IBM Tivoli Network Manager

Best Practices for Custom Distributed Discovery

Abstract
One of the requirements of a multi-tenancy environment is to be able to manage remote customer networks from a central location. And one of the challenges which come from such a deployment is the fact that the networks being discovered and polled are likely to be at the end of WAN links. This document describes how you can distribute ITNM core processes so that discovery and monitoring can execute locally on the remote network. It compares this custom distributed architecture to the standard architectures.

Scott McQuillan
Krishna Kodali
Rob Clark

Version 2.3
February, 2017

Contents

Overview

This paper describes three deployment architectures for ITNM to handle large geo-distributed networks, or help manage a Managed Services business with multiple customers behind firewalls.

The third example is a custom distributed architecture that has certain performance advantages. We will describe in detail the steps for setting up this custom architecture.

This is an advanced procedure that is supported by IBM, and as such it is expected that the reader have some familiarity with Netcool and ITNM in particular.

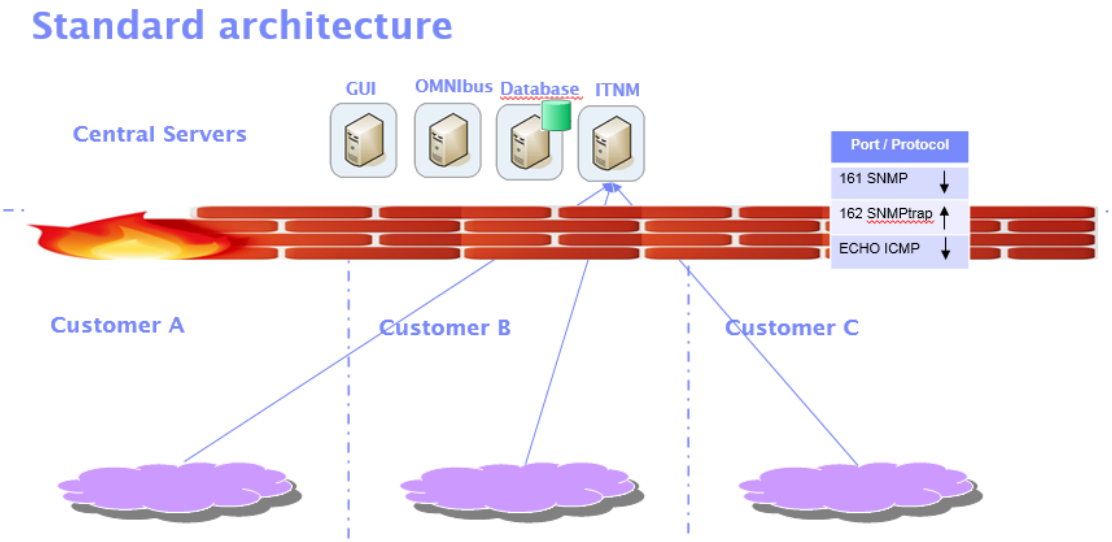
Architecture options

Standard central architecture

The standard recommended architecture for scaling to large networks involves a central deployment of ITNM core on one or more servers and partitioning the network into multiple domains. Each domain is polled from the central servers and supports domains that represent overlapping IP address spaces, including the use of dedicated NICs per domain to facilitate routing issues.

There are two obstacles that limit this approach. Firewalls guarding the central system that require opening the ECHO PING and SNMP ports for remote networks, increasing security risks. If the remote networks are accessed over WAN links, such as VPNs, then the network latency may start to affect the discovery times.

At some point the discovery time, or IT traffic, becomes unacceptable, or the firewall risks are not tolerated, and another approach is needed.



