

D2K Web Service Deployment: Resource Usage Accounting

Robert E. McGrath, Emily Wu, Doru Marcusiu, Andrew Shirk

11 April 2006

Revised 18 May 2006

1 Introduction

This note discusses a proposal for resource usage accounting in the D2K Web Service. The D2K Web Service can be implemented in a variety of topologies. In general, the service has four tiers, as shown in Figure 1. These systems may be dedicated to a particular application or shared by several.

In this architecture, the main resource usage for a job will be on the D2K Server (Server 3 in Figure 1) and/or the external resources (Oracle in Figure 1). Therefore, these resources are the primary targets for accounting/charging. However, the D2K Web Service (Server 2 in Figure 1) manages the D2K jobs, so it is the natural location to collect and manage usage information. Furthermore, the servlets (Server 1) are the only components that will be able to collect user ID and other information about the client and users. Therefore, resource accounting actually must be a collaboration between the tiers, with the data collected and disseminated (e.g., as logs) at Server 2.

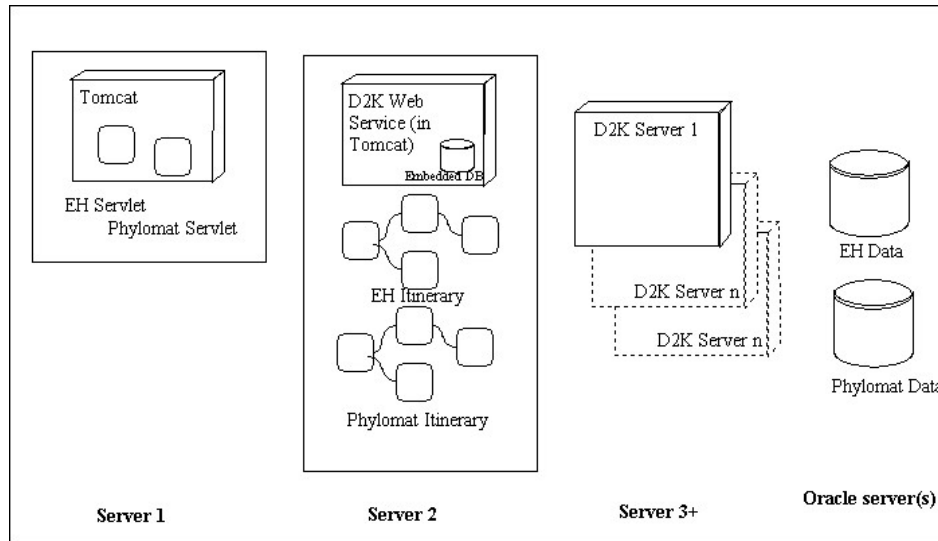


Figure 1

The remainder of this note summarizes the resource usage policy, usage questions that will be answered, and data that will be collected. Section 3 presents a definition of the logs that will be created by the service. Section 4 discusses additional technical issues.

2 Resource Usage Policy

In the initial deployment of D2K Web Service we propose that the only charges are for executing the jobs, i.e., running on Server 3 in Figure 1. The Web Services (Server 1 and Server 2) will not be charged, and any external resources such as databases will be charged according to policies

for that resource. (For example, if a database server has usage accounting, the D2K module will accrue charges on that server through a separate mechanism.) Table 1 summarizes these policies.

Table 1. Summary of Resource Usage Policy

Tier	Proposed Policy
Servlets (Server 1)	No charging.
D2K Web Service (Server 2)	No charging.
D2K Servers (Server 3, et al.)	Charge for usage. See below.
Other services (Oracle, HPC resource, etc.)	Charge according to policies for that service.

In the initial deployment of the D2K Web Service we will provide the service to any users, free of charge. The resource usage will be tracked by activity (which itinerary ran) and by jobs. In the initial deployment, individual users will *not* be authenticated or charged. Effectively, there will be one account for Evolution Highway, and one for Phylomat.

In future work, we will add options for “community” logins and for individual logins.

3 Accounting Data

3.1 Identification of the Required Data

Based on discussions with PI, we need to be able to answer the following questions about the use of the production systems.

1. What was the total usage of the system
2. What was the usage for each service/community (i.e., Evolution Highway vs Phylomat)
3. What was the usage for each job, where a job is defined to be all usage associated with the request for execution of one or more itineraries constituting a single logical job?

In addition, it would be desirable to answer one more question:

4. Identify usage by individual users, an identity could be a source IP address, an email address, a user name, etc.?

We have identified the information that should be collected and reported for each D2K job. Table 2 lists these items. This data is readily available from the current server, although we need to report the data in a convenient format.

