An Hierarchical Policy-Based Architecture for Integrated Management of Grids and Networks

Ricardo Neisse
Evandro Della Vechia Pereira
Lisandro Zambenedetti Granville
Maria Janilce Bosquirolí Almeida
Liane Margarida Rockenbach Tarouco

Federal University of Rio Grande do Sul
Outline

- Introduction
- Grids, networks and policies
- Hierarchical mapping architecture
- System prototype
- Conclusions and future work
Introduction

- Configuration of the underlying network to allow the grid operation
- Grid resources distributed along several network administrative domains: management problems?
- Network policies x Grid policies
- A system to generate network policies based on grid policies
Grids, networks and policies

Grids, networks and policies

Grid Management Infrastructure (Toolkit)

Grid node
(users and resources)

Administrative domain

Ricardo Neisse - Policy 2004
Slide 4 of 15
Grids, networks and policies

Grid administrator

Web Services (SOAP/HTTP)

Services

Resource (Cluster)

Local network administrator

Network infrastructure
Hierarchical policies and grids

- Grid management policies
  - Mapping defined by the network administrator
- Network management policies
  - Policy deployment via PDPs
- Configuration actions into devices
if (user == "neisse" and
    startTime >= "11/25/2003 00:00:00" and
    endTime <= "11/25/2003 23:59:59")
{
    if (resource == "LabTec Cluster") {
        allowAccess = true;
        login = griduser;
        maxProcessing = 50%;
        networkQoS = remoteProccessControl;
    }
    if (proxy == "LabTec Cluster" and
        resource == "UFRGS Data Server")
    {
        allowAccess = true;
        maxAllowedStorage = 40GB;
        networkQoS = highThroughputDataIntensive;
    }
}
if (srcResource.address/24 == 143.54.47.0/24 and
dstResource.address/24 != 143.54.47.0/24 and
dstResource.port == 80 and
dstResource.protocol == TCP)
{
    p1 = new NetworkPolicy();
    p1.addCondition(startTime,">=", schedule.startTime);
    p1.addCondition(endTime, "<=" , schedule.endTime);
    p1.addCondition(srcAddress, "==", srcResource.address);
    p1.addCondition(dstAddress, "==", dstResource.address);
    p1.addCondition(dstPort, "==", dstResource.port);
    p1.addCondition(dstProtocol, "==", "tcp");
    p1.addAction(DSCP, 2);

    p2 = new NetworkPolicy();
    p2.addCondition(startTime,">=", schedule.startTime);
    p2.addCondition(endTime, "<=" , schedule.endTime);
    p2.addCondition(DSCP, 2);
    p2.addAction(bandwidth, requiredQoS.requiredBandwidth);
}
Mapping architecture

```java
if (srcResource.address/24 == 143.54.47.0/24 and
    dstResource.address/24 != 143.54.47.0/24 and
    dstResource.port == 80 and
    dstResource.protocol == TCP)
{
    p1 = new NetworkPolicy();
    ...
    inPEPs = select pep
        .within[srcResource.address, 143.54.47.1]
        .direction["in"]
        from device.type["DiffServDevice"];
    inPEPs[0].deployPolicy(p1);

    p2 = new NetworkPolicy();
    ...
    outPEPs = select pep
        .within[srcResource.address, 143.54.47.1]
        .direction["out"]
        from device.type["DiffServDevice"];
    outPEPs.deployPolicy(p2);
}
```
System prototype

Grid Policies

Policy List

// UFRGS VO Policies
if (user == 'neisse' AND startTime >= '25/11/2003 00:00:00' AND endTime <= '25/11/2003 23:59:59') {
    Insert(Rule/Action) ::
    if (resource == 'Cluster LABTEC') {
        Insert(Rule/Action) ::
        AllowAccess = true;
        Login = gridUser;
        NetworkQoS = RemoteProcessControl;
        MaxProcessing = 50%;
    }
    if (resource == 'Data Server' AND proxy == 'Cluster LABTEC') {
        Insert(Rule/Action) ::
        AllowAccess = true;
        NetworkQoS = HighThroughputDataIntensive;
        MaxAllowedStorage = 40Gb;
    }
}

::: New Policy :::
System prototype

Mapping Rules

```java
// Mapping Rule test
if (srcResource.address == "143.54.47.0/24" AND
    dstResource.address != "143.54.47.0/24" AND
    dstResource.port == "80" AND
    dstResource.protocol == "tcp")
{
    ... Insert Condition ...;
    p1 = new NetworkPolicy();
    p1.addCondition(startTime, ">=", schedule.startTime);
    p1.addCondition(endTime, ">=", schedule.endTime);
    p1.addCondition(srcAddress, ">=", srcResource.address);
    p1.addCondition(dstAddress, ">=", dstResource.address);
    p1.addCondition(dstPort, ">=", dstResource.port);
    p1.addCondition(dstProtocol, ">=", "tcp");
    p1.addAction(DSCP, 2);
    inPEPS =
    select pep.within[srcResource,"143.54.47.0/24", direction,"in"]
    from devices.type["DiffServRouter"];
    inPEPS[0].deployPolicy(p1);
    p2 = new NetworkPolicy();
    p2.addCondition(startTime, ">=", schedule.startTime);
    p2.addCondition(endTime, ">=", schedule.endTime);
    p2.addCondition(DSCP, 2);
    p2.addAction(bandwidth, requiredQoS, requiredBandwidth);
    outPEPS =
    select pep.within[srcResource,dstResource].direction["out"]
    from devices.type["DiffServRouter"];
    outPEPS.deployPolicy(p2);
}
... New Mapping Rule ...
```
Conclusions

- Grid policies: they are needed, but with network policies integration
- Mapping rules are not easy to define, requires:
  - Preview agreement between grid and network administrator
  - Good knowledge of the network and grid infrastructure
- Future work
  - How to make the definition of mapping rules easier?
  - Bandwidth and performance evaluation
  - Policy conflicts
Questions?

- **Contact information:**
  - Ricardo Neisse
  - Federal University of Rio Grande do Sul
  - E-mail: neisse@inf.ufrgs.br
  - http://gerencia.inf.ufrgs.br

- **Thanks for your attention!**