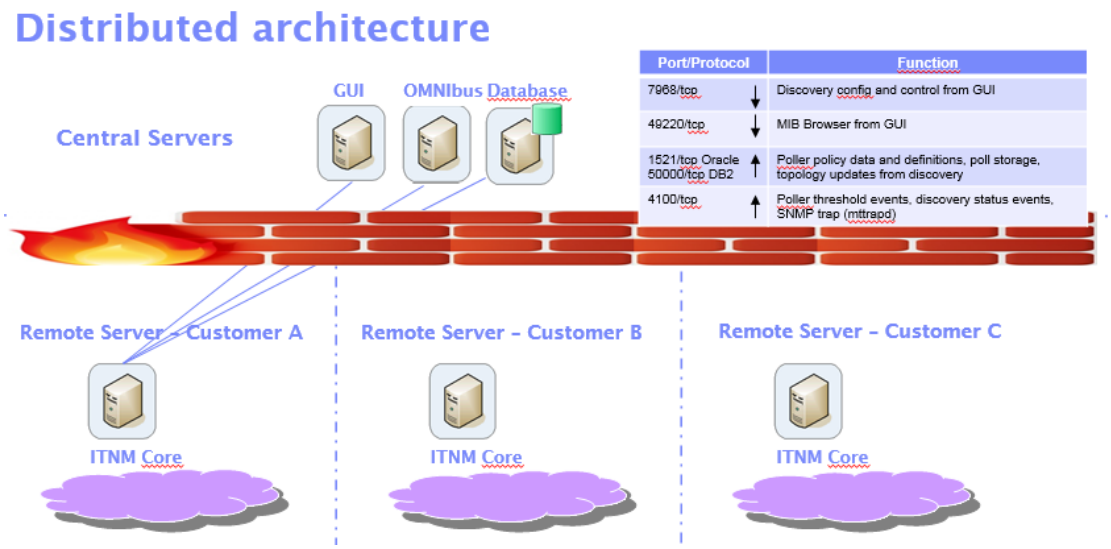
Standard distributed architecture

This deployment involves installing the complete ITNM core server on the remote client network resulting in local polling for discovery and monitoring with centralized data storage and GUI.

The GUI, OMNIbus, and NCIM database are installed at the central site.

Table of ports requiring firewall access:

1	7968/tcp	Discovery config and control from GUI
2	49220/tcp	MIB Browser from GUI (port number can be fixed)
3	1521/tcp Oracle 50000/tcp DB2	Poller policy data and definitions, poll storage
4	4100/tcp	Poller threshold events, discovery status events



Deployment, day-to-day administration, and upgrades is straightforward. All ITNM features are supported.

However, we know from experience with existing customers that having a WAN link between the ncp_model process and the NCIM database can result in very poor performance. This is caused by the high network latency encountered on each of the many thousands of SQL statements executed by Model against NCIM.

Custom distributed architecture

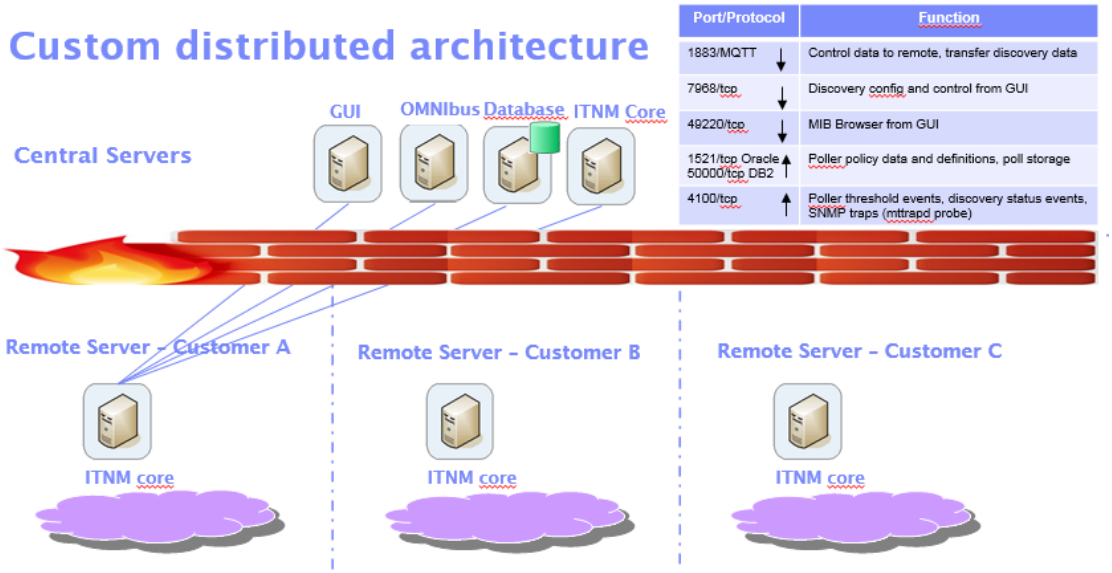
In order to support the discovery of customer sites over WAN links we look to ITNM's ability to run the discovery and monitoring processes remotely, and the Mod-

el and the ITNM Event gateway on a server on the central site with the NCIM database and GUI server.

