows the number of bytes sent by the DB2 Connect gateway to the host, excluding communication protocol overhead, for example, TCP/IP or SNA headers.

For a data transmission, this is the number of bytes sent by the DB2 Connect gateway to the host during the processing of all the statements that used this number of data transmissions.

Total out bytes received
Shows the number of bytes received by the DB2 Connect gateway from the host, excluding communication protocol overhead, for example, TCP/IP or SNA headers.

For a data transmission, this is the number of bytes received by the DB2 Connect gateway from the host during the processing of all the statements that used this number of data transmissions.



The Statement Information section of the Application Details screen provides information about the last SQL statement that was executed by the application. An analyst may want to look at the number of rows estimate and compare it to the actual rows retrieved to determine if the optimizer is doing a good job of estimating and accessing the data. The network statistics section gives a measure of the throughput of the network from the gateway to the host database. Most importantly, the times section shows the portion of the total time that gateway and host are using.

Section number
Shows the internal section number in the package for the SQL statement that is currently processing or has processed most recently.

For a static SQL, you can use this counter together with the creator, package version, and package name to query the SYSCAT.STATEMENTS system catalog table and obtain the static SQL statement text, using the sample query as follows:
SELECT SEQNO, SUBSTR(TEXT,1,120)
FROM SYSCAT.STATEMENTS
WHERE PKGNAME = 'package_name' AND
PKGSCHEMA = 'creator' AND
VERSION = 'package_version_id' AND
SECTNO = section_number
ORDER BY SEQNO
Note: This query can cause lock contentions. Therefore, try to use it only when there is little other activity against the database.

Query cost estimate
Shows the estimated cost, in timerons, for a query, as determined by the SQL compiler. It allows correlation of actual run-time with the compile-time estimates. In addition, it returns information for the following SQL statements when you are monitoring DB2 Connect.
PREPARE represents the relative cost of the prepared SQL statement.
FETCH contains the length of the row retrieved if the DRDA server is DB2 for OS/400. Otherwise, this counter is set to zero.

Note: If the DRDA server is DB2 for OS/390 and z/OS, this estimate could be higher than 2**32 - 1 (the maximum integer number that can be expressed through an unsigned long variable). In that case, the value returned by the monitor for this counter is 2**32 - 1.

Query number of rows estimate
Shows the estimated number of rows that is returned by a query. This estimate by the SQL compiler can be compared with the run-time actuals.

This counter also returns information for the following SQL statements when you are monitoring DB2 Connect.
INSERT, UPDATE, and DELETE indicate the number of rows affected.
PREPARE estimates the number of rows that are returned if the DRDA server is DB2 Universal Database, DB2 for VM and VSE, or DB2 for OS/400.
FETCH sets to the number of rows fetched if the DRDA server is DB2 for OS/400. Otherwise, this counter is set to zero.

Statement operation
Shows the statement operation that is currently being processed or has processed most recently (if none is currently running).

You can use this counter to determine the operation that is executing or recently finished. It can be one of the following.
For SQL operations:
SELECT
PREPARE
EXECUTE
EXECUTE IMMEDIATE
OPEN
FETCH
CLOSE
DESCRIBE
STATIC COMMIT
STATIC ROLLBACK
FREE LOCATOR

PREP_COMMIT
CALL
PREP_OPEN
PREP_EXEC
COMPILE
For non-SQL operations:
RUN STATISTICS
REORG
REBIND
REDISTRIBUTE
GET TABLE AUTHORIZATION
GET ADMINISTRATIVE AUTHORIZATION

Note: API users should refer to the sqlmon.h header file containing definitions of database system monitor constants.

Number of successful fetches
For statement snapshot monitoring and the statement event type, this is the number of successful fetches performed on a specific cursor.

For DCS statement snapshot monitoring, this is the number of attempted physical fetches during the execution of a statement regardless of how many rows were fetched by the application. That is, this counter shows the number of times the server needed to send a reply data back to the gateway while processing a statement.

You can use this counter to gain insight into the current level of activity within the database manager.
Background and tuning information
For performance reasons, a statement event monitor does not generate a statement event record for every FETCH statement. A record event is only generated when a FETCH returns a nonzero SQLCODE.

Blocking cursor
Indicates whether the statement being executed is using a blocking cursor.

Using blocking for data transfer for a query can improve its performance. The SQL used for a query can affect the use of blocking and might require some modification.

Outbound blocking cursor
Indicates whether blocking is used for data transfer from the DRDA server to the DB2 Connect gateway for a particular query.

Using blocking for data transfer for a query can improve its performance. The SQL used for a query can affect the use of blocking and might require some modification.

Application creator
Shows the authorization ID of the user who precompiled the application.

You can use this counter to identify the SQL statement that is processing, in conjunction with the CREATOR column of the package section information in the catalogs.

Package name
Shows the name of the package that contains the SQL statement that is currently executing.

You can use this counter to identify the application program and the SQL statement that is executing.

Stmt trans: No of transmissions
Shows the number of data transmissions between the DB2 Connect gateway and the host that was used to process this DCS statement. One data transmission consists of one send or one receive.

Stmt trans: No of statements
Shows the number of SQL statements that have been attempted since the latter of: application startup, database activation, or last reset.

For a data transmission, this is the number of SQL statements that have been attempted against this DCS database or in this DCS application since the database was activated, the connection to it was established by the application, or RESET MONITOR was issued against the database, and that used this number of data transmissions between the DB2 Connect gateway and the host during statement processing.

Inbound number of bytes sent
Shows the number of bytes sent by the DB2 Connect gateway to the client, excluding communication protocol overhead, for example, TCP/IP or SNA headers.

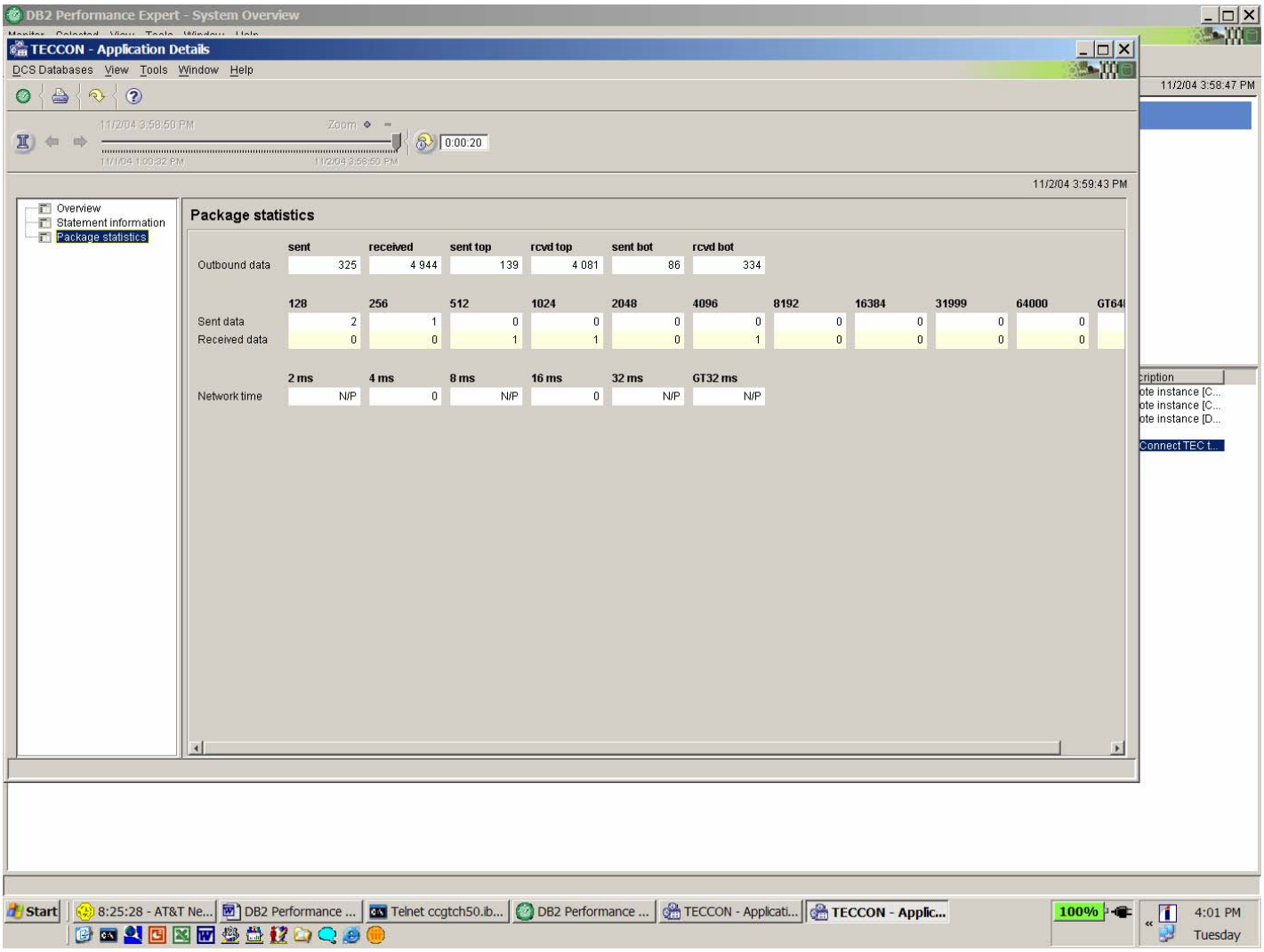
Inbound number of bytes received
Shows the number of bytes received by the DB2 Connect gateway from the client, excluding communication protocol overhead, for example, TCP/IP or SNA headers.

Outbound number of bytes sent
Shows the number of bytes sent by the DB2 Connect gateway to the host, excluding communication protocol overhead, for example, TCP/IP or SNA headers.

For a data transmission, this is the number of bytes sent by the DB2 Connect gateway to the host during the processing of all the statements that used this number of data transmissions.

Outbound bytes received
Shows the number of bytes received by the DB2 Connect gateway from the host, excluding communication protocol overhead, for example, TCP/IP or SNA headers.

For a data transmission, this is the number of bytes received by the DB2 Connect gateway from the host during the processing of all the statements that used this number of data transmissions.



The Package Statistics section of the Application Details screen helps you determine the throughput of the network between the host database and the gateway. Outbound data is measured in bytes. Sent/Received data and Network time counters are measured in number of statements.

