IPv6 support in NorduGrid ARC middleware

Gábor Rőczei
roczei@niif.hu

NIIF / KnowARC / NorduGrid
The Hosting Environment Daemon (HED) is the container of all the functional components of the next generation of the Advanced Resource Connector (ARC) middleware on the server side. It is the central part in a new very lightweight incarnation of ARC that is aimed at - but not limited to - providing Web Service.

The whole design of the HED is built around the idea of flexibility and modularity. Inside HED the developer or deployer is supposed to use the minimum amount of components and external dependencies only. This is why the HED mostly consists of pluggable modules with some glue among them.
Main features of ARC1

- It uses web-service technology: HTML, SOAP, XML, WSDL
- Several platforms are supported: Linux, Mac OS X, Solaris, and Windows soon will be too
- The following languages can be used for service development: Python, C++, Java
- It consists of loadable modules
- Several security types are supported inside the container: X509Token, UsernameToken, SAMLToken
- IPv6 also supported
Complex configuration

IPv6 relevant part

Local policy

Policy Decision Point

Policy Decision Point

Policy

File service

A-REX service

TCP MCC

TLS MCC

HTTP MCC

HTTP MCC

Plexer

SOAP MCC

Counter

Logging

Info engine

Loader

Config

INTERNET

INTERNET

INTERNET

INTERNET

INTERNET
Server part (MCCTCP.cpp)

Uses getaddrinfo() for opening listening socket(s)

- hints(STREAM, TCP, PASSIVE)

getaddrinfo() - protocol independent name service function

sockaddr results

- socket() parameters
- bind() parameters
Client part (PayloadTCPsocket.cpp)

Uses getaddrinfo() for connecting server socket

```
hints(AF_UNSPEC, STREAM, TCP)
```

AF_UNSPEC means automatically selects the IPv6 or IPv4 address of the server

```
getaddrinfo() - protocol independent name service function
```

```sockaddr results
```

socket() parameters

connect() parameters
<Component name="tcp.service" id="tcp">
  <next id="tls"/>
  <tcp:Listen>
    <tcp:Port>60000</tcp:Port>
    <tcp:Version>4</tcp:Version>
  </tcp:Listen>
  <tcp:Listen>
    <tcp:Port>60001</tcp:Port>
    <tcp:Version>6</tcp:Version>
  </tcp:Listen>
</Component>
[roczei@knowarc4:~] $ps aux | grep arched
roczei 25751  0.0  4.1  20120  5400 pts/0  Sl+
   14:23   0:00 arched -f -c server.xml
roczei 25773  0.0  0.4   1968   628 pts/2  S+
   14:39   0:00 grep arched

[roczei@knowarc4:~] $

[roczei@knowarc4:~] $netstat -a | grep 6000
| tcp      | 0      | 0      | ::60000 | *:*     |
  |         | LISTEN |
| tcp6     | 0      | 0      | ::60001 | *:*     |
  |         | LISTEN |

[roczei@knowarc4:~] $
• ARC architecture is rational, because the network sub system role is well defined
• The original code had to be modified only in two files to support the IPv6 stack: MCCTCP.cpp, PayloadTCPSocket.cpp
• HED services have automatically IPv6 support, since they are users of the TCP MCC layer
• Using function getaddrinfo() has the following advantage: protocol independent, maybe later IPv7?
Live demo if possible
Questions?!