The advantage of this approach is that all the device polling is done locally and the end result of the discovery, the topology, is transferred efficiently as a batch and merged into the central NCIM database by the Model process.

Table of ports requiring firewall access:

1	1883/tcp	Control data to remote, transfer discovery data
2	7968/tcp	Discovery config and control from GUI
3	49220/tcp	MIB Browser from GUI (the port number can be fixed)
4	1521/tcp Oracle 50000/tcp DB2	Poller policy data and definitions, poll storage
5	4100/tcp	Poller threshold events, discovery status events



The above figure illustrates a distributed deployment of ITNM in which the majority of the ITNM processes run on a server at the remote end of the WAN link and the Model and Gateway processes run on the Central server at the other end of the WAN link.

Setup for the Custom Distributed Architecture

This diagram offers an overview of the architecture showing the traffic across the firewall.

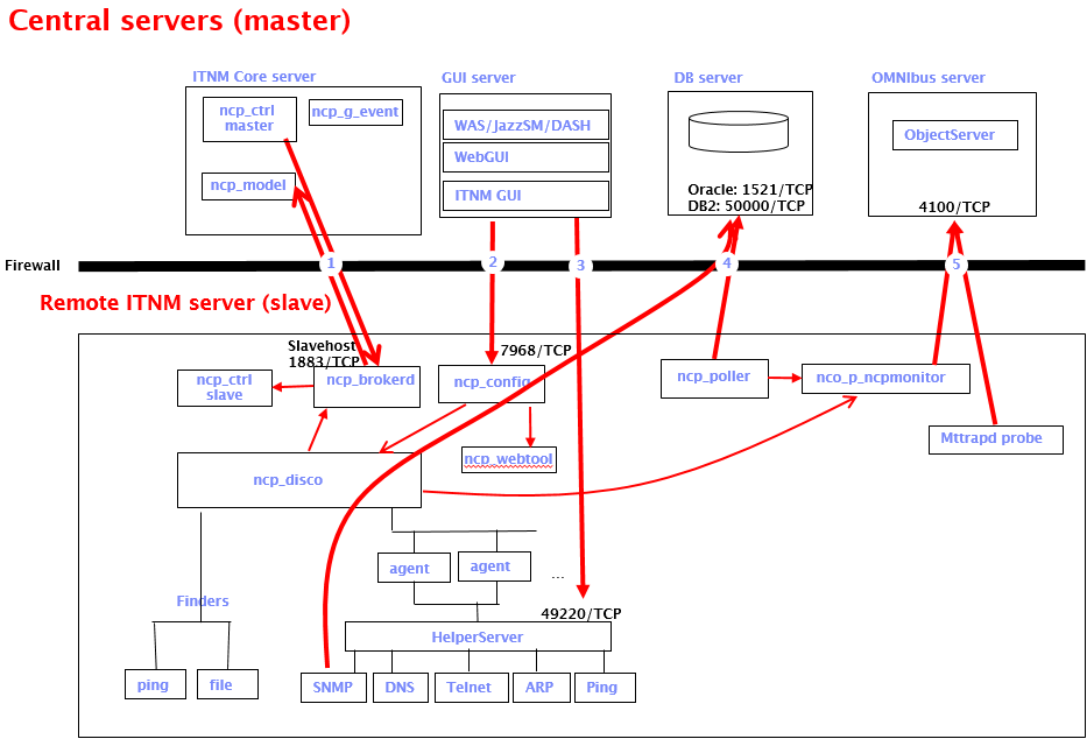


Table of ports requiring firewall access:

	Port/Protocol	Listening	Function
1	1883/tcp	On remote	Control data to remote, transfer discovery data
2	7968/tcp	On remote	Discovery config and control from GUI
3	49220/tcp	On remote	MIB Browser from GUI (port number can be fixed)
4	1521/tcp Oracle 50000/tcp DB2	On central	Poller policy data and definitions, poll storage
5	4100/tcp	On central	Poller threshold events, discovery status events

Installations

To ensure proper expectations for DNS resolution, add entries to the /etc/hosts file for each of the netcool servers, which includes the IP address, FQHN, and short name. For example,

```
127.0.0.1   localhost localhost.localdomain
10.10.10.1  centralserver.com centralserver
10.100.1.1  remoteserver.com remoteserver
```

It is important that the remote short names must also match the Unix hostname command.

