

Grid Middleware: L'interazione con IPv6

Valentino R. Carcione
valentino.carcione@garr.it
GARR

[GARR WS7-Roma-16-11-2006]

Grid e IPv6, quali vantaggi?

- ▶ **IPv6** offre uno spazio di indirizzamento molto ampio che può consentire connettività end-to-end a tutti i nodi della griglia senza utilizzare NAT.
- ▶ I meccanismi di autoconfigurazioni presenti nel protocollo rendono il setup dei nodi più semplice specialmente nel caso in cui sia necessario rinumerare una grossa server farm.

Grid e IPv6, quali vantaggi?

- ▶ Le caratteristiche intrinseche del protocollo IPv6 rendono più efficiente l'uso della rete. Alcune caratteristiche che per IPv4 sono un "add-on" sono native in IPv6, per esempio:
 - mobilità
 - Security e crittografia
 - multicast vs broadcast
 - Header del pacchetto estendibile
- ▶ Meccanismi di autenticazione integrata possono essere utilizzati per migliorare, dal punto di vista della sicurezza, le performance della griglia.

Perchè IPv6?

- ▶ IPv6 è attualmente utilizzato parallelamente ad IPv4, anche se una data di transizione non è prevista si suppone una probabile migrazione graduale ad IPv6
- ▶ L'affermarsi di nuove comunità di utenti renderà sempre più necessario l'utilizzo di un nuovo spazio di indirizzamento
- ▶ Diversi progetti relativi alle griglie in Europa e specialmente in Asia hanno già previsto di utilizzare il protocollo IPv6.
- ▶ Il middleware di **EGEE**, che è attualmente lo standard Europeo "de facto", non supporta questo protocollo.

L'interazione con IPv6

- ▶ Per poter utilizzare IPv6 il middleware deve:
 - utilizzare API apposite in grado di indirizzare l'address-family IPv6 per tutte le funzioni che accedono alla rete
 - Utilizzare strutture dati idonee a contenere le informazioni relative al protocollo IPv6 (l'indirizzo IPv6 è grande 128bit contro i 32bit di IPv4)
 - Utilizzare librerie di sistema compatibili con IPv6
 - Essere scritto secondo uno stile di programmazione che garantisca l'indipendenza dal protocollo utilizzato. Vanno evitati indirizzi e nomi host hard-coded nel codice e/o l'utilizzo esplicito di token dipendenti dalla piattaforma e dal protocollo

Un esempio

```
if ( (sock=socket(AF_INET, SOCK_DGRAM, 0)) < 0) {  
    perror("Socket creation error");  
    return (-1);  
}
```

- ▶ Il codice riportato è un esempio di programmazione dipendente dal protocollo

Problematiche classiche

- ▶ Riassumendo ecco una lista delle tipiche problematiche che sono di ostacolo alla portabilità del codice:
 - Indirizzi IPv4 scritti direttamente nel codice (p.es. 127.0.0.1 invece di *localhost*)
 - Funzioni e chiamate di sistema dipendenti dal protocollo
 - Parametri o costanti dipendenti dal protocollo
 - Librerie incompatibili con il protocollo IPv6
 - Utilizzo di indirizzi numerici invece che di nomi-host
 - Strutture dati troppo piccole per contenere le informazioni IPv6

RFC-3493

- ▶ L'interfaccia generica per la programmazione di rete multiprotocollo è descritta nella *RFC-3493 "Basic socket interface extension for IPv6"*
- ▶ La RFC definisce nuove:
 - Funzioni per l'accesso ai socket
 - Strutture dati per l'indirizzamento
 - Funzioni per la traduzione di nomi host a indirizzi
 - Funzioni per la conversione di indirizzi di rete
- ▶ Le interfacce di programmazione descritte nella *RFC-3493* forniscono un set di istruzioni che consente una facile conversione del codice esistente in IPv4 ad un ambiente multiprotocollo
- ▶ Il livello di astrazione fornito dalle API rende possibile l'indipendenza del codice dalla tipologia di rete utilizzata