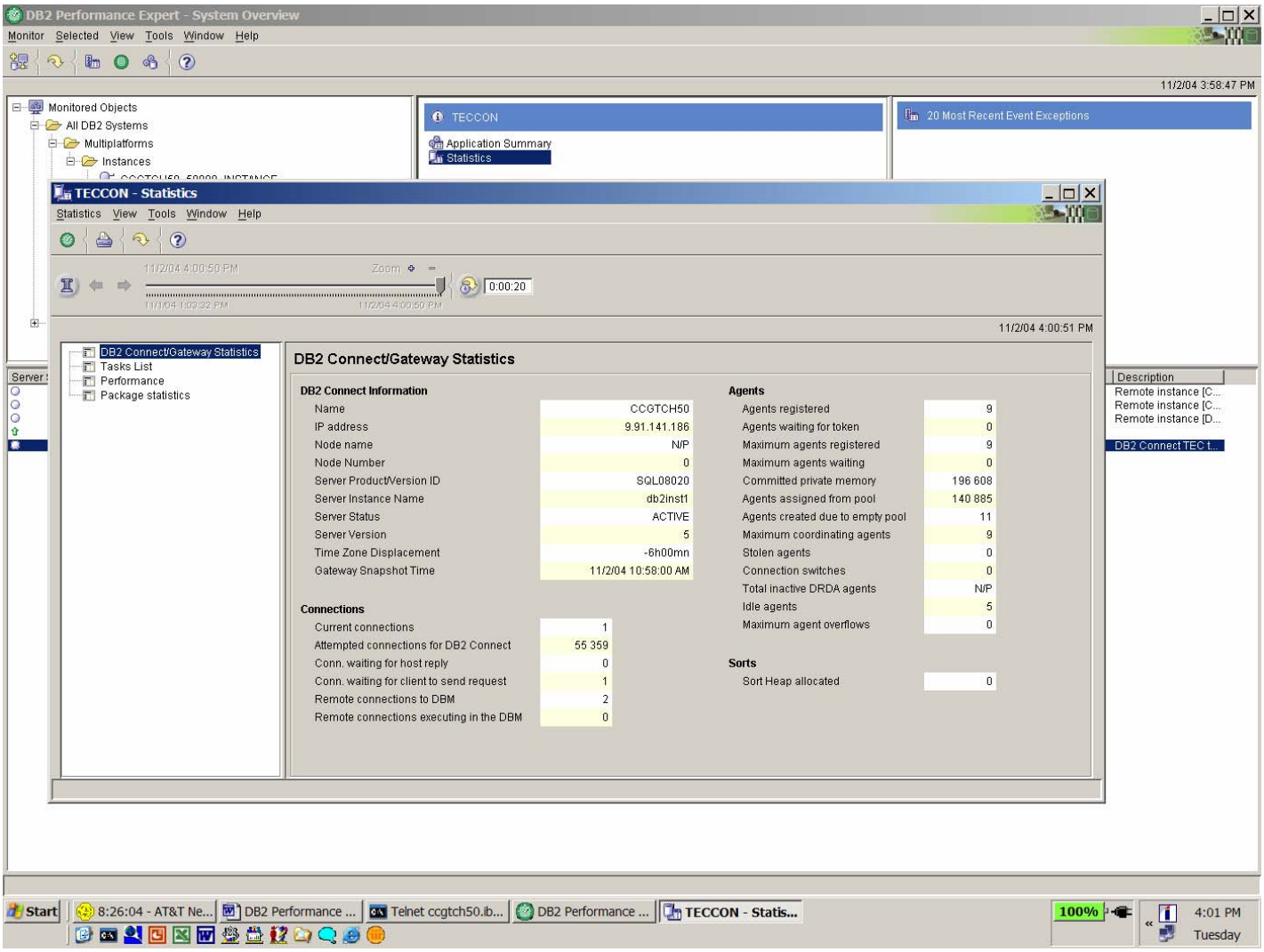
Outbound data - sent
Shows the number of bytes sent by the DB2 Connect gateway to the host, excluding communication protocol overhead, for example, TCP/IP or SNA headers.
For a data transmission, this is the number of bytes sent by the DB2 Connect gateway to the host during the processing of all the statements that used this number of data transmissions.

Sent data - 128
Shows the number of statements with outbound bytes sent from 1 through 128.

Received data - 128
Shows the number of statements with outbound bytes received from 1 through 128.

Network time - 2 ms
Shows the number of statements whose network time was less than, or equal to, 2 milliseconds.

Network time is the difference between the host response time and the elapsed execution time for a statement.



The Statistics window under the DB2 Connect gateway object monitoring section provides DCS snapshot information. Usage of this information is discussed in the Practical Usage section. The Agents section is particularly useful in determining if the gateway is effectively handling the demand.

Gateway snapshot time
Shows the date and time at which the database system monitor information was collected.

IP address
Shows the current IP address.

Name
Shows the name in the database manager configuration file at the client node. It identifies the client node that is running the application.

If you are using the database system monitor APIs, note that the API constant SQLM_IDENT_SZ is used to define the length of the name. Only the first 8 characters are currently used.

Node name
Shows the name of the node being monitored by the database system monitor. It identifies the database server node you are monitoring.

This information can be useful if you are saving your monitor output in a file or database for later analysis and you need to differentiate the data from different database server nodes. This node name is determined based on the NNAME configuration parameter.

If you are using the database system monitor APIs, note that the API constant SQLM_IDENT_SZ is used to define the length of the name. Only the first 8 characters are currently used.

Node number
Shows the number assigned to the node in the db2nodes.cfg file.

Server instance name
Shows the name of the database manager instance for which the snapshot was taken.

If a system contains more than one instance of the database manager, this name is used to uniquely identify the instance for which the snapshot call was issued. Along with configuration NNAME at monitoring (server) node, this information can be useful if you are saving your monitor output in a file or database for later analysis, and you need to differentiate the data from different instances of the database manager.

If you are using the database system monitor APIs, note that the API constant SQLM_IDENT_SZ is used to define the length of this counter. Only the first 8 characters are currently used.

Server product/version ID
Shows the product and version that is running on the server in the form pppvvrrm, where
ppp stands for SQL
vv identifies a 2-digit version number (with high-order 0 in the case of a 1-digit version)
rr identifies a 2-digit release number (with high-order 0 in the case of a 1-digit release)
m identifies a 1-digit modification level

If you are using the database system monitor APIs, note that the API constant SQLM_IDENT_SZ is used to define the length of this counter. Only the first 8 characters are currently used.

Server status
Shows whether the server is active or inactive.

Server version
Shows the version of the server returning the information.

Time zone displacement
Shows the number of seconds that the local time zone is displaced from Greenwich Mean Time (GMT).

Current connections
Shows the number of applications that are currently connected to the database

You can use this counter to understand the level of activity within a database and the amount of system resource being used. It can help you adjust the setting of the MAXAPPLS (maximum number of applications) and MAXAGENTS (maximum number of agents) configuration parameters. If its value is always the same as MAXAPPLS, you may want to increase the value of MAXAPPLS. If it is always less than MAXAPPLS, you may want to increase the value of MAXAGENTS.

Attempted connections for DB2 Connect
Shows the total number of current connections initiated from remote clients to the instance of the database manager that is being monitored. It also shows the level of activity between this instance and other instances of the database manager.

Conn. waiting for host reply
Shows the current number of connections to the host databases that are handled by the DB2 Connect gateway and are waiting for a reply from the host.

Conn. waiting for client to send request
Shows the current number of connections to the host databases that are handled by the DB2 Connect gateway and are waiting for the client to send a request.

Remote connections to DBM
Shows the current number of connections initiated from remote clients to the instance of the database manager that is being monitored.

Remote connections executing in the DBM
Shows the number of remote applications that are currently connected to a database and are currently processing a unit of work within the database manager instance being monitored.

Agents registered
Shows the number of agents registered in the database manager instance that is being monitored.

You can use this counter in conjunction with the Agents - Agents waiting for token counter to determine the percentage of agents waiting for a token so they can perform a transaction in the database manager. If the percentage is high, you can improve the concurrency in the database manager by increasing the MAXCAGENTS (maximum number of concurrent agents) configuration parameter. This number is always greater than, or equal to, the number of local databases with current connects.
Background and tuning information
Each application has a dedicated agent to process database requests within the database manager. Each agent has to get a token before it can perform a transaction. The maximum number of agents that can execute database manager transactions is limited by the MAXCAGENTS configuration parameter.

Agents waiting for token
Shows the number of agents waiting for a token so they can perform a transaction in the database manager.

You can use this counter in conjunction with the Agents - Agents registered counter to determine the percentage of "sleeping" agents. If the percentage is high, you can you can improve the concurrency in the database manager by increasing the MAXCAGENTS (maximum number of concurrent agents) configuration parameter.
Background and tuning information
Each application has a dedicated agent to process database requests within the database manager. Each agent has to get a token before it can perform a transaction. The maximum number of agents that can execute database manager transactions is limited by the MAXCAGENTS configuration parameter.

Maximum agents registered
Shows the maximum number of agents that the database manager has registered at the same time since it was started.

You can use this counter to evaluate your setting of the MAXAGENTS (maximum number of agents) configuration parameter. The number of agents registered at the time the snapshot was taken is recorded in the Agents - Agents registered counter.

Maximum agents waiting
Shows the maximum number of agents that have been waiting for a token at the same time since the database manager was started.

You can use this counter to evaluate your setting of the MAXCAGENTS (maximum number of concurrent agents) configuration parameter. If the MAXCAGENTS parameter is set to its default value, which is -1, no agents should wait for a token and the value for this counter should be zero.

The number of agents waiting for a token at the time the snapshot was taken is recorded in the Agents waiting for token counter.

Committed private memory
Shows the amount of private memory that the instance of the database manager has committed at the time of the snapshot.

You can use this counter to set the MIN_PRIV_MEM (minimum committed private memory)configuration parameter to ensure you have enough private memory available. This counter is only applicable to platforms containing an agent pool, such as OS/2.

Agents assigned from pool
Shows the number of agents assigned by an agent pool.

Agents created due to empty pool
Shows the number of agents created because the agent pool was empty.

Maximum coordinating agents
Shows the maximum number of coordinating agents working at one time.

Stolen agents
Shows the number of times that agents are stolen from an application. Agents are stolen when an idle agent associated with an application is reassigned to work on a different application.

Connection switches
Shows the number of the times that an agent from the agent pool was primed with a connection and was stolen for use with a different DRDA database.

Total inactive DRDA agents
Shows the number of connections made by a subagent to the database at the node.

Idle agents
Shows the number of agents in the agent pool that are currently unassigned to an application and are, therefore, idle.

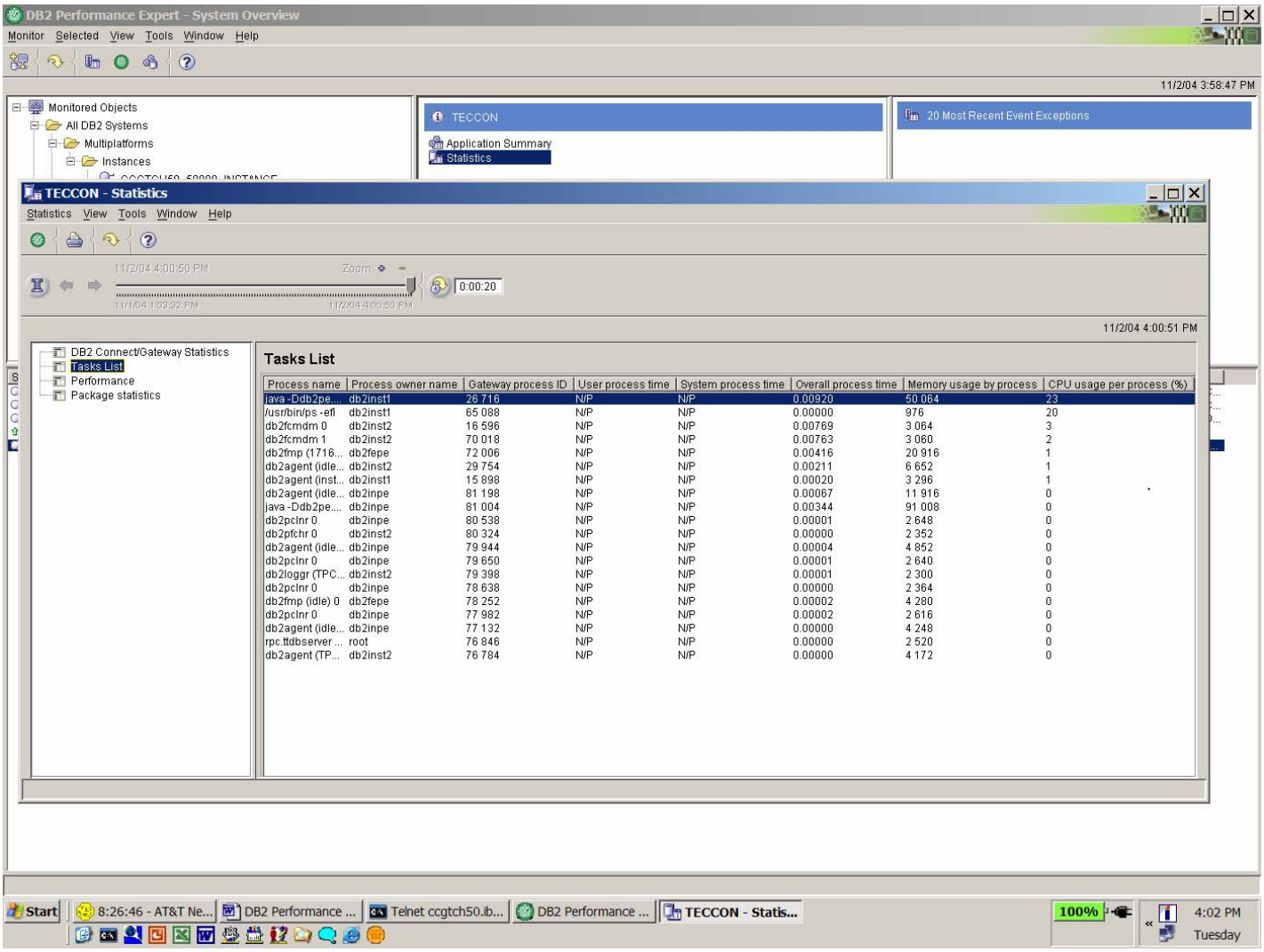
You can use this counter to set the MAX_IDLEAGENTS (maximum number of idle agents) configuration parameter. Having idle agents available to service requests for agents can improve the performance.