1. The entire ITNM core server should be installed just as with any standard ITNM installation.
   Connect to the central ObjectServer and the central database as usual so that the installer performs the ITNM configuration for the ObjectServer, database schemas, and the first ITNM domain.

2. After installing the ITNM Core, you can install ITNM GUI on the GUI server as usual. There are no specific post-install steps required for the GUI in this scenario.

3. On the ITNM Remote server the entire ITNM core server should be installed just as with any standard ITNM installation. Connect to the same ObjectServer and database you used in step 1. Ensure the domain you wish to use for this remote site has been created on the central server, or let the installer create the first domain for you.

At this point, continue with the post-install steps to get the first remote server working, including a first discovery, before repeating the steps to install and set up the next remote server.

Post-install steps on the Central Server

All files that are domain specific are indicated here and must be created if they don't exist.

1. Configuring the message bus

File: `$NCHOME/etc/precision/Precision.broker.`*`domain`*`.cfg`

The message broker is used for internal communications between ITNM processes. The message brokers will run, one per domain, on each remote server for efficiency to keep the bulk of the traffic local on the remote.

Each instance of this file on the central service must be configured so the local processes can find the remote broker process.

(By default the message broker process will bind itself to the loop back address and will therefore not allow remote processes to connect to it.)

Edit the file to specify the remote server and port:

```
Service: 1883
Network: IP address
```

This file should be identical on the remote server. This is documented here:

https://www.ibm.com/support/knowledgecenter/SSSHRK_4.2.0/itnm/ip/wip/admin/task/nmip_adm_changinghostandportsettingsforrsmb.html

2. Configuring where the ITNM processes will run

File: `$NCHOME/etc/precision/CtrlServices.domain.cfg`

We will choose the central ITNM server to be the MASTER and the remote server the SLAVE.

The CtrlServices configuration file should be modified such that all the processes are executed on the remote server except,

- `ncp_model`
- `ncp_g_event`

Add the remote hostname for each process intended to be run remotely, similar to this example for the ncp_disco process:

```
insert into services.inTray
(
    serviceName,
    binaryName,
    servicePath,
    domainName,
    hostName,
    argList,
    dependsOn,
    retryCount
)
values
(
    "ncp_disco",
    "ncp_disco",
    "$PRECISION_HOME/platform/$PLATFORM/bin",
    "$PRECISION_DOMAIN",
    "remoteserver",
    [ "-domain" , "$PRECISION_DOMAIN", "-discoOnStartup", "0" ,
"-latency" , "100000" , "-debug", "0", "-messagelevel",
"warn"],
    [ "ncp_d_helpserv", "ncp_model" ],
    5
);
```

Note: It is important that the hostname is the same that is returned by the Unix command `hostname` on the remote server. If it does not match exactly then the slave ncp_ctrl will not start the process.

This is documented here:

http://www.ibm.com/support/knowledgecenter/SSSHRK_4.2.0/itnm/ip/wip/install/task/nmip_ins_settingupaslavencpctrlprocess.html

