The Grid Workload Format

1. Introduction

This represents an extension to the Standard Workload Format (swf, [1]) used for the traces stored in the Parallel Workloads Archive (PWA,[2]).

2. The Grid Workload Format

Goals:

- Provide a unitary format for Grid workloads;
- Make the format interoperable between text and relational databases.

Color codes

Identification fields	
Time- and status-related	
Resource consumption	
Job structure	
Job request information	
Others	

ID	SWF	GWF	Info
1	Job Number	JobID	a counter field, starting from 1.
2	Submit Time	SubmitTime	in seconds. The earliest time the log refers to is zero, and is the submittal time the of the first job. The lines in the log are sorted by ascending submittal times. In makes sense for jobs to also be numbered in this order.
3	Wait Time	WaitTime	in seconds. The difference between the job's submit time and the time at which it actually began to run. Naturally, this is only relevant to real logs, not to models.
4	Run Time	RunTime	in seconds. The wall clock time the job was running (end time minus start time). We decided to use ``wait time" and ``run time" instead of the equivalent ``start time" and ``end time" because they are directly attributable to the scheduler and application, and are more suitable for models where only the run time is relevant. Note that when values are rounded to an integral number of seconds (as often happens in logs) a run time of 0 is possible and means the job ran for less than 0.5 seconds. On the other hand it is permissable to use floating point values for time fields.
5	Number of Allocated Processors	NProc	an integer. In most cases this is also the number of processors the job uses; if the job does not use all of them, we typically don't know about it.

ID	SWF	GWF	Info
6	Average CPU Time Used		both user and system, in seconds. This is the average over all processors of the CPU time used, and may therefore be smaller than the wall clock runtime. If a log contains the total CPU time used by all the processors, it is divided by the number of allocated processors to derive the average.
7	Used Memory	Used Memory	in kilobytes. This is again the average per processor.
8	Requested Number of Processors	ReqNProcs	
9	Requested Time	ReqTime	This can be either runtime (measured in wallclock seconds), or average CPU time per processor (also in seconds) the exact meaning is determined by a header comment. In many logs this field is used for the user runtime estimate (or upper bound) used in backfilling. If a log contains a request for total CPU time, it is divided by the number of requested processors.
10	Requested Memory	ReqMemory	(again kilobytes per processor).
11	Status	???	SWF: 1 if the job was completed, 0 if it failed, and 5 if cancelled. If information about chekcpointing or swapping is included, other values are also possible. **See usage note below.** This field is meaningless for models, so would be -1. **Usage of the Status field** The main usage of the status field is to note the job's status. This isn't as straightforward as it sounds. The simple case is jobs that complete normally, and have status 1. The harder case is jobs that don't complete normally. This can happen for several

ID	SWF	GWF	Info
			reasons: 1. The job failed (e.g. segmentation fault). This is given status 0. 2. The job was cancelled by the user (like ^C in Unix). This is given status 5. Note that cancelled jobs may have positive runtimes and processors if cancelled after they started to run, or 0 or -1 if cancelled while waiting in the queue. 3. The job was killed by the system (e.g. because it exceeded its requested run time). This may be given different status values in different logs; it will typically be 0 or 5, but might also be 1. Note also that the distinction between failure / cancellation / killing is not necessarily accurate, as the distinction typically does not appear in the original logs. If a log contains information about checkpoints and swapping out of jobs, a job can have multiple lines in the log. In fact, we propose that the job information appear *twice*. First, there will be one line that summarizes the whole job: its submit time is the submit time of the job, its runtime is the sum of all partial runtimes, and its code is 0 or 1 according to the completion status of the whole job. In addition, there will be separate lines for each instance of partial execution between being swapped out. All these lines have the same job ID and appear consecutively in the log. Only the first has a submit time; the rest only have a wait time since the previous burst. The completed code for all these lines is 2, meaning ``to be continued''; the completion code for the *last* such line is 3 or 4, corresponding to completion or being killed. It should be noted that such details are only useful for studying the behavior of the logged system, and are not a feature of the workload. Such studies should ignore lines with completion codes of 0 and 1, and only use lines with 2, 3, and 4. For workload studies, only the single-line summary of the job should be used, as identified by a code of 0 or 1.

<table>
<tr><th>ID</th><th>SWF</th><th>GWF</th><th>Info</th></tr>
<tr><td></td><td></td><td></td><td>To summarize, the status field codes are (or should be) as follows:

<table><tr><td>0</td><td>Job Failed</td></tr><tr><td>1</td><td>Job completed successfully</td></tr><tr><td>2</td><td>This partial execution will be continued</td></tr><tr><td>3</td><td>This is the last partial execution, job completed</td></tr><tr><td>4</td><td>This is the last partial execution, job failed</td></tr><tr><td>5</td><td>Job was cancelled (either before starting or during run)</td></tr></table>
GWA: need to agree on a new format for this field.</td></tr>
<tr><td>12</td><td>User ID</td><td>UserID</td><td>SWF: a natural number, between one and the number of different users.
GWF: a string, e.g., A.Iosup.</td></tr>
<tr><td>13</td><td>Group ID</td><td>GroupID</td><td>SWF: a natural number, between one and the number of different groups. Some systems control resource usage by groups rather than by individual users.
GWF: a string, e.g., PDS.</td></tr>
<tr><td>14</td><td>Executable (Application) Number</td><td>ExecutableID</td><td>SWF: a natural number, between one and the number of different applications appearing in the workload. in some logs, this might represent a script file used to run jobs rather than the executable directly; this should be noted in a header comment.
GWF: a string. Used to categorize or describe the application, or can be simply a counter that distinguishes all the applications that run on the system. A tentative string format is ExecutableName,ExecutableVersion,ExecutableParams.</td></tr>
<tr><td>15</td><td>Queue Number</td><td>QueueID</td><td>SWF: a natural number, between one and the number of different queues in the system. The nature of the system's queues should be explained in a header comment. This field</td></tr>
</table>