Maximum agent overflows
Shows the number of times a request to create a new agent was received when the MAXAGENTS (maximum number of agents) configuration parameter had already been reached.

Sort heap allocated
Shows the total number of allocated pages of sort heap space for all sorts at the level chosen and at the time the snapshot was taken.

The amount of memory allocated for each sort can be part of or the entire sort heap size available. Sort heap size is the amount of memory available for each sort as defined in the SORTHEAP database configuration parameter. It is possible for a single application to have concurrent sorts active. For example, in some cases a SELECT statement with a subquery can cause concurrent sorts. Information can be collected at two levels:
At the database manager level, it represents the sum of sort heap space allocated for all sorts in all active databases in the database manager.
At the database level, it represents the sum of the sort heap space allocated for all sorts in a database.
Background and tuning information
Normal memory estimates do not include sort heap space. If excessive sorting occurs, the extra memory used for the sort heap should be added to the base memory requirements for running the database manager. Generally, the larger the sort heap, the more efficient the sort. Appropriate use of indexes can reduce the amount of sorting required.

You can use the information returned at the database manager level to tune the SHEAPTHRES configuration parameter. If the value is greater than, or equal to, SHEAPTHRES, the sorts are not getting the full sort heap as defined by the SORTHEAP parameter.



The Tasks List screen displays information about the processing workload on the gateway. The memory and process id columns are only filled in if the PE Agent is running on a Unix or Linux based server. Use this screen to determine if the gateway server is overloaded by DB2 Connect or some other collocated application.

CPU usage per process (%)
Shows the percentage of time that a process used the CPU since the last update.

Gateway process ID
Shows the numerical identifier that uniquely distinguishes a process while it runs.

Memory usage by process
Shows the current working set of a process, in kilobytes. The current working set is the number of pages currently resident in memory.

This column only contains a value if Performance Expert Agent is installed on a UNIX-based operating system, such as AIX, HP-UX, Linux, Linux on zSeries, or the Solaris Operating Environment. Otherwise, these columns show N/P (Not present).

Overall process time
Shows the sum of the times contained in the System process time and User process time columns.

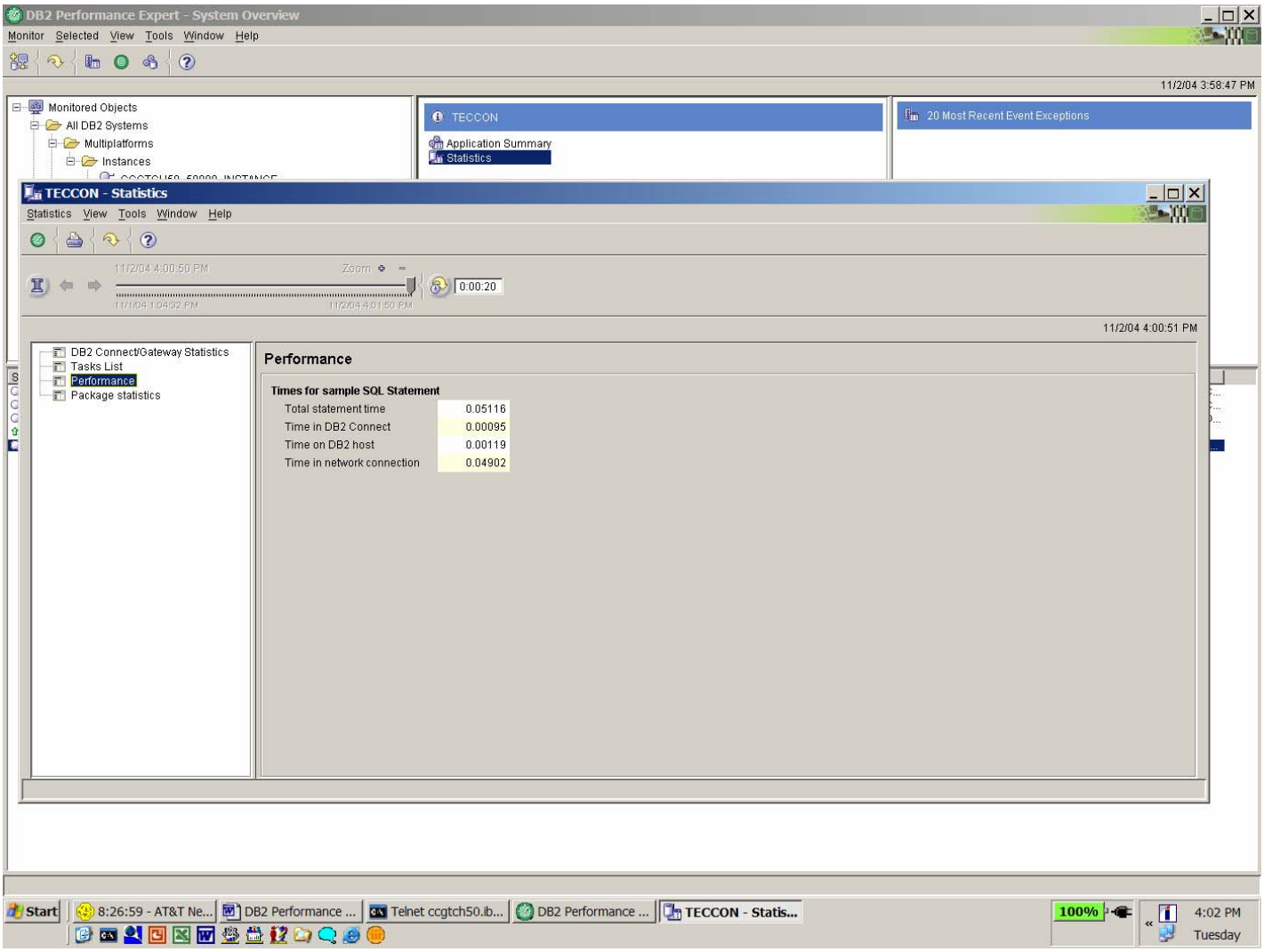
Process name
Shows the name of the process.

Process owner name
Shows the session ID that owns the process.

This column only contains a value if Performance Expert Agent is installed on a UNIX-based operating system, such as AIX, HP-UX, Linux, Linux on zSeries, or the Solaris Operating Environment. Otherwise, these columns show N/P (Not present).

System process time
Shows the total system CPU time, in seconds and microseconds, used by the database manager agent process, the unit of work, or the statement.

User process time
Shows the total user CPU time, in seconds and microseconds, used by the database manager agent process, the unit of work, or the statement.



The Performance screen displays information obtained from running a sample SQL statement between the gateway and the host database. This key screen illustrates where a bottleneck may be located. For instance, an analyst could review past executions of the statement to determine the normal time for each segment. Then, by looking at an execution during a problem time, the offending segment may contain a larger time value than normal. For example, if the Time in network connection (in seconds and microseconds) is normally around .04, a current value of 5.2 may point towards a network issue.

Total statement time
This counter shows the response time between start and end (response) of a SQL statement measured by DB2 Connect. For a DCS statement, this counter shows the elapsed time between the time the statement was sent from the DB2 Connect gateway to the host for processing and the time at which the result was received from the host.

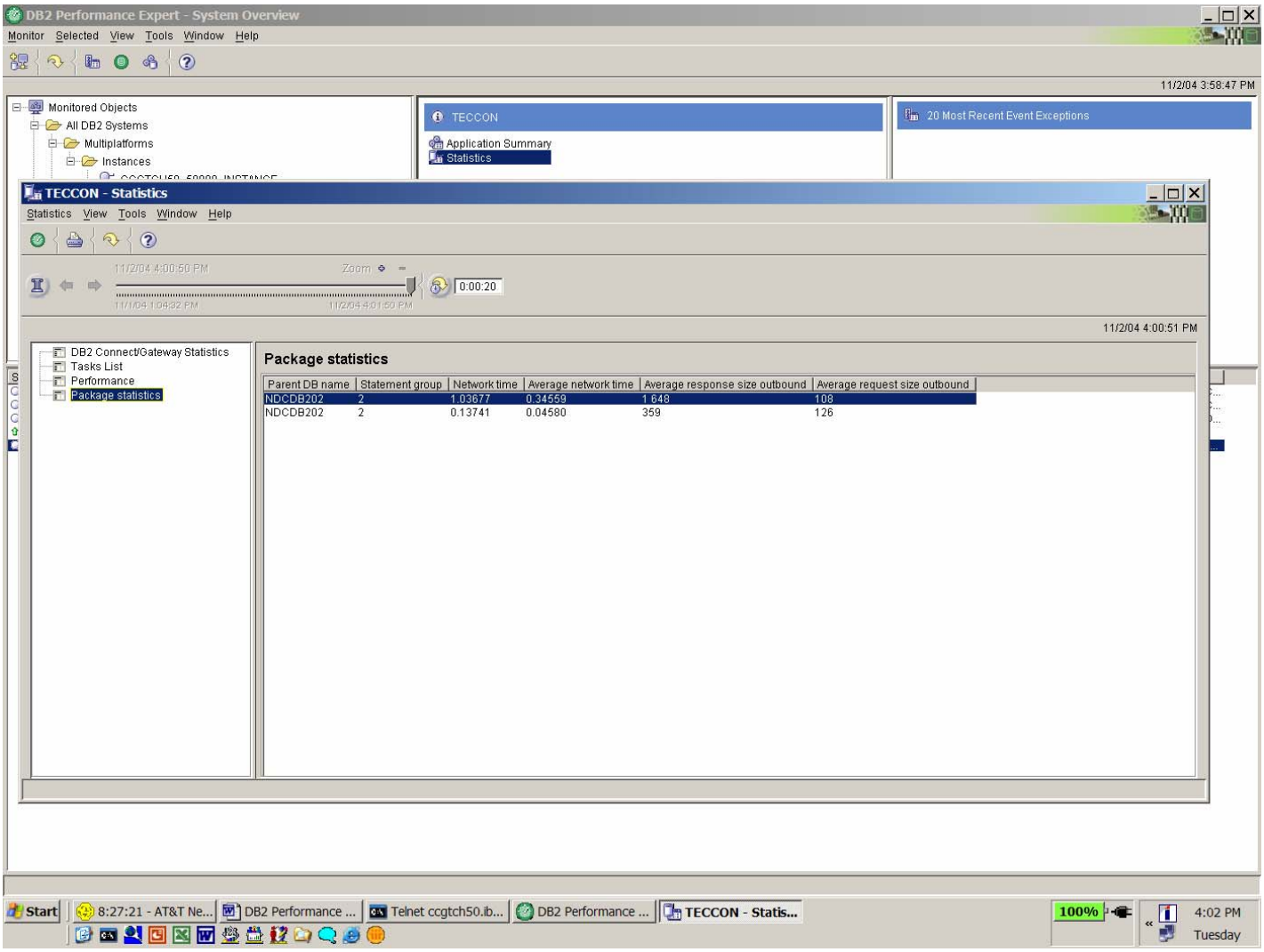
For a DCS database or application, this counter shows the sum of the elapsed times for all the statements that were executed for a particular application or database.