3. Configuring the ncp_config port

File: `$NCHOME/etc/precision/ModelNcimDb.domain.cfg`

Model is responsible for setting some domain attributes in the `ncim.domainMgr` table for the GUI and other processes to use. The GUI uses the `domainHost` and `domainPort` from the `domainMgr` table in NCIM to provide it with OQL access for the discovery status and configuration. The GUI topology views use `webT-opDataSource` as the datasource name to access the objectserver. These fields are populated by ncp_model. Therefore you must modify the ModelNcimDb configuration file to provide the host and port of the data collection server as follows:

```
insert into dbModel.access
(
    EnumGroupFilter,
    TransactionLength,
    ValidateCacheFile,
    WebTopDataSource,
    DomainHost,
    DomainPort
)
values
(
    "enumGroup in ('ifAdminStatus', 'ifOperStatus', 'sysServices',
'ifType',
    'cefcFRUPowerAdminStatus', 'cefcFRUPowerOperStatus', 'TruthValue',
    'entSensorType', 'entSensorScale', 'entSensorStatus',
    'cefcModuleAdminStatus', 'cefcModuleOperStatus', 'ipForwarding',
    'cefcPowerRedundancyMode', 'EntityType', 'ospfIfState', 'ospfIfType',
    'dot3StatsDuplexStatus', 'accessProtocol')",
    500,
    0,
    "datasource",
    "remoteserver.com",
     7968
);
```

The WebTopDataSource is the datasource name used by the GUI for topology status. Check that this name exists in `/opt/IBM/netcool/gui/omnibus_webgui/etc/datasources/ncwDataSourceDefinitions.xml`
You will probably be using the same ObjectServer for the events for all the domains, so you can use the same datasource name for all domains.

Important: Disco (`ncp_disco`) also reads this file for DNCIM, but ignores the `DomainHost` and `DomainPort`. After any changes you make to this file, copy it to the remote server.

4. Configuring the fixed ports

File: `$NCHOME/etc/precision/ServiceData.cfg`

Check that the **ncp_config** service entry is created correctly after the processes are running for each domain with the correct port number,

`SERVICE: ncp_config DOMAIN: domain ADDRESS: remote IP PORT: 7968 SERVERNAME: remote host DYNAMIC: NO`

To fix the **Helper** service port for the MIB Browser GUI to use, you must create this entry and make sure it is set to DYNAMIC: NO. Select a consistent port number across all domains to make administration simpler,

```
SERVICE: Helper DOMAIN: domain ADDRESS: remote IP
PORT: 49220 SERVERNAME: remote host DYNAMIC: NO
```

Note: that multicast port 33000 is not used across the firewall.

Important: `ServiceData.cfg` is not a domain specify type of file – this file contains entries for all domains. For more on this, see,
https://developer.ibm.com/answers/questions/295678/domain-specific-servicedatacfg/

5. Changes are not normally required for these files

- File: `$NCHOME/etc/precision/DbLogins.domain.cfg`
  The installer will set all entries to the central database specified during the installation. The DNCIM database is not used on the central server and can be ignored.

- File: `$NCHOME/etc/precision/MibDbLogin.domain.cfg`
  The GUI will normally use the central database for the ncmib schema.

Post-install steps on the Remote Server

The following steps must be carried out on the remote ITNM server. All files that are domain specific are indicated here and must be created if they don't exist.

1. Set up the message bus

File: `$NCHOME/etc/precision/Precision.broker.domain.cfg`

Edit the file to specify the remote server and port (identical to the central server):

```
Service: 1883
Network: remoteserver.com or IP address
```

All the ITNM processes will use this broker to communicate with each other, regardless of where they are running. This is documented here:

https://www.ibm.com/support/knowledgecenter/SSSHRK_4.2.0/itnm/ip/wip/admin/task/nmip_adm_changinghostandportsettingsforrsmb.html

Note: If the ITNM server is to run multiple domains then an additional step is required. This is to create a domain specific version of the broker configuration file (e.g. Precision.broker.NCOMS.cfg)

2. Configuring the slave ncp_ctrl

File: `$NCHOME/etc/precision/CtrlServices.domain.cfg`

Once the ncp_ctrl process is started in slave mode on the remote, it will get its instructions on what processes to run from the master ncp_ctrl process running on the central server.

Therefore, the CtrlServices domain specific configuration file should be modified such that it exists but is empty. Put a comment in there to the effect that the file is deliberately empty.

We will change the `ncp_control.sh` file in a later step so that the `itnm_start` command will start it in slave mode.

3. Configuring the fixed ports

File: `$NCHOME/etc/precision/ServiceData.cfg`

The **ncp_config** service should be created automatically and should match the port set in ModelNcimDb.*domain*.cfg on the central server. Check that this entry for ncp_config is correct:

SERVICE: ncp_config DOMAIN: *domain ADDRESS: remote IP* PORT: 7968
SERVERNAME: *remoteserver* DYNAMIC: NO

To fix the **Helper** service port for the MIB Browser GUI to use, you must create this entry and make sure it is set to DYNAMIC: NO. Select a consistent port number across all domains to make administration simpler.

