NAME

gcf, GCF – Grid Configuration File format

DESCRIPTION

A file in the GCF (Grid Configuration File) format has a **.gcf** extension and specifies a list of "instances" (i.e., distributed processes) to be launched by the CADP tools providing distributed verification, such as **distributor**(LOCAL), **bcg_merge**(LOCAL), and the PBG tools as well: **pbg_cp**(LOCAL), **pbg_info**(LOCAL), etc. Such a tool will be called the "parent program" in the sequel. The GCF format also specifies the various variables used for launching, connecting and parameterizing instances on remote machines.

Each instance corresponds to a pair (*machine*, *directory*). For each instance, a distributed process executing on *machine* will be launched by the parent program. This distributed process will store its files in *directory* located on some filesystem of *machine*. *directory* is called the "working directory" of the instance.

Instances may or may not be launched on the local machine, depending on the constraints on grid usage. For instance, clusters often distinguish between one frontal node (i.e., the local machine) used to submit jobs to the cluster, and many computing nodes (i.e., the remote machines) that perform distributed computations.

Each instance is given a unique number greater or equal to 1. Numbers are assigned in the same order as instances appear in the grid configuration file. Number 0 is reserved for the process corresponding to the execution of the parent program on the local machine.

If the grid configuration file contains duplicated instances (i.e., same machine and same directory), only the last one is considered, the previous ones being entirely discarded.

There should be at least two different instances. Several instances may execute on the same machine, provided that their working directories are different. Also, a working directory may be either local to its machine or shared between several machines (using NFS, Samba, etc.).

The grid configuration file also specifies the value of (predefined) variables used to control the way instances are launched and communicate with each other.

SYNTAX OF THE GCF FORMAT

The grid configuration file has the following syntax:

```
<gcf>         ::= <global_opt> <instance_opt>*
<global_opt>  ::= <directive>*
<instance_opt> ::= <machine> <machine>* <directive>*
<machine>     ::= <address-or-name> "\n"
<directive>   ::= <variable> "=" <value> "\n"
<variable>    ::= "buffer_size" | "cadp" | "connect_timeout"
                | "directory" | "files" | "hash" | "memory"
                | "port" | "rcp" | "rsh" | "time" | "user"
<value>       ::= <character-string>
```

where:

- **<global_opt>** is a (possibly empty) list of directives applicable to every instance.

- **<instance_opt>** defines one or several instances together with a (possibly empty) list of directives applicable to these instances exclusively.

- **<machine>** is the name of a (local or remote) machine, either given as a numerical IP address (e.g., "138.96.146.2") or as an Internet machine name (e.g., "vasy.inria.fr"), followed by a newline character.

- **<directive>** is an assignment of a **<value>** to a **<variable>**, followed by a newline character.

Any line starting with the "#" character is considered as a comment and ignored. Spaces and tabulations can be inserted before, between, or after terminal symbols.

GCF VARIABLES

Variables can be modified by directives as follows. First, some variables have a default value, which will be used unless overriden by some directive. A **<directive>** occurring in the **<global_opt>** list assigns its **<variable>** for all machines. A directive occurring in a **<instance_opt>** list assigns its **<variable>** only for the machines mentioned in this **<instance_opt>**, possibly overriding the value specified for this **<variable>** in the **<global_opt>** list.

The meaning of the different variables is the following:

buffer_size:

This variable specifies the size (in bytes) of the communication buffers used between instances (either for incoming data or outgoing data). This variable can only be set in the **<global_opt>** list, meaning that all communication buffers must have the same size. Maximal value is 2,147,483,648. Default value is 65536.

cadp: This variable specifies the pathname of the directory in which the CADP toolbox is installed. This pathname must be an absolute one, i.e., it must start with a "/" character. It may contain spaces provided that the entire pathname is enclosed between double quotes.