This counter shows the sum of the values for the Times for sample SQL statement - Time in DB2 Connect and Times for sample SQL statement - Time in network connection counters.

Time in DB2 Connect
Shows the total time, in seconds and microseconds, at the DB2 Connect gateway to process the sample statement.

Time on DB2 host
Shows the total time, in seconds and microseconds, that was reported by the DB2 host system to DB2 Connect for this SQL statement.

Time in network connection
Is the calculated time spent in the network in seconds and microseconds. = (Total Statement Time - (Time in DB2 Connect + Time on DB2 host))



The Package statistics screen displays details about packages that were received at the gateway. The average time is generated from the sample SQL statement.

Average network time
Show the result of the value for the Network time counter divided by the number of SQL chains being transferred.

Average request size outbound
Shows the number of bytes received by the DB2 Connect gateway from the client, excluding communication protocol overhead, divided by the number of SQL chains being transferred.

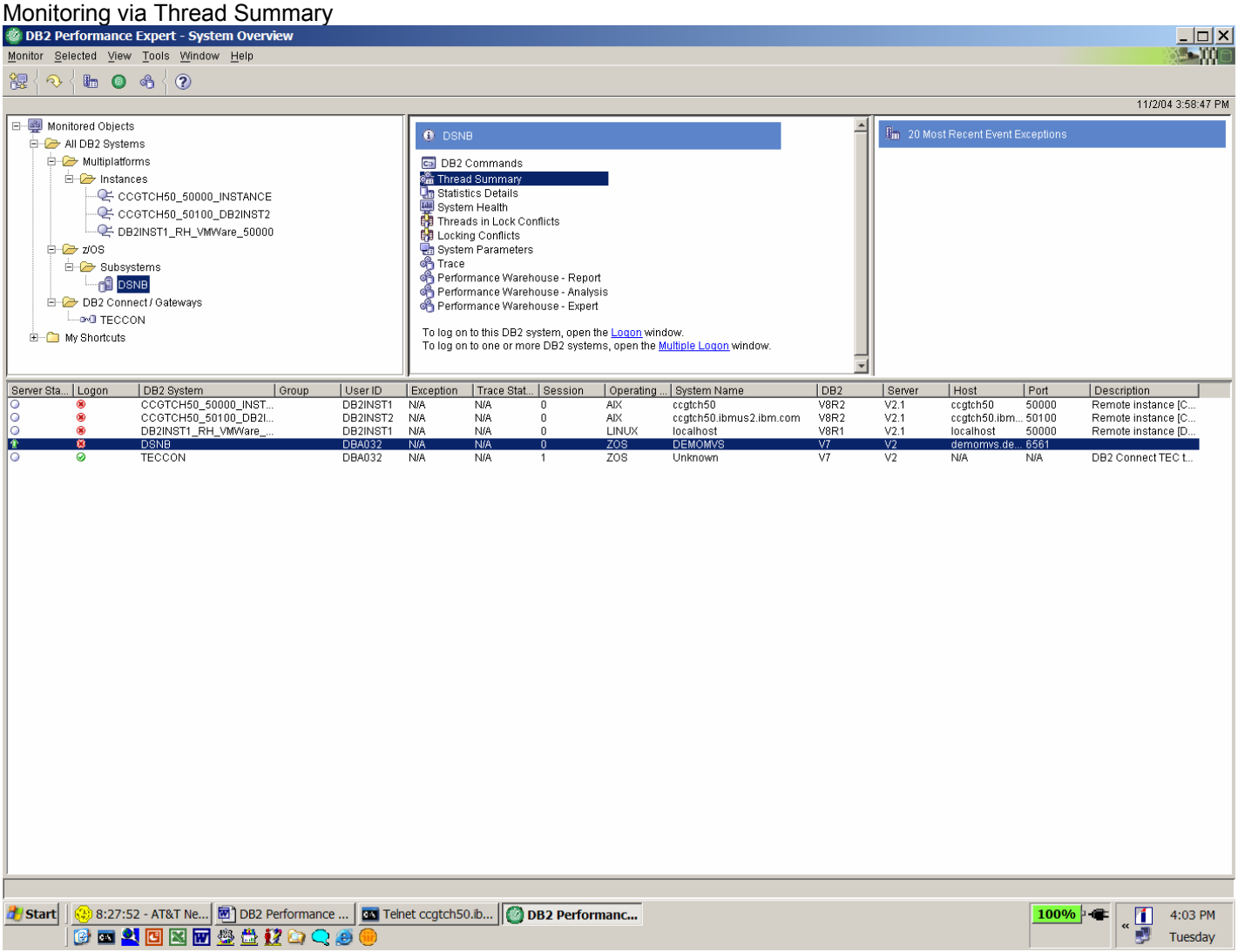
Average response size outbound
Shows the number of bytes received by the DB2 Connect gateway from the client, excluding communication protocol overhead, divided by the number of SQL chains being transferred.

Network time
Shows the difference between the value for the Times for sample SQL statement - Total statement time counter and the value of the Times for sample SQL statement - Time on DB2 host counter. Both counters are on the Performance page.

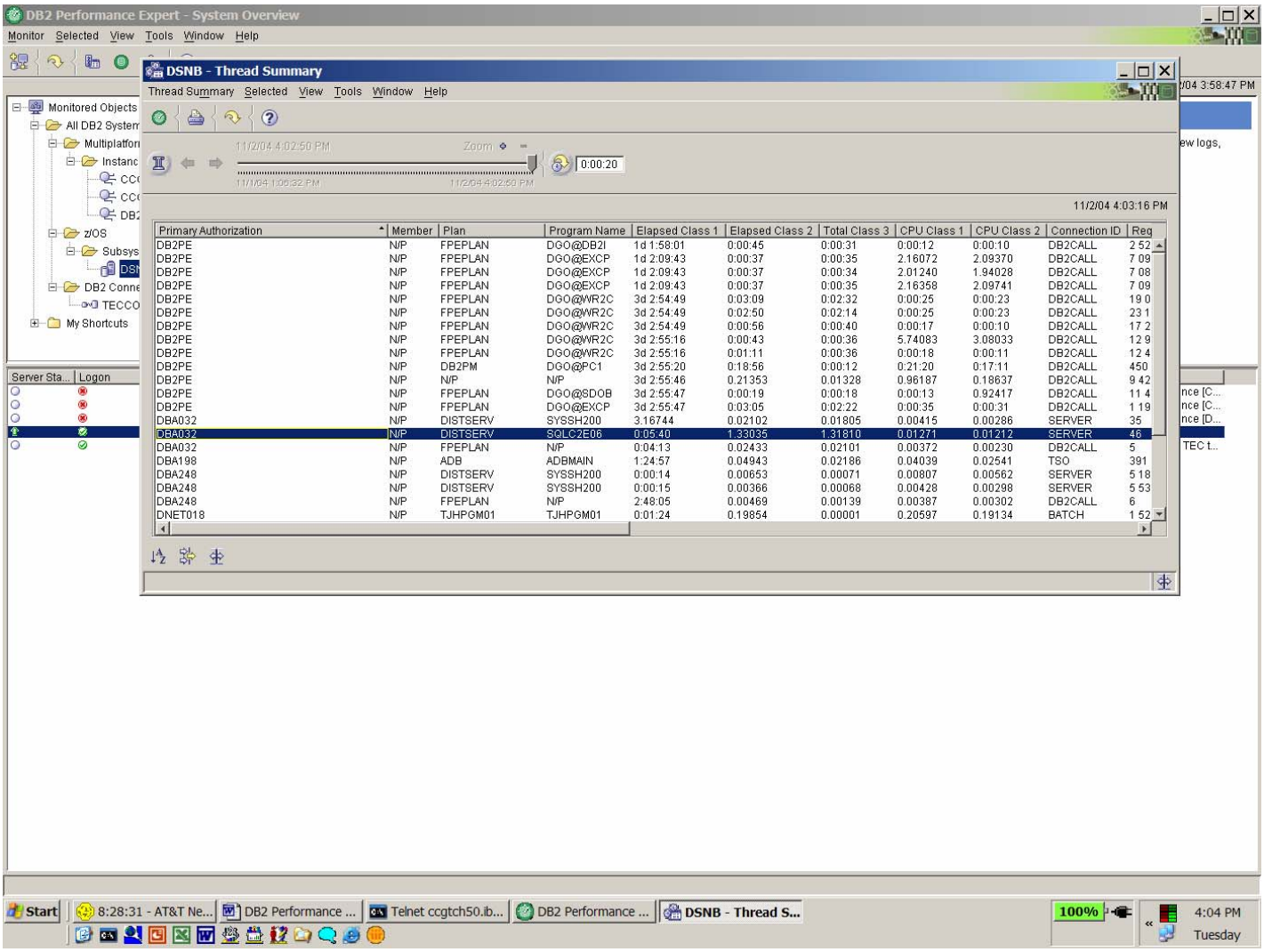
Parent DB name
Shows the name of the remote database to which the remote application is connected.