SERVICE: Helper DOMAIN: *domain ADDRESS: remote IP* PORT: 49220
SERVERNAME: *remoteserver* DYNAMIC: NO

4. Model to NCIM changes

File: `$NCHOME/etc/precision/ModelNcimDb.`*`domain`*`.cfg`

Model does not run on the remote server, but `ncp_disco` does and it reads this file for the DNCIM modelling. Therefore ensure this file is identical to the one on the Central server. (The `DomainHost` and `DomainPort` will be ignored by `ncp_disco`, so you can simply copy the file from the Central server.)

5. Changes are not normally required for these files

- File: `$NCHOME/etc/precision/DbLogins.`*`domain`*`.cfg`
  No change. All the database schemas will point to the central DB set up from the installation, but used only by the Poller.
  The DNCIM is local, but again, should be set correctly by the installer.

- File: `$NCHOME/etc/precision/MibDbLogin.`*`domain`*`.cfg`
  No change. The ncmib schema should point to the central DB set up from the installation.

6. Set hostname for disco managed processes - finders

File: `$NCHOME/etc/precision/DiscoConfig.domain.cfg`

Steps 5, 6, and 7 must be done so that the processes under the control of ncp_disco will run on the remote machine. If this is not done, these processes will likely start on the Central server and thus not work for you.

Add the remote hostname for all the finder managedProcesses:

- ncp_df_ping
- ncp_df_collector
- ncp_df_file

Alter each the inserts to `disco.managedProcesses` to specify that the Finders should run on the remote host:

```
insert into disco.managedProcesses
(
    m_Host,
    m_Name // , m_Args, m_LogFile
)
values
(
    "remotehost.com",
    "ncp_df_ping" // , [ "-debug", "0" ],
'$NCHOME/log/precision/pingFinder.log'
);
```

7. Set hostname for disco managed processes - helpers

File: `$NCHOME/etc/precision/DiscoHelperServerSchema.domain.cfg`

Note: This file does not need to be modified in ITNM 4.2 Fix Pack 1 and later.

8. Set hostname for disco managed processes - agents

File: `$NCHOME/etc/precision/DiscoAgents.domain.cfg`

Add the remote hostname for all agents in this file:

Alter all the inserts to `disco.agents` to specify that the discovery Agents should run on the remote host, for example:

Note: This file demands the most edits. To make it easier, you could use a find and replace command to replace,

`m_MessageLevel,` with `m_MessageLevel, m_HostName,`

and

`"warn",` with `"warn", "remotehost.com",`

```
insert into disco.agents
(
    m_AgentName,
    m_Valid,
    m_ValidOnPartial,
    m_AgentClass,
    m_IsIndirect,
    m_Precedence,
    m_DebugLevel,
    m_MessageLevel,
    m_HostName,
    m_EndSignal,
    m_NumThreads
)
values
(
    "Details",
    1,
    1,
    0,
    0,
    1,
    0,
    "warn",
    "remotehost.com",
    2,
    100
);
```

9. Starting ITNM on the remote server

File: `$NCHOME/precision/bin/ncp_control.sh`

On the remote ITNM server running the slave `ncp_ctrl` process the ncp_control.sh script must be changed so that the `itnm_start` script can be used to start ncp_ctrl as the slave.

Modify the script to add the `-slave` argument as shown in this example,

```
${NCHOME}/precision/bin/ncp_ctrl -domain $PRECISION_DOMAIN \
  -slave \
  -logdir "${NCHOME}/log/precision" > \
  "${NCHOME}/log/precision/ncp_ctrl.${PRECISION_DOMAIN}.trace" 2>&1 &
```

Note: Do not leave any spaces after the backspace '\'

Starting the ITNM core processes

The ITNM core processes on the central server can be started/stopped as normal using the default "itnm_start" and "itnm_stop" scripts.

a) **First start the processes on remote server**
   This will ensure the message broker is running and will start the ncp_ctrl process which will be used by the central server to control the remote processes.
   Run,
   `itnm_start –domain <domain>`
b) **Then start the processes on the central server**
   Run,
   `itnm_start –domain <domain>`

