

Verso la standardizzazione delle informazioni relative al middleware applicativo di supporto alle applicazioni di calcolo scientifico

Descrizione del problema

Le applicazioni di calcolo scientifico in grado di affrontare i cosiddetti problemi-sfida sono spesso, se non sempre, basate su un insieme anche cospicuo di strumenti software, che d'ora in poi chiameremo middleware applicativo (librerie, applicativi, problem solving environment) che forniscono il necessario ambiente per la loro esecuzione efficace ed efficiente sulle risorse di calcolo. Le comunità relative a tali applicazioni hanno la possibilità o di utilizzare risorse locali o remote, inserite in contesti di calcolo distribuito. Se da un lato l'utilizzo di risorse locali garantisce la possibilità di un controllo più diretto delle modalità di configurazione del software (grazie ad un maggior livello di interazione con i gestori delle risorse), dall'altro si può manifestare il problema della limitatezza delle risorse hardware a disposizione della comunità che può necessitare di maggiori risorse per un più rapido avanzamento della conoscenza nel proprio settore di competenza.

L'utilizzo di ambienti in cui le risorse di calcolo sono distribuite fornisce invece una buona risposta al problema sopra descritto. Tuttavia non può essere trascurato che, in tali contesti, nascono problematiche relative a:

- come verificare la presenza dei software di interesse sulle risorse di calcolo dell'infrastruttura distribuita
- come avere informazioni sul software (versione installata, compilatori utilizzati per l'installazione, librerie richiamate dal software e relative versioni, se è MPI-based quale distribuzione MPI utilizza, ...)
- come reperire informazioni sulle modalità di fruizione (documentazione ed esempi d'uso).

Di seguito viene descritto lo stato dell'arte sui metodi e sugli strumenti oggi disponibili, in IGI/EGI, per la corretta gestione del middleware applicativo. Infine viene riportata una proposta per la creazione di un Virtual Team EGI che abbia l'obiettivo di definire lo standard di gestione del middleware applicativo (relative alle procedure per la sua installazione, documentazione e disseminazione).

Stato dell'arte

Attualmente non sono disponibili procedure e strumenti per l'installazione del middleware applicativo che garantiscano un certo QoS: in particolare non esistono procedure automatiche che consentano l'installazione di librerie ottimizzate (mediante ad es. compilazione con flavour particolari, con compilatori diversi da quello di default, ecc.). L'installazione ottimizzata di librerie e software va fatta manualmente ad opera degli amministratori di sito.

Inoltre tra gli strumenti per la pubblicazione delle informazioni sul middleware applicativo possiamo annoverare solo:

- lo strumento lcg-tags e
- l'Application Database

Il secondo non è facilmente integrabile con le informazioni cui sono interessate le comunità di utenti. Il primo consente di pubblicare informazioni sulle librerie software disponibili su un determinato sito ed accessibili mediante il CE.

Per gli amministratori della VO che si autenticano cioè con il ruolo lcgadmin ("voms-proxy-init --voms gridit:/gridit/Role=lcgadmin") sono disponibili i comandi:

Verso la standardizzazione delle informazioni relative al middleware applicativo di supporto alle applicazioni

```
> lcg-tags --ce <ce_hostname> --vo <VOname> --add --tags VO-VOname-TAGname  
> lcg-tags --ce <ce_hostname> --vo <VOname> --remove --tags VO-VOname-TAGname
```

gli utenti della VO che si autenticano con il comando "voms-proxy-init --voms gridit", possono ottenere informazioni sulla dotazione software di un certo cluster mediante il comando:

```
> lcg-tags --ce <ce_hostname> --vo <VOname> --list
```

Proposta

L'obiettivo ultimo da raggiungere è la definizione:

1. di standard, a livello del sistema informativo, per la corretta descrizione dello strato software applicativo
2. CVMfs
3. di un template di documentazione del software che semplifichi l'interazione dell'utente con le risorse di calcolo che hanno una certa dotazione di software applicativo

La realizzazione del punto 1 rappresenterebbe l'optimum per tutte le comunità di e-Science (HPC e non) ma è un obiettivo realizzabile solo attraverso uno sforzo congiunto e coordinato di più gruppi di lavoro: da quello che lavora al Middleware, al FUS assieme al gruppo MPI-Multicore, al gruppo che si occupa in EGI del Glue schema, eccetera.

Tuttavia un primo approccio per soddisfare almeno in parte le esigenze delle comunità di utenti potrebbe essere il seguente:

- richiedere al gruppo middleware di inserire, tra le dipendenze dei metapacchetti relativi ai Worker Nodes, anche un set minimale di librerie scientifiche consolidate di base
- utilizzare la variabile yaim CE-RUNTIMENV per pubblicare informazioni relative all'interconnessione, alle versioni di MPI disponibili specificando laddove significativo anche la versione del compilatore utilizzato dalla distribuzione MPI ed eventualmente il flavour OPENMP
- utilizzare gli lcg-tags in maniera estesa e significativa per segnalare in maniera esaustiva (il più possibile) la presenza di una certa libreria.
- infine in merito alla documentazione, sempre mediante la variabile CE-RUNTIMENV si potrebbe pubblicare una variabile SOFTWARE-DOC-URL che punta ad un URL dove si possa scaricare il manuale d'uso del software installato sul sito.

Esempi Operativi

Librerie di Base:

L'elenco dei pacchetti che dovrebbero essere installati sui WNs include:

```
lapack-devel, blas-devel gsl-devel, fftw3-devel, gmp-devel, cfitsio-devel, netcdf-devel, hdf5-devel
```

Per dimostrare questa funzionalità e' stato creato il meta-pacchetto igi-wn_env che installa (come dipendenze) i pacchetti di cui sopra:

```
wget http://www.fis.unipr.it/grid/rpm/igi-wn_env-20120707-s15.noarch.rpm  
yum install igi-wn_env-20120707-s15.noarch.rpm
```

-- VaniaBoccia - 2012-06-25

This topic: MPI > AmbienteSoftwareApplicazioni
Topic revision: r9 - 2012-10-15 - RobertoAlfieri



Copyright © 2008-2020 by the contributing authors. All material on this collaboration platform is the property of the contributing authors.

Ideas, requests, problems regarding TWiki? Send feedback