Statement group
Shows the number of statements with outbound bytes

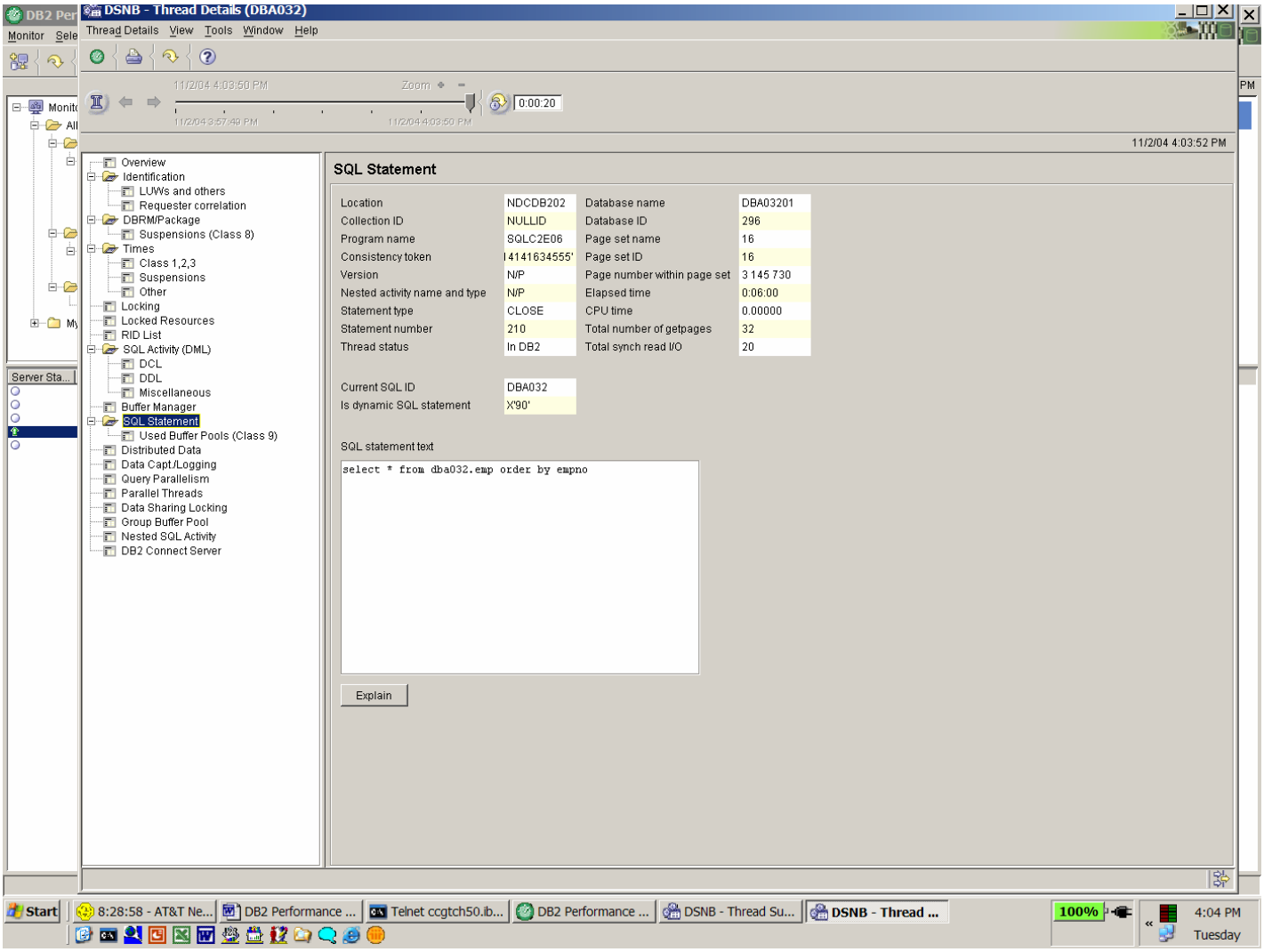
Monitoring via Thread Summary



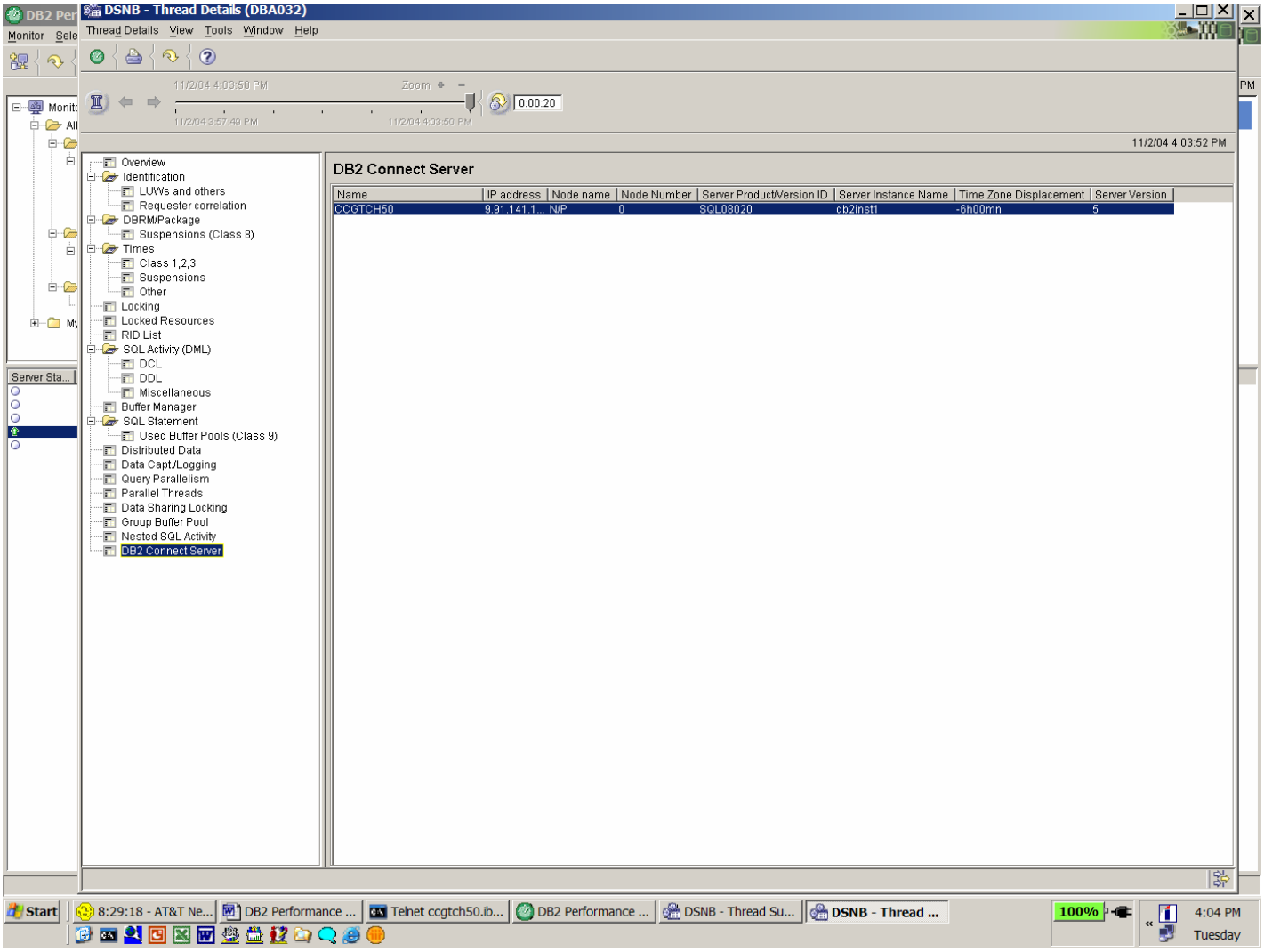
Information regarding distributed applications is also displayed via normal DB2 subsystem object monitoring.



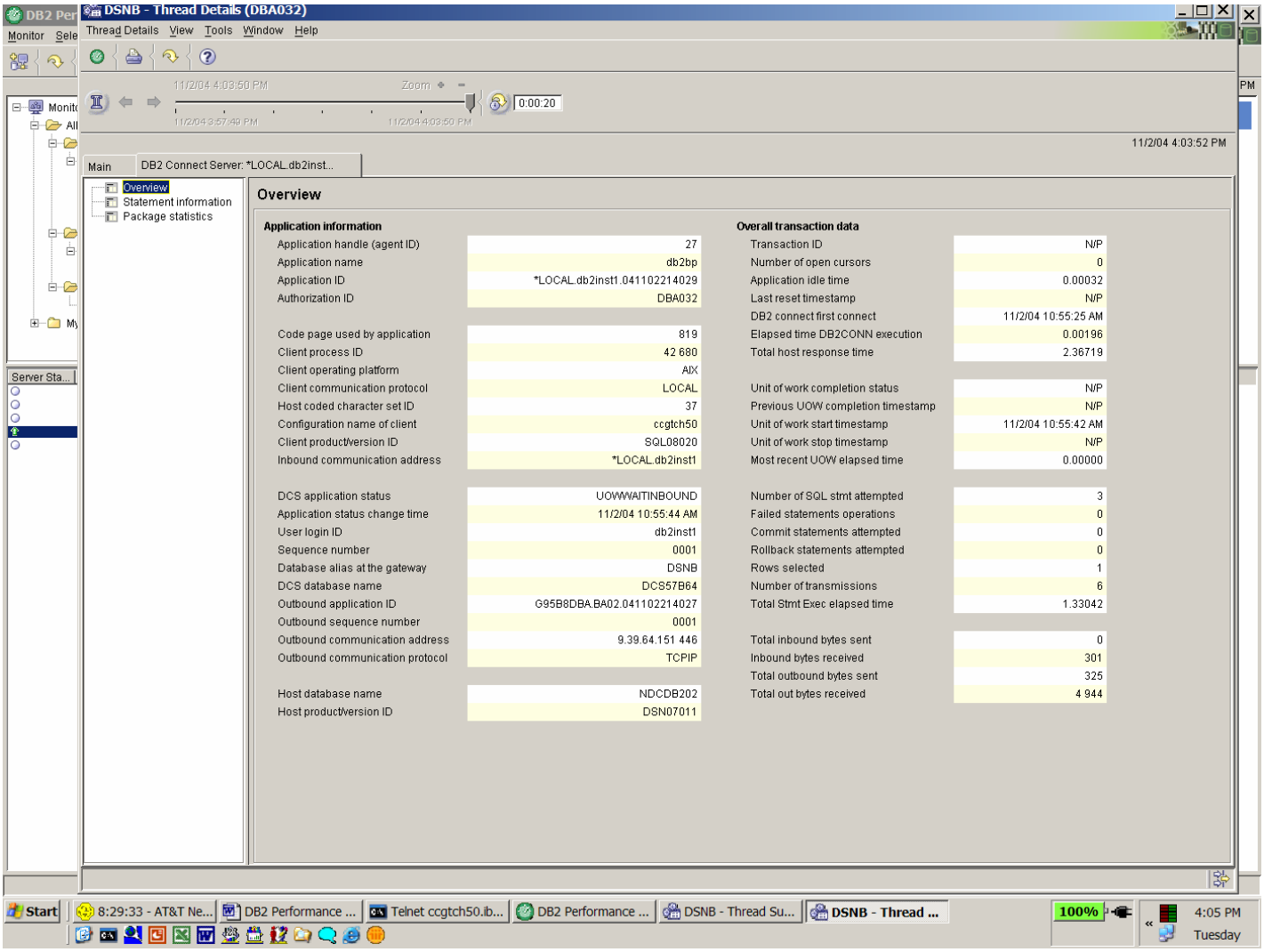
Distributed threads are identified via the DISTSERV plan.