The `itnm_status` output will show the domain and remote hostname it is running on. If no host shown, then it is running on the central server. This example output shows `removeserver` as the remote hostname.

```
#> itnm_status ncp
Network Manager:
  Domain:  AEROSPACE
      ncp_ctrl              RUNNING  PID=16618 AEROSPACE
      ncp_store             RUNNING  PID=30450 AEROSPACE host: removeserver
      ncp_class             RUNNING  PID=30452 AEROSPACE host: removeserver
      ncp_model             RUNNING  PID=17139 AEROSPACE
      ncp_disco             WAIT_HB  PID=30743 AEROSPACE host: removeserver
      ncp_d_helpserv        RUNNING  PID=30453 AEROSPACE host: removeserver
      ncp_config            RUNNING  PID=30456 AEROSPACE host: removeserver
      nco_p_ncpmonitor      WAITING  PID=30706 AEROSPACE host: removeserver
      ncp_g_event           WAITING  PID=      AEROSPACE
      ncp_webtool           RUNNING  PID=30471 AEROSPACE host: removeserver
      ncp_virtualdomain     WAITING  PID=      AEROSPACE host: removeserver
```

If everything is RUNNING, congratulations! If not, the next section can help you find what went wrong.

Troubleshooting

Check the firewall access

You may see the process running, but if the firewall is blocking the port then connections from the other side will fail.

To check remote access when the process is running, use a tool to test the port such as telnet (`telnet <ip addr> port`) or netcat (`nc –nv <ip addr> port`)

Run from	Port (examples used in this document)	Running process
Central to Remote	1883	ncp_broker
Central to Remote	7968	ncp_config
Central to Remote	49220	ncp_d_helpserv
Remote to Central	1521 (Oracle) 50000 (DB2)	Database
Remote to Central	4100	Object Server

Syntax errors

Check the log or trace for syntax errors you might have made.

Central server

```
ncp_ctrl.<domain>.log
ncp_model.<domain>.log
```

Remote server

```
ncp_brokerd_1883.log
ncp_disco.<domain>.log
```

itnm_start fails with an error for ncp_ctrl on remote
Common mistake is to leave a space after the backslash at the end of the line.

Processes in WAITING state

Run `itnm_status ncp` on the central server and you see all processes are in the WAITING state.

On the remote server, run,
ps –ef |grep ncp_ctrl

Check,

a) ncp_ctrl is running
b) `-slave` is included in the arguments
c) The hostName field in CtrlServices.cfg on the central server must match the output to the hostname command run on the remote server
d) The central server must have access to the ncp_broker port on the remote server – see the "Check the firewall access" paragraph above.

Missing process

Check all processes are running on the intended server. Run,

```
itnm_status ncp
```

If a process is running on the wrong server, check the CtrlServices.cfg on the central server.

These are the only ncp_* processes that should be running on the central server:

- ncp_ctrl
- ncp_model
- ncp_g_event

Do this again during discovery to make sure all the ncp_disco processes are running on the remote server:

- ncp_df_ping
- ncp_df_collector
- ncp_df_file

- ncp_agent –domain <domain> (…one per agent)