Analisi di compatibilità del middleware

- ▶ Da un primo controllo sommario del codice di **gLite** (il middleware sviluppato da EGEE) si evidenzia l'incompatibilità del codice rispetto alla *RFC-3493*

```
[~/org.glite.wms-utils.tls] $ grep F_INET ./src/socket++/Socket*.cpp
SocketClient.cpp: agent -> peeraddr_in.sin_family = AF_INET;
SocketClient.cpp: if( (agent -> sck = socket(AAF_INET, SOCK_STREAM, 0))
    == -1 ) {
SocketServer.cpp: myaddr_in.sin_family = AF_INET;
SocketServer.cpp: sck = socket (AF_INET, SOCK_STREAM, 0);
```

- ▶ Purtroppo questo tipo di dipendenze può essere risolto solo dagli sviluppatori.

Analisi di compatibilità del middleware

- ▶ Una analisi più dettagliata è stata avviata su uno *Storage Element*
- ▶ Nell'esempio si vede come nonostante le librerie di sistema risolvano correttamente gli indirizzi IPv6, il software non è in grado di interpretare correttamente le informazioni ricevute

```
[ipv6@roma02 ipv6]$ dpns-ls /
send2nsd: NS009 - fatal configuration error: Host unknown:
  roma02.ipv6.euchinagrid.eu
[ipv6@roma02 ipv6]$ strace dpns-ls /
...
recvfrom(3, "\263\354\201\203\0\1\0\0\0\1\0\0\6roma02\4ipv6\veuchin"... ,1024,
0, {sa_family=AF_INET, sin_port=htons(53),
  sin_addr=inet_addr("193.206.158.1")}, [16]) = 123
close(3)
```

Verso un middleware IPv6?

- ▶ All'interno del progetto Europeo **EUChinaGrid** sono state avviate alcune attività volte a promuovere lo sviluppo di un middleware di griglia compatibile con IPv6
- ▶ Queste attività sono state avviate anche con il supporto di un apposito testbed IPv6 realizzato presso GARR

Il progetto EuChinaGRID

- ▶ EuChinaGrid “Interconnection and Interoperability of Grids between Europe and China”, è una iniziativa per estendere l’infrastruttura di griglia Europea per l’e-science alla Cina
- ▶ L’obiettivo principale del progetto è quello di facilitare lo scambio e l’elaborazione dei dati scientifici, favorendo l’interoperabilità delle più importanti strutture di Griglia Europee (sviluppate all'interno del Progetto Europeo EGEE) e Cinesi (gestite dal Progetto CNGrid)

EuChinaGrid WP2

- ▶ Lo studio e l'analisi delle problematiche di rete è svolto all'interno del WP2

- ▶ I principali obiettivi del WP2 sono:
 - Studio della connettività presente e futura e promozione di nuovi servizi di connettività a banda larga tra Europa e Cina

 - Studio della disponibilità di middleware di griglia con supporto IPv6 e dell'interazione tra i servizi di griglia e la rete IPv6 per fornire indicazioni e proporre modifiche agli sviluppatori.

Attività su IPv6

- ▶ All'interno dell'activity 2.2 vengono affrontate le problematiche relative al protocollo IPv6

- ▶ Gli obiettivi dell'activity sono:
 - Promuovere l'integrazione delle infrastrutture di griglia tra Europa e Cina in un ambiente multiprotocollo

 - Promuovere l'utilizzo del protocollo IPv6 nell'infrastruttura di griglia Europea in collaborazione con altri progetti europei (EGEE)

 - Collaborare alla realizzazione di una versione IPv6-compatibile del middleware di griglia.

Attività in atto

- ▶ Conferenza EGEE06: attivato un punto di contatto tra il progetto EUChinaGrid ed **EGEE-JRA1** per promuovere lo sviluppo di una versione IPv6 enabled della griglia.