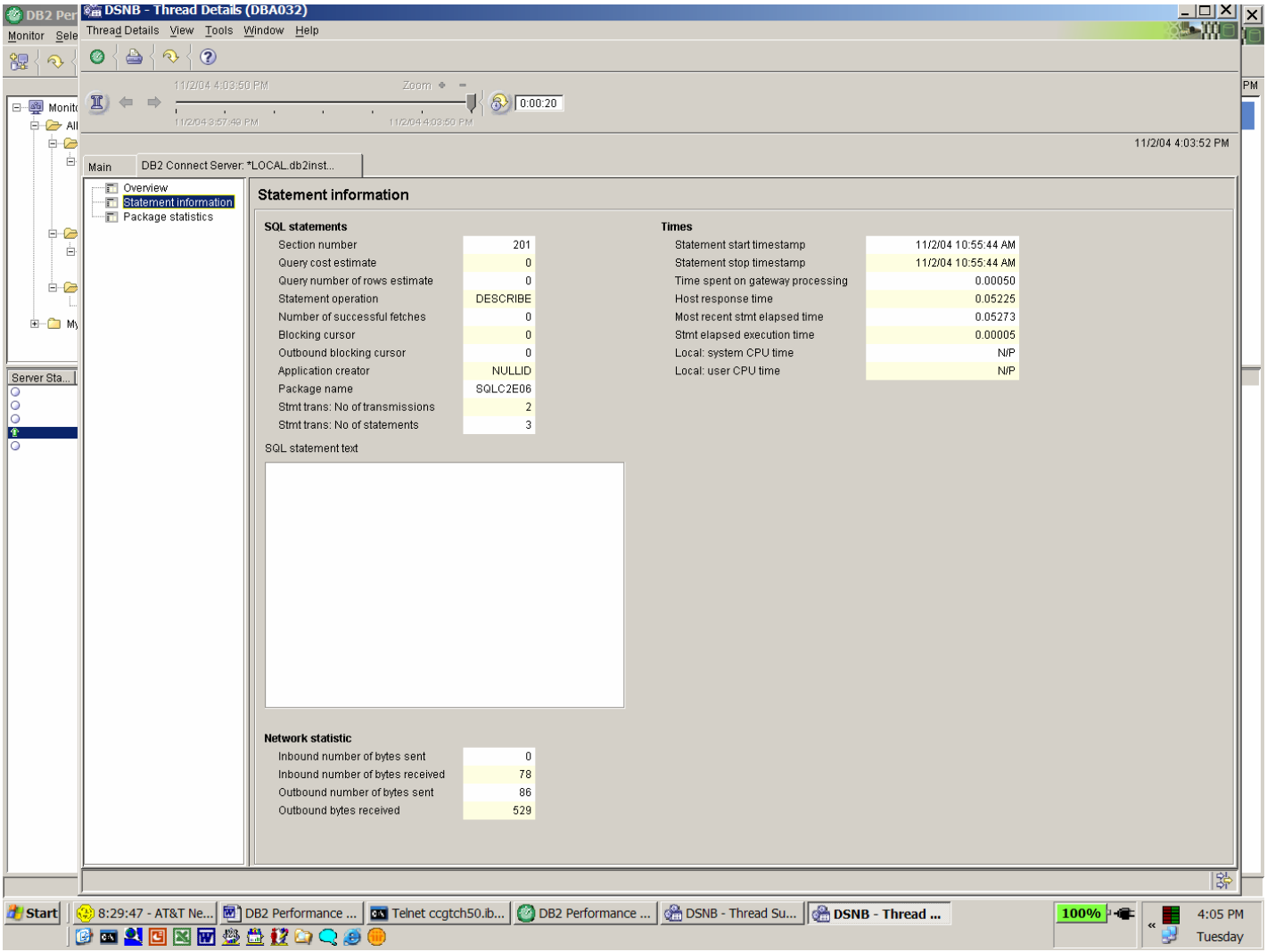
The SQL statement executing at the time of the snapshot is displayed in the Thread Details screen. The statement can be explained via the Explain button. As of November 2004, DB2 Visual Explain V7 can be launched via the button, but DB2 Visual Explain V8 can not.



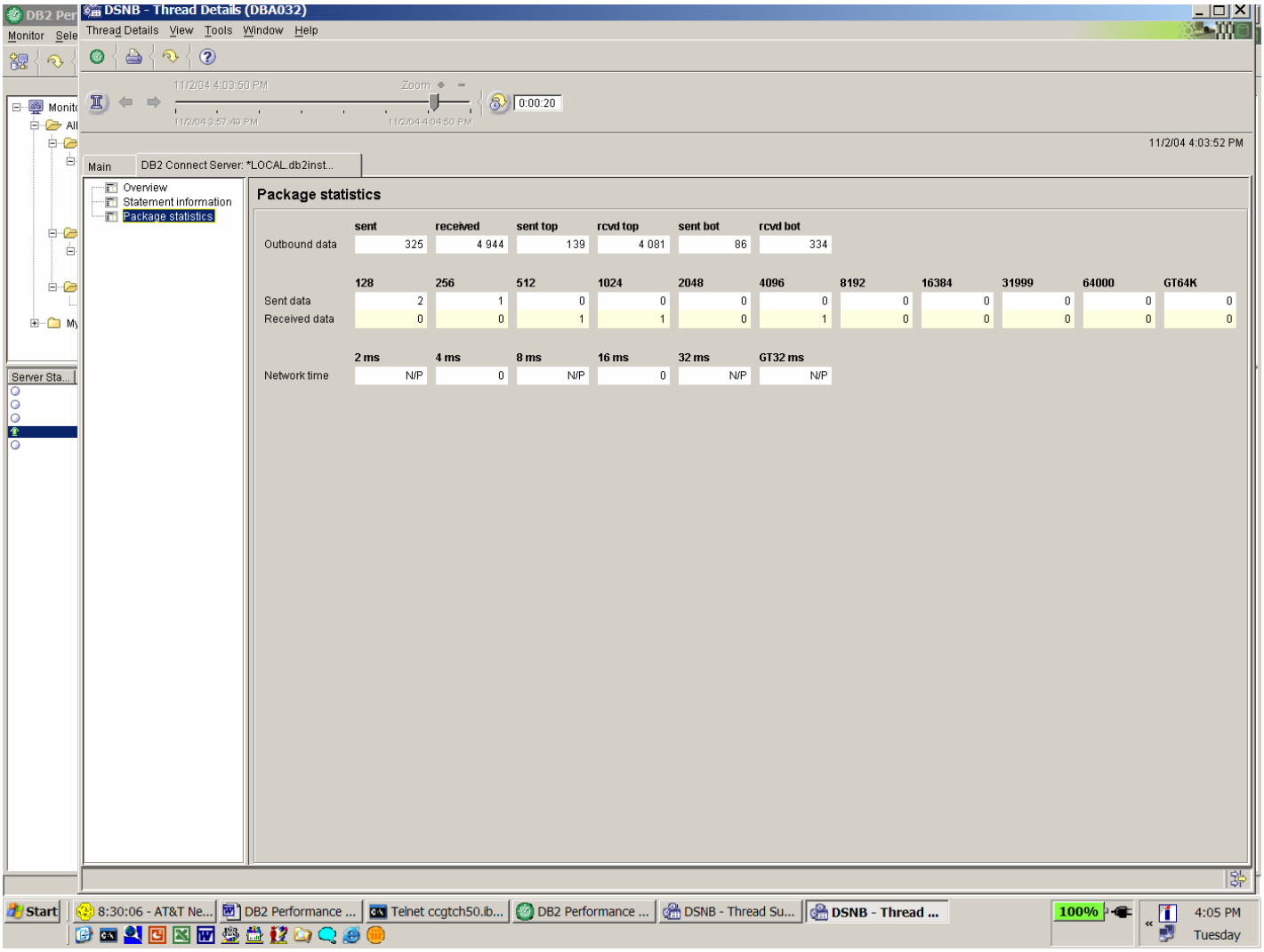
Also on the Thread Details screen, the DCS information that pertains to this thread is displayed. Drill-down is available to obtain more information.



This is the same information that can be accessed via the application details screen of the DB2 Connect gateway object monitoring.

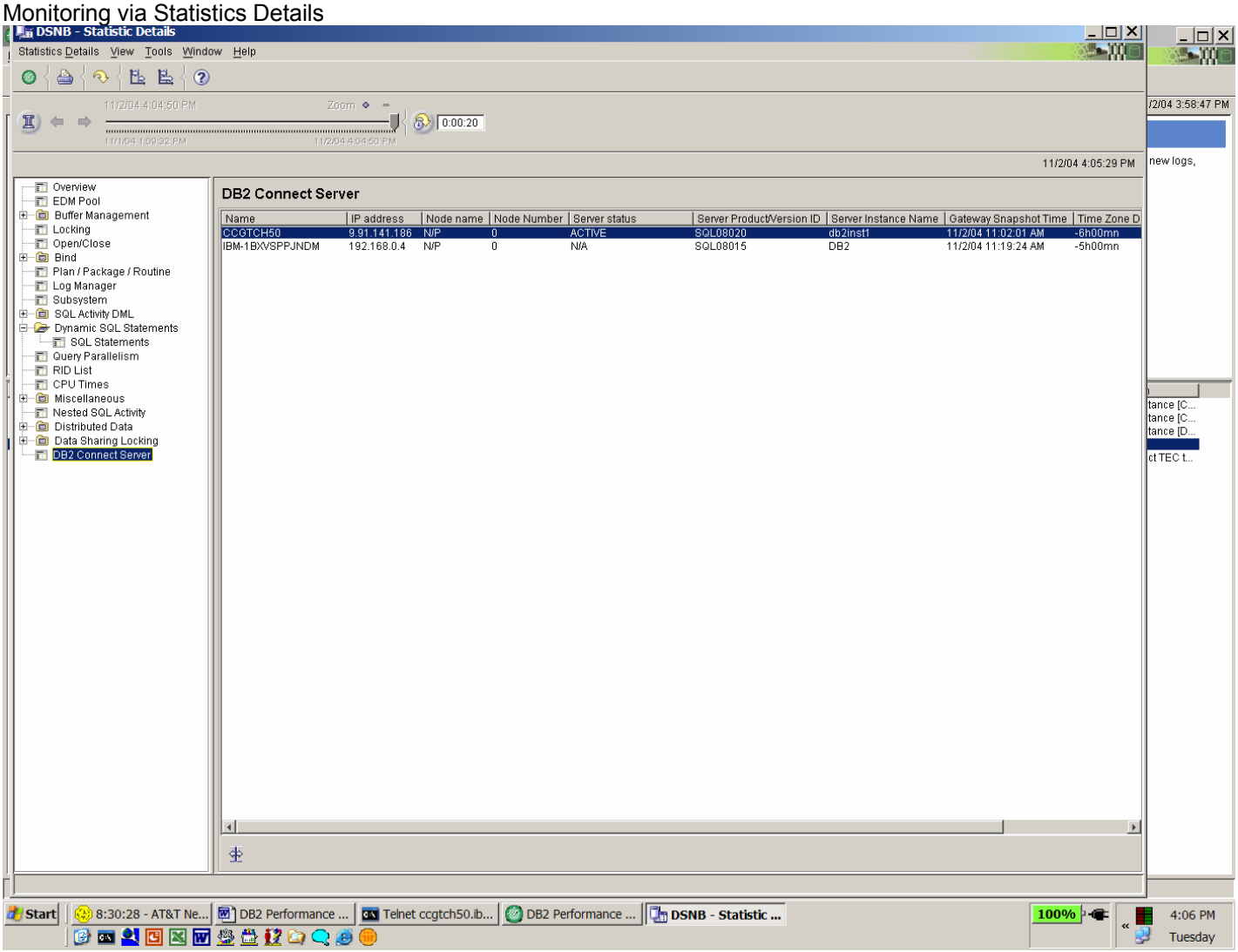


This is the same information that can be accessed via the application details screen of the DB2 Connect gateway object monitoring. The difference is the SQL statement text window. According to product development, this window may be removed from this screen in a future release. It rarely contains the statement – if ever.