The pathname may be the same for all machines (e.g., if the CADP toolbox is installed on a common filesystem shared between all machines) or may be different on each machine. In any case, the CADP toolbox should be properly installed on the local and remote machines. Default value is given by the **$CADP** environment variable.

connect_timeout:

This variable specifies the timeout (in seconds) allowed for establishing a connection to a remote machine. This variable can only be set in the **<global_opt>** list, meaning that each connection must succeed within the specified timeout. Default value is 20 seconds.

directory:

This variable specifies the pathname of the working directory in which an instance will place all the files needed for its execution or created during its execution (e.g., a copy of the binary code of the parent program, a copy of the GCF file, a generated BCG file fragment, log files, etc.). If the specified directory does not exist, it will be created.

This pathname can be either an absolute one (starting with a "/" character) or a relative one, in which case it is interpreted with respect to the current directory on the local machine and to the user's home directory on remote machines. It may contain spaces provided that the entire pathname is enclosed between double quotes. It should not contain the tilde character "~". Default value is the user's home directory (which is not recommended unless when using a cluster of machines).

files: This variable specifies a set of files to be copied to the remote machines (in the working directory) before launching the different instances. Setting this variable provides to instances executing on remote machines additional input files needed for their execution. This allows to take into account "hidden" dependences, as these input files may or may not be mentioned on the command line used to invoke the parent program.

For instance, when running the parent program **distributor**(LOCAL) using **bcg_open**(LOCAL), the *spec*.**bcg** file must be present on the remote machines. Similarly, when running **distributor**(LOCAL) using **exp.open**(LOCAL), all the BCG files referenced in *spec*.**exp** must also be present on the remote machines.

This variable may specify either a single filename or a set of filenames separated by spaces. Filenames may contain the wildcard characters ("*", "?", "[", "]", ...) according to the Bourne shell conventions. Default value is empty.

hash: This variable specifies the hash function to be used for partitioning states between machines (see [GMS01] for details). Its value is an integer *n* between 0 and 7 denoting the corresponding function **CAESAR_STATE_***n*__HASH()** of the **caesar_hash**(LOCAL) library of OPEN/CAESAR. This variable can only be set in the **<global_opt>** list, meaning that all instances use the same hash function. Default value is 3.

memory:
This variable has currently no effect. It is reserved for future use.

port: This variable specifies the TCP port to be used for communicating with the instance. Usually, it is not necessary to set this variable explicitly unless to comply with firewall restrictions. A TCP port is an integer value in the range 1...65,535. Using a TCP port lower than 1024 is not recommended and may require special privileges (such as being the super-user on Unix systems). By default, or in case of failure when opening the specified port, the port value is chosen automatically.

rcp: This variable specifies the command used to copy files from the local machine to the working directories on the remote machines. This command must be present in one of the directories given by the **$PATH** environment variable. Its command line syntax should be the same as for the **rcp**(1) command, i.e.,

rcp *local_files username@hostname*:*directory*

Default value is **rcp**. Other possible values are **scp**, **kcp**, or any similar variant of **rcp**(1). Remote machines should be configured in such a way that remote copy can be done automatically without asking for a password or a passphrase; this is usually done by setting an appropriate ".rhosts" or ".ssh/authorized_keys" file and/or launching an SSH agent before running the parent program.

There is also a special value **nfs**, which indicates that all instances have the same working directory on a filesystem shared between all machines (e.g., using NFS), so that no remote copy is

necessary. Setting variable **rcp** to the value **nfs** can only be done in the **<global_opt>** list.

rsh: This variable specifies the command used to lauch instances on remote machines from the local machine. This command must be present in one of the directories given by the **$PATH** environment variable. Its command line syntax should be the same as for the **rsh**(1) command, i.e.,
rsh -l *username hostname* "*shell_command ...*"

Default value is **rsh**. Other possible values are **ssh**, **krsh**, or any similar variant of **rsh**(1). Remote machines should be configured in such a way that remote authentication can be done automatically without asking for a password or a passphrase; this is usually done by setting an appropriate ".rhosts" or ".ssh/authorized_keys" file and/or launching an SSH agent before running the parent program.