- ▶ Testbed IPv6 Grid

IPv6 Programming guidelines

- ▶ Una delle richieste degli sviluppatori del middleware è stata quella di avere un documento che descriva le linee guida per una programmazione di rete compatibile IPv6
- ▶ Il documento descrive i principi di base della programmazione di rete in IPv6 nei quattro linguaggi di programmazione utilizzati dagli sviluppatori di gLite:
 - C
 - Perl
 - Python
 - Java
- ▶ Lo scopo del documento è di essere una guida iniziale che fornisca le basi per la programmazione di rete multiprotocollo.
- ▶ In fase di pubblicazione

Third-party middleware compatibility

- ▶ Una ulteriore analisi di compatibilità è stata avviata sulle componenti sviluppate da terze parti (GLOBUS, CONDOR, etc.)
- ▶ è stata analizzata la documentazione relativa all'uso e all'implementazione degli applicativi di terze parti per verificarne il funzionamento multiprotocollo ed individuare eventuali workaround che ne consentano l'utilizzo su una rete IPv6
- ▶ Ove possibile si è cercato di verificare la compatibilità IPv6 degli applicativi direttamente con gli sviluppatori
- ▶ è in fase di pubblicazione un report che descrive i risultati di questa analisi

IPv6 Code checker

- ▶ Per consentire agli sviluppatori di individuare più facilmente problemi di dipendenze relativi all'utilizzo di codice non portabile è stato sviluppato il tool *codechecker*
- ▶ Si tratta di uno script ottimizzato per analizzare l'albero dei codici sorgenti di gLite alla ricerca di token IPv4 specifici
- ▶ Lo script può analizzare codici scritti in:
 - C/C++
 - Java
 - Python
 - Perl

IPv6 Code checker

▶ Esempio di output

```
org.glite.security.voms
INADDR_ [ FAILED ]
addr_in [ FAILED ]
F_INET$ [ PASSED ]
gethostbyname [ FAILED ]
inet_addr [ PASSED ]
inet_ntoa [ PASSED ]
Inet4Address [ PASSED ]
inet_aton [ PASSED ]
gethostbyname_ex [ PASSED ]
INADDR_BROADCAST [ PASSED ]
0.0.0.0 [ FAILED ]
127.0.0.1 [ PASSED ]
255.255.255.255 [ PASSED ]
```

IPv6 testbed

- ▶ E' in fase di attivazione un testbed IPv6 nativo all'interno del progetto EUChinaGrid
- ▶ Hanno manifestato interesse al testbed anche partner europei non direttamente coinvolti nel progetto
- ▶ Un apposito spazio di indirizzamento è stato allocato da GARR per gli utenti italiani che partecipano al testbed