ID	SWF	GWF	Info
			is where batch and interactive jobs should be differentiated: we suggest the convention of denoting interactive jobs by 0. **GWF: a string.**
16	Partition Number	PartitionID	SWF: a natural number, between one and the number of different partitions in the systems. The nature of the system's partitions should be explained in a header comment. For example, it is possible to use partition numbers to identify which machine in a cluster was used. **GWF: a string.**
17	Preceding Job Number	n/a	SWF: this is the number of a previous job in the workload, such that the current job can only start after the termination of this preceding job. Together with the next field, this allows the workload to include feedback as described below. **GWF: Superseded by JobStructure and JobStructureParams.**
18	Think Time from Preceding Job	n/a	SWF: this is the number of seconds that should elapse between the termination of the preceding job and the submittal of this one. GWF: not needed, as it can be computed using also the JobStructure and the JobStructureParams fields.
19	n/a	OrigSiteID	A string. Used to categorize or describe the site from which the job originated, or can be simply a counter that distinguishes all the sites in the system.
20	n/a	LastRunSiteID	A string. Used to describe the last site in which the job run.
21	n/a	JobStructure	{Structure, one of **COMPOSITE** or **UNITARY**}**:**{Composition Type, one of **BoT**, **DAG**, **DCG**, **Other**}

<table>
<tr><th>ID</th><th>SWF</th><th>GWF</th><th>Info</th></tr>
<tr><td>22</td><td>n/a</td><td>JobStructureParams</td><td>
<table>
<tr><td>Composition Type</td><td>Params format</td></tr>
<tr><td>BoT</td><td>Bag of Tasks</td></tr>
<tr><td>DAG</td><td>Directional Acyclic Graph
DAGPrev,DAGNext
▪ DAGPrev - A list of comma-separated indexes, describing the preceding jobs;
▪ DAGNext - A list of comma-separated indexes, describing the following jobs.</td></tr>
<tr><td>DCG</td><td>Directional Cyclic Graph</td></tr>
<tr><td>Other</td><td>Other types
ExtensionID,Params
The ExtensionID must be unique, and registered with the Grid Workloads Archive.</td></tr>
</table>
</td></tr>
<tr><td>23</td><td>n/a</td><td>Used Network</td><td>in kilobytes/s. This is again the average per processor.</td></tr>
<tr><td>24</td><td>n/a</td><td>Used Local Disk Space</td><td>in megabytes. This is again the average per processor.</td></tr>
<tr><td>25</td><td>n/a</td><td>Used Resources</td><td>List of comma-separated ResourceDescription:Consumption.
Both ResourceDescription and Consumption are strings. The ResourceDescription must be described within the GWF trace, or be registered with the Grid Workloads Archive.</td></tr>
<tr><td>26</td><td>n/a</td><td>ReqPlatform</td><td>CPUArchitecture,OS,OSVersion, e.g., x86,Linux,FedoraCore3/9.0, x86,Linux,7.2</td></tr>
<tr><td>27</td><td>n/a</td><td>ReqNetwork</td><td></td></tr>
<tr><td>28</td><td>n/a</td><td>Requested Local</td><td></td></tr>
</table>

<table>
<tr><th>ID</th><th>SWF</th><th>GWF</th><th>Info</th></tr>
<tr><td></td><td></td><td>Disk Space</td><td></td></tr>
<tr><td>29</td><td>n/a</td><td>Requested Resources</td><td>Refers to other requested resources, or other request restrictions.
<table>
<tr><th>Request Type</th><th>Information and example</th></tr>
<tr><td>Site</td><td>Request that the jobs run at a specific site:
Site=mysite.com</td></tr>
</table></td></tr>
<tr><td>30</td><td>n/a</td><td>Virtual Organization ID</td><td>The VO ID, e.g., DAS-2/TU Delft; a string.</td></tr>
<tr><td>31</td><td>n/a</td><td>Project ID</td><td>The project ID, e.g., DutchGrid; a string.</td></tr>
</table>

Extensions:

- co-allocation: fixed, non-fixed, semi-fixed (see syntax for Koala's logs) ;
- job flexibility;
- checkpointing;
- migration;
- reservations;
- failures: FailureOrigin (Infrastructure, Middleware, Application, User);
- economic aspects: price, utility function, etc.

Notes:

- there can be 0-CPU jobs (data transfers);

3. Related work

JSDL – Job Submission Description Language, JSDL-WG, GGF [https://forge.gridforum.org/projects/jsdl-wg/]
UR – Usage Record Format

References

1. Steve J. Chapin, Walfredo Cirne, Dror G. Feitelson, James Patton Jones, Scott T. Leutenegger, Uwe Schwiegelshohn, Warren Smith, and David Talby, ``Benchmarks and Standards for the Evaluation of Parallel Job Schedulers". In Job Scheduling Strategies for Parallel Processing, D. G. Feitelson and L. Rudolph (Eds.), Springer-Verlag, 1999, Lect. Notes Comput. Sci. vol. 1659, pp. 66-89.
2. Dror G. Feitelson et al., The Parallel Workloads Archive, [Online] Available: http://www.cs.huji.ac.il/labs/parallel/workload/ . Nov 2006.