It is possible to supply options to the command assigned to the *rsh* variable (such as **rsh=ssh -6**) but certain options (e.g., **ssh -v**) program.

time: This variable, if set, contains a Unix command name that will be inserted before the shell commands used to launch instances on remote machines. Possible values of this variable are "/usr/bin/time", "memtime", "valgrind", etc. It is therefore possible to monitor the time, memory, etc. used by remote instances on these machines. Default value is the empty string.

Alternatively, one can obtain the same effect by setting the (undocumented) **$CADP_TIME** environment variable; this is equivalent to setting in the ".gcf" file the "time" variable for all remote instances.

user: This variable specifies the user login name (or account name) to be used for authentication when connecting to the remote machine(s). Default value is the name of the user running the parent program on the local machine.

EXAMPLES OF GCF FILES

In its simplest form, a grid configuration file may simply list the remote machines to be used:

```
machine1
machine2
machine3
machine4
```

It is also possible to write a grid configuration file with several instances on the same machine:

```
machine1
  directory=/tmp/instance1
machine1
  directory=/tmp/instance2
machine1
  directory=/tmp/instance3
```

This is a more complex example involving all the features described above:

```
buffer_size = 32768
cadp = /usr/local/cadp
connect_timeout = 10
```

```
directory = /home/vasy/distributor
files = graph-*.bcg
hash = 4
port = 8016
rcp = scp
rsh = ssh
user = inria
machine1.domain.org
machine2.domain.org
  user = vasy
machine3.domain.org
  directory = /users/inria/distributor
```

Notice that the directive "**`user = vasy`**" applies to both **`machine1`** and **`machine2`**.

BIBLIOGRAPHY

[GMS01] Hubert Garavel, Radu Mateescu, and Irina Smarandache. Parallel State Space Construction for Model-Checking. In Matthew B. Dwyer, ed, Proceedings of the 8th International SPIN Workshop on Model Checking of Software (SPIN'01), Toronto, Canada, LNCS 2057, pp. 217-234, May 2001. Revised version available as INRIA Research Report RR-4341, December 2001. Available from http://cadp.inria.fr/publications/Garavel-Mateescu-Smarandache-01.html

[GMB+06] Hubert Garavel, Radu Mateescu, Damien Bergamini, Adrian Curic, Nicolas Descoubes, Christophe Joubert, Irina Smarandache-Sturm, and Gilles Stragier. DISTRIBUTOR and BCG_MERGE: Tools for Distributed Explicit State Space Generation. In Holger Hermanns and Jens Palberg, eds., Proceedings of the 12th International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS'06), Vienna, Austria, LNCS 3920, pp. 445-449, March-April 2006. Available from http://cadp.inria.fr/publications/Garavel-Mateescu-Bergamini-et-al-06.html

[GMS12] Hubert Garavel, Radu Mateescu, and Wendelin Serwe. Large-scale Distributed Verification using CADP: Beyond Clusters to Grids. Electronic Notes in Theoretical Computer Science, vol. 296, pp. 145-161, August 2012. Available from http://cadp.inria.fr/publications/Garavel-Mateescu-Serwe-12.html

SEE ALSO

bcg(LOCAL), **pbg**(LOCAL), **bcg_merge**(LOCAL) **distributor**(LOCAL), **pbg_cp**(LOCAL), **pbg_info**(LOCAL), **pbg_mv**(LOCAL), **pbg_rm**(LOCAL)

Additional information is available from the CADP Web page located at http://cadp.inria.fr

Directives for installation are given in files **$CADP/INSTALLATION_***.

Recent changes and improvements to this software are reported and commented in file **$CADP/HISTORY**.

BUGS

Please report bugs to cadp@inria.fr