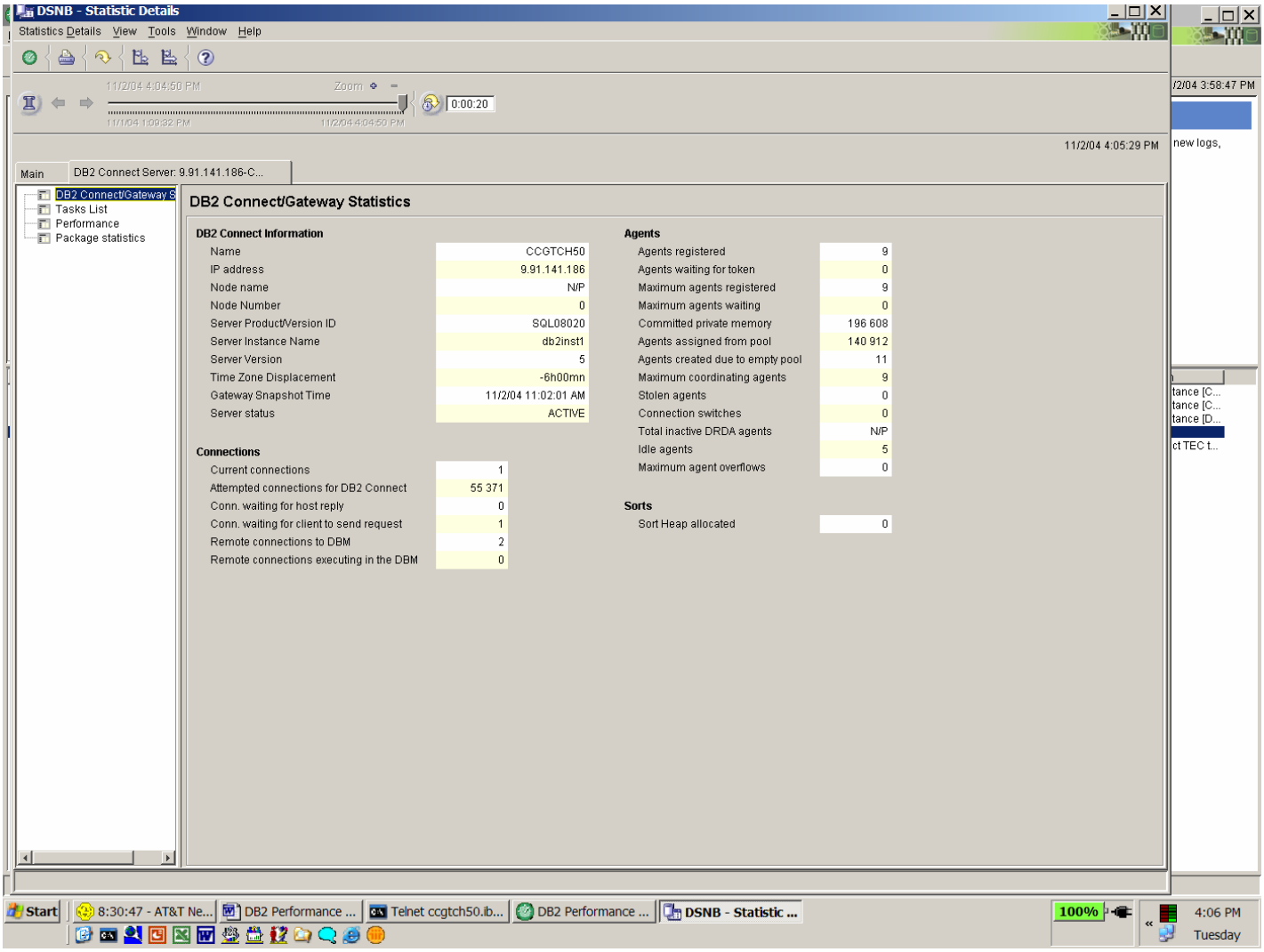


This is the same information that can be accessed via the application details screen of the DB2 Connect gateway object monitoring.

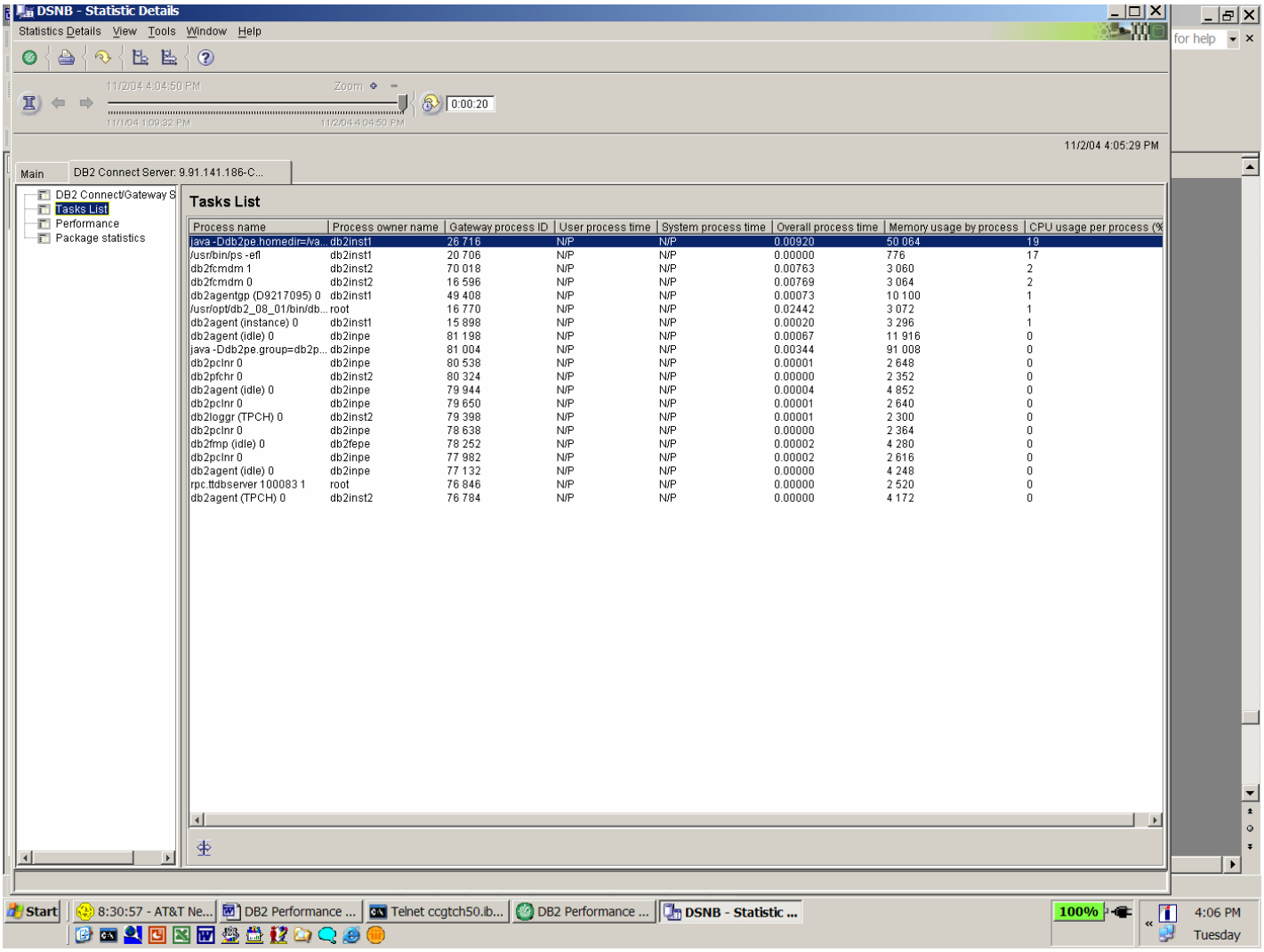
Monitoring via Statistics Details



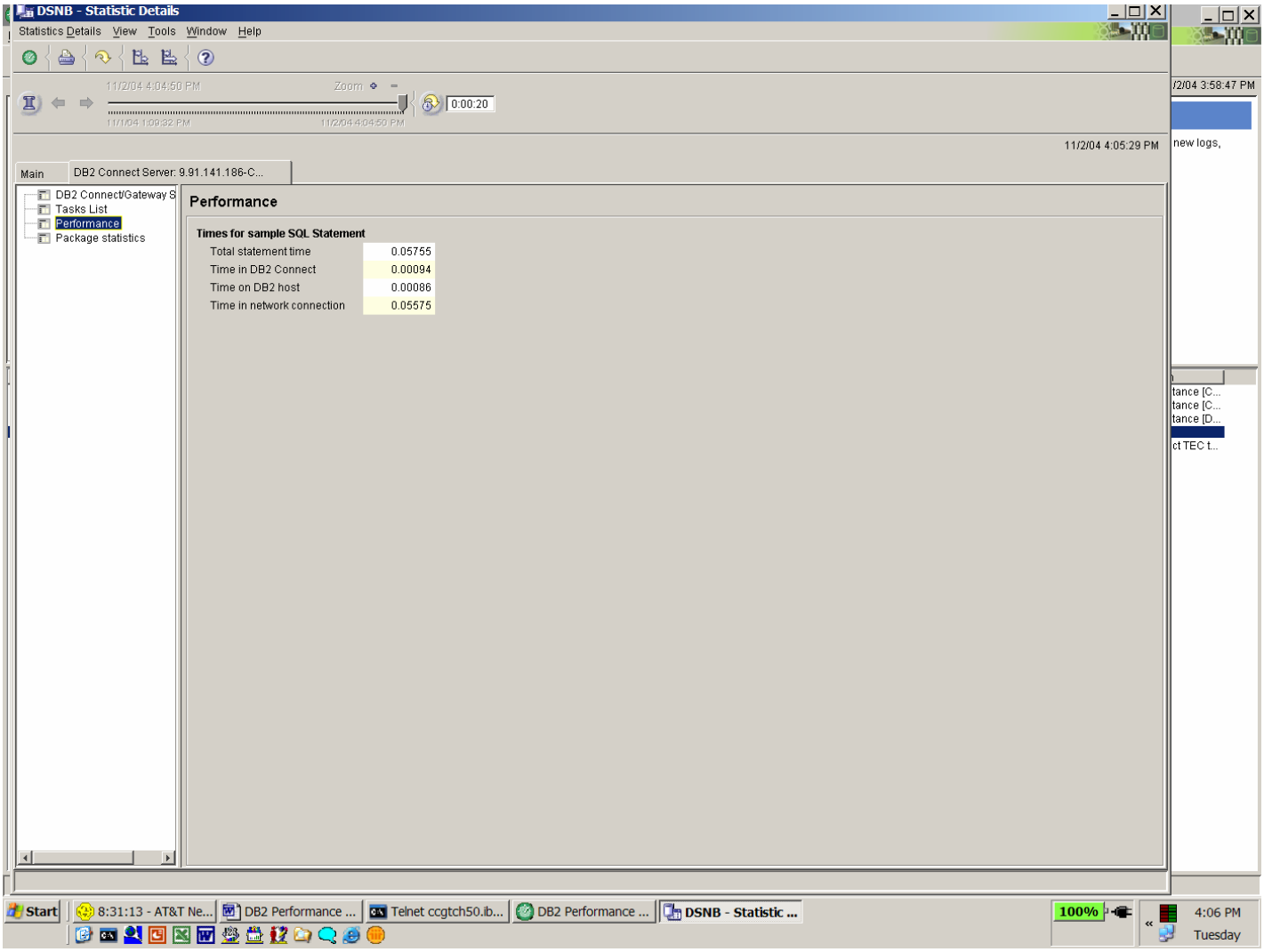
The final way to access DB2 Connect gateway information is via the subsystem Statistic Details screens. The last option, DB2 Connect Server, displays all of the DB2 Connect Servers that are connected to the subsystem.