Alternative configuration

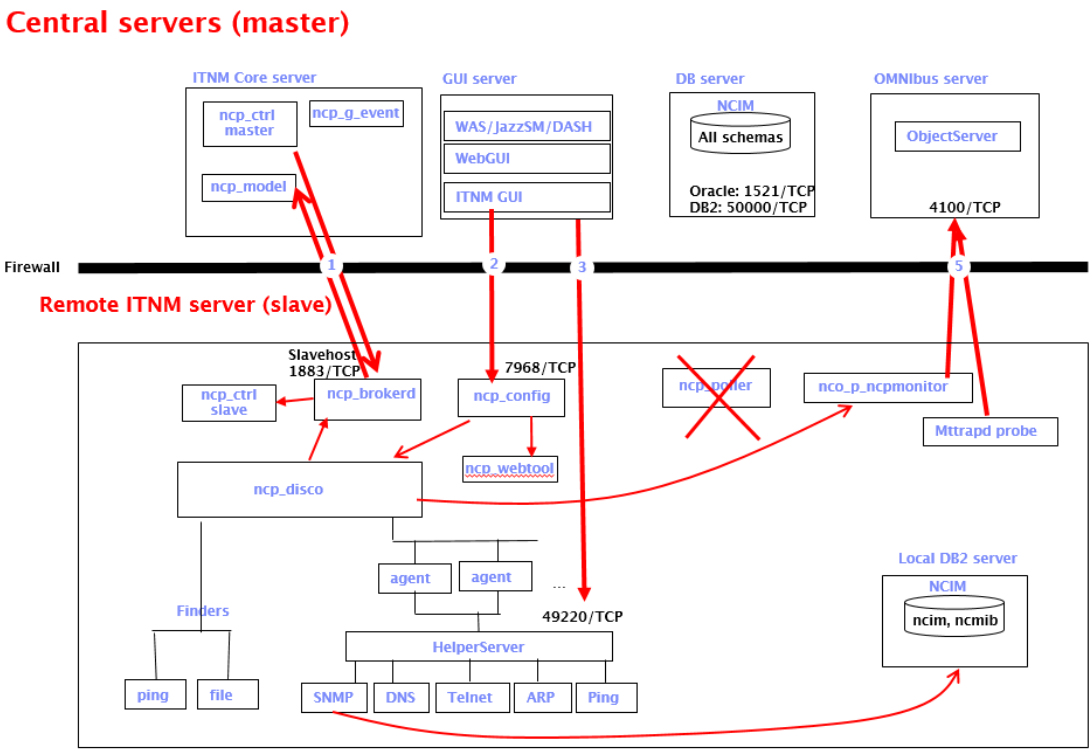
Restrict access to the central database

If you do not wish to open the database port from the remote server across the firewall to the central server, then you can use a local database on the remote server.

However, this introduces a restriction in function:

- The poller on the remote server cannot be used for polling. The poller must be able to read the tables in the Central database for the policy scopes and definitions.

The architecture diagram will look like this:



To modify the custom distributed architecture described above, the following changes must be made.

Changes on the central server

1. CtrlServices.domain.cfg: comment out the ncp_poller entries
2. Don't start the Storm processes on the Central server – it will have nothing to do.

Changes on the remote server

1. Before installing the ITNM core server, install a separate DB2 server on the remote server and create the database using the `create_db2_database.sh` script as usual.
2. During the ITNM core installation, connect to this local DB2 and allow the installer to create the schemas. This will set up the DbLogins and MibDbLogins files to use the local database.
3. If new SNMP MIBs are added to ITNM, they must be added to both the central and remote servers. Run `ncp_mib` separately on both the central and all remote servers to load the new MIBs into the `ncmib` tables on the respective databases.

Known Issues

We found two issues with the custom distributed setup using ITNM 4.2 FP1.

- APAR IV87956 - In a distributed environment Helper Server(ncp_d_helpserv) process core dumps during Phase#3 of a discovery cycle
- ncp_ctrl core dumps on slave server occasionally.
- ncp_ctrl on master fails to stop all processes successfully on the slave.

Notices

This information was developed for products and services offered in the U.S.A.
IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
1623-14, Shimotsuruma, Yamato-shi
Kanagawa 242-8502 Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS"
WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.
Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
958/NH04
IBM Centre, St Leonards
601 Pacific Hwy
St Leonards, NSW, 2069
Australia

IBM Corporation
896471/H128B
76 Upper Ground
London SE1 9PZ
United Kingdom

IBM Corporation
JBFA/SOM1
294 Route 100
Somers, NY, 10589-0100
United States of America

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compati-

bility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

All IBM prices shown are IBM's suggested retail prices, are current and are subject to change without notice. Dealer prices may vary.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs. Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

© (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. _enter the year or years_.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Trademarks

IBM, the IBM logo, ibm.com, Netcool, Netcool/OMNIbus,and Tivoli are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at www.ibm.com/legal/copytrade.shtml.

Adobe, Acrobat, and Portable Document Format (PDF) are trademarks or registered trademarks of Adobe Systems Incorporated in the United States, other countries, or both.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Microsoft and Windows are trademarks of Microsoft Corporation in the United States, other countries, or both.