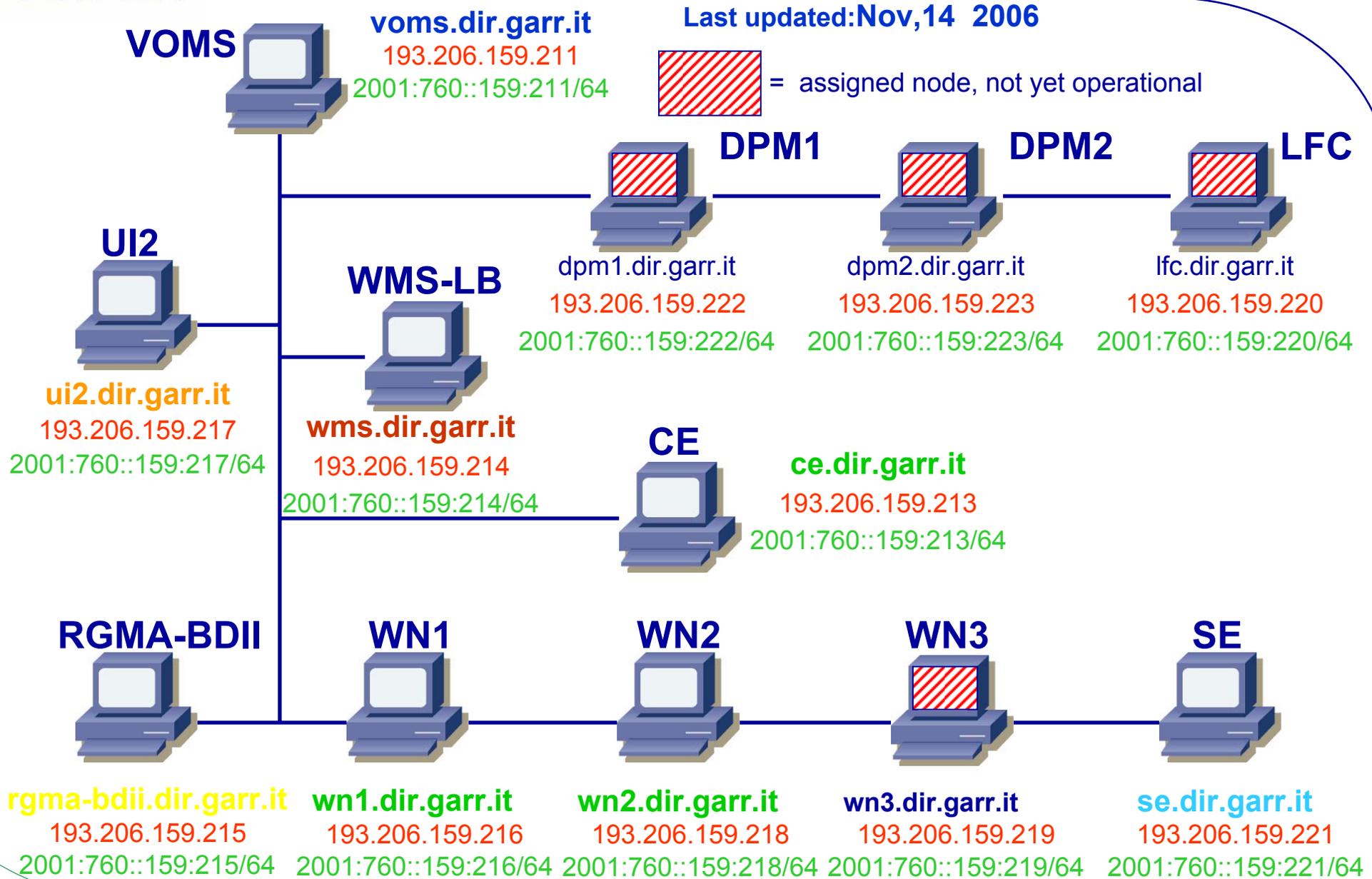
IPv6 testbed @GARR

- ▶ Attivato un sito di test per il middleware grid completo (o quasi)
- ▶ è composto da:
 - UI User Interface
 - WMS-LB Workload Managent and Logging and Bookkeepeing
 - VOMS Virtual Organization Management Server
 - RGMA-BDII GRID Information Systems
 - CE Computing Element (Torque)
 - WN Worker Nodes (3 nodi)
 - SE Classic Storage Elements
 - LFC LCG File Catalog
 - DPM Disk Pool Manager (2 nodi)

Last updated: Nov, 14 2006



= assigned node, not yet operational



IPv6 testbed @GARR

- ▶ Hardware: Macchine virtuali su VMWARE ESX server
- ▶ Software:
 - Sistema Operativo Scientific Linux CERN 3.0.8
 - gLite 3.0.6
- ▶ In fase di completamento

Riferimenti

- ▶ The EUChinaGRID initiative
<http://www.euchinagrid.eu/>
- ▶ EuChinaGrid IPv6 web site
<http://www.euchinagrid.org/IPv6/>
- ▶ The EGEE Project Enabling Grids for E-scienceE <http://public.eu-egee.org/>

GRAZIE!