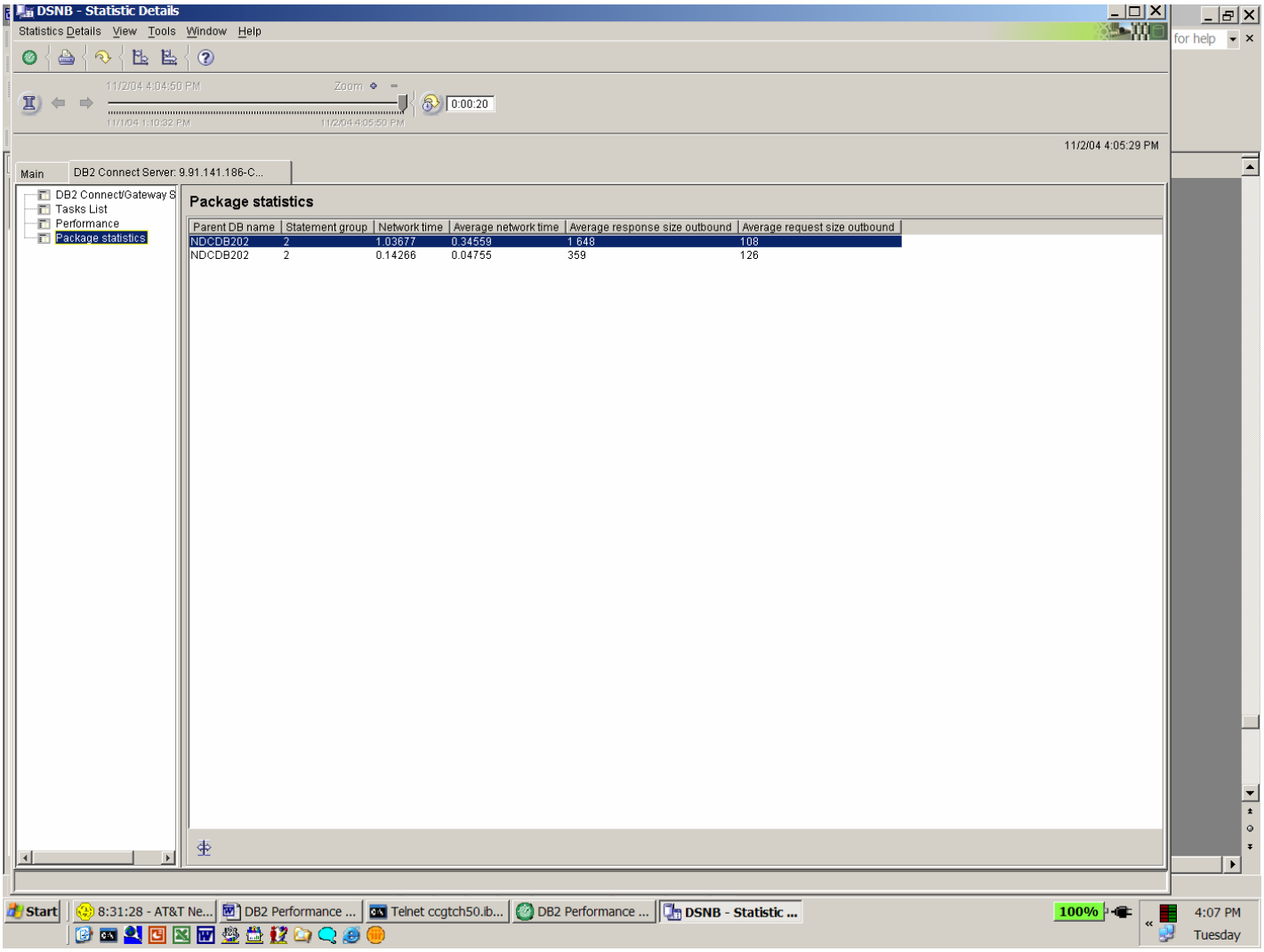
This is the same information that can be accessed via the statistics screen of the connect gateway object monitoring.



This is the same information that can be accessed via the statistics screen of the connect gateway object monitoring.



This is the same information that can be accessed via the statistics screen of the connect gateway object monitoring.



This is the same information that can be accessed via the statistics screen of the connect gateway object monitoring.

Practical Usage

These usage tips are documented in detail in the IBM DB2 Performance Expert for z/OS Version 2 Redbook (see Resources).

Connection Pooling – If there is a high number in the “Agents created due to empty pool” field, you may want to adjust NUM_POOLAGENTS. This number should be less than 2% of MAXAGENTS. A high number in the “Agents assigned from the pool” shows that connection pooling is providing some benefit in your system. Samples of this output are located on pages 9 and 22.

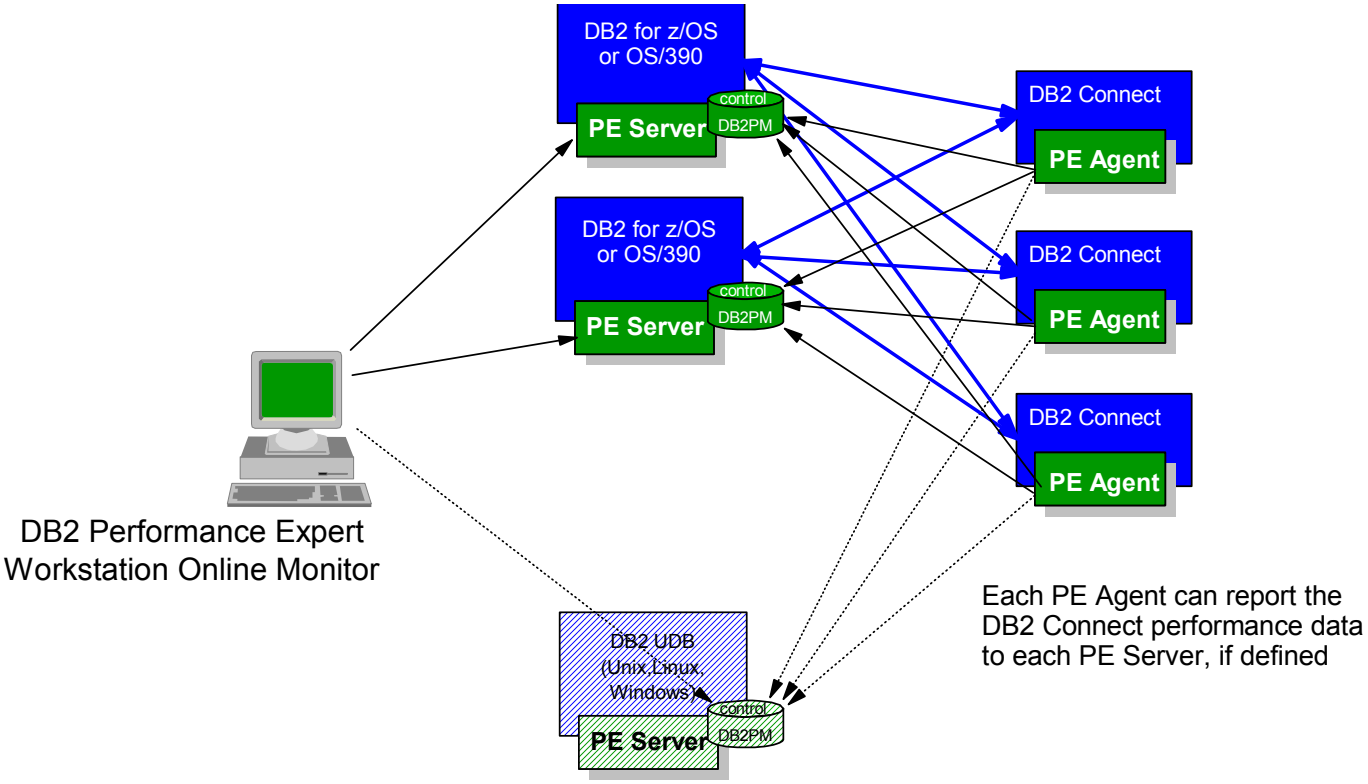
Busy Connections – An increasing number in “Connections waiting for host reply” may indicate the need to increase MAXAGENTS.

CPU and Working set size – CPU for DB2 Connect agents on a server should be less than 25%.

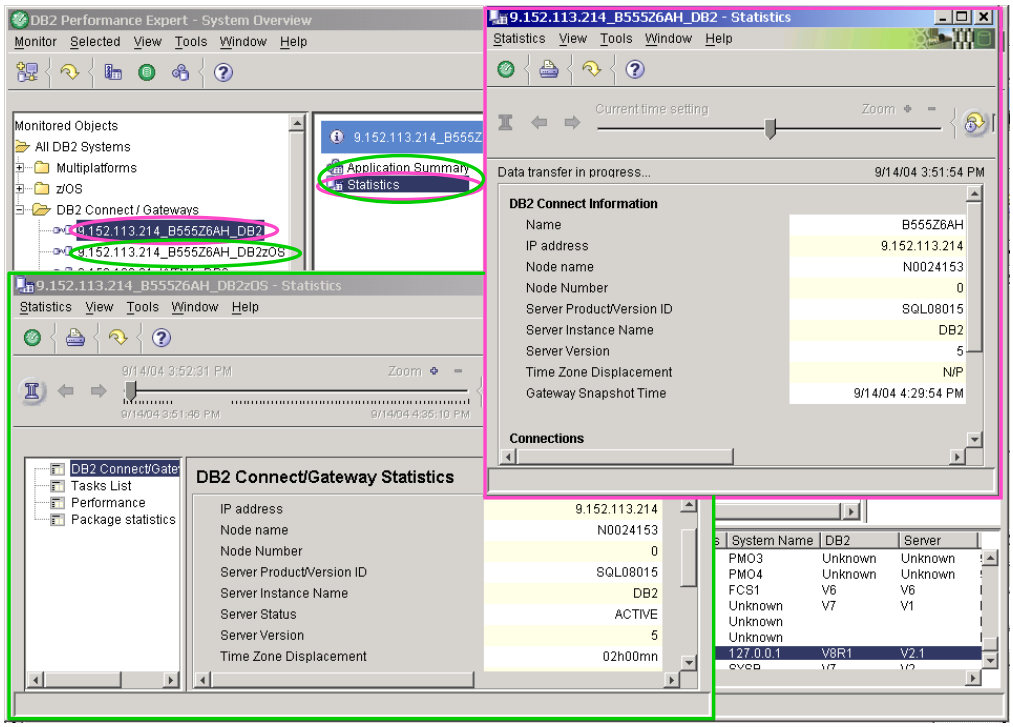
Response Time – Utilize the Sample SQL Statement information to determine the effect each part of the solution is having on total performance.

DB2 Performance Expert for z/OS versus Multiplatforms

What is the difference in support between DB2 PE for z/OS and DB2 PE for MP?



DB2 PE for MP (pink outline) allows only monitoring of general DB2 Connect Statistics data, but not any application related data. Also, there is no correlation to the DB2 for z/OS performance data.



Resources

IBM DB2 Performance Expert for z/OS Version 2 at http://www.redbooks.ibm.com/redbooks/pdfs/sg246867.pdf
Chapter 8, DB2 Connect Monitoring contains an overview of DB2 Connect Monitoring, Essential Components, Command Line Interface, Statistics Details, Monitoring from Thread Details

Monitoring Performance from the Workstation at http://publib.boulder.ibm.com/epubs/pdf/fpempb12.pdf
Chapter 8, Monitoring activities of DB2 Connect gateways and connections of DCS applications

Installation and Configuration at http://publib.boulder.ibm.com/epubs/pdf/fpeinb12.pdf
Chapters regarding installing the agent on Unix and Windows

Norbert Jenninger and Ernie Mancill presentations provided some screen shots and setup advice – see the DB2 and IMS Tools and Engine Team Room.