The third column in Table 2 shows that this information is sufficient to answer all four questions required.

Table 2. Data collected by D2K Web Service

Data Item	Per Job Record	Answers Question #	Comments
1	Unique Job ID	3	Is there a standard that should be followed?
2	NCSA user identity	2	Initially, this will be fixed, e.g., a hard coded ID for Evolution Highway or Phylomat.
3	Allocation to charge (NCSA PSN)	2	Initially, this will be assigned to the community.
4	Wall clock time	1	Computable from (stop time – start time)
5	Execution host name	1	
6	Job start time	1	
7	Job end time	1	
8	Number of processors		May need to distinguish hosts + processors on host.
9	Submit Host	4	The name of the host submitting the itinerary execution request. This can be used to approximately distinguish users.
10	Usage type	3	A static label describing the usage as a D2K service

3.2 Log Records

The data will be logged by the D2K Web Service. The logs will conform to the NCSA Account Usage Data Format, Version 1.01 [1]. The log format consists of a header record, one or more data records, and a trailer record.

3.2.1 Logging Strategy

The D2K Web Service will generate log files that conform to the NCSA standard. Each log file will contain one record per D2K Job. A log file for a given period would have all the use of a give service for that period, by all users.

The trailer record would be the total use of the service for that period.

The header record would contain an ID for the D2K Web Service.

3.2.2 Data Records

Table 3 shows the format of the data record for one D2K job. The records are text, as described in [1].

Table 3. Data Record

Field	Type*	Value	Definition
“user”	string	“unix login name”	A reserved name for the service, e.g., “d2kehws”. TBD.
“name”	string	“UNIX login name”	From above.
“psn”	string	Project identifier	A PSN will be assigned for each service. All records will have the PSN for the service.
“queue”	string	One of: { “interact”, “batch”, “batch_queue_name” }	** need to define a new value for this field
“cpu”	Number	CPU time in hours	Wall clock time of the job. (see Table 2.)
“mem”	Number	Average memory use (MB)	NULL
“conn”	Number	Connect time in hours	NULL
“su”	Number	Service Units in hours	** need to define formula for calculating SU.
“concurrency”	Number	Number of processors	NULL
“service_type”	String	One of {??}	** Need to define a new service type.
“sess_id”	Number?	Unique number	** Need to define this number. D2K has a unique Job ID.

*All data is represented as strings. Numbers are represented in scientific notation.

** These fields need to be defined.

3.2.3 Header and Trailer Records

The header record records the source and time period covered by the log. Table 4 shows the format of the header records.

Table 4. Header Format

Field	Type*	Value	Description
1	String	“Charges” “Refund”	Always must be “Charges”
2	Char*2	machine ID (??)	** Designate a machine ID for each D2K service? TBD
3	Date, yyyyymmdd:hmmss	Beginning of accounting period	
4	Date, yyyyymmdd:hmmss	End of accounting period	
5		spare	
6	string	“SESS_ACCT” “PROC_ACCT”	** ?? Need new category?

*All data is represented as strings. Numbers are represented in scientific notation.

** These fields need to be defined.

The trailer record is a summary. Table 5 shows the content of this record.

Table 5. Trailer Format

Field	Type	Value	Definition
“totalcpu”	Number	Total in hours	Sum of “cpu” for all data records
“totalmem”	Number	Total memory used (MB)	NULL
“totalconn”	Number	Total connect time (hours)	NULL
“totalsu”	Number	Total Service units (hours)	Sum of “su” for all data records

*All data is represented as strings. Numbers are represented in scientific notation.

** These fields need to be defined.

4 Technical Issues

This section reviews two technical issues that need to be resolved.

4.1 Data Collection Process

1. Should there be one log per service? Or all services mixed together?
2. What period should be covered by each log?

4.2 Fields Yet To be Defined

This note identifies fields that must be defined.

In the header record:

1. machine id—one code for each service? Or for the physical server? Or what?
2. “SESS_ACCT” | “PROC_ACCT” – what should be in here?

In the Data Record:

1. “user ID” for each service
2. PSN for each service
3. What “queue” should be—define a new category?
4. Formula for calculating SU
5. “service_type” – what should this be?
6. “sess_id” – Should be use a D2K Job ID?

4.3 Additions to the Log Format Specification

The specification for the log format has examples but no complete specification for the format of numbers. It would be useful to know what “scientific notation” must be, to assure that we produce conformant records.

Section 4.2 identifies several records what may need new values (e.g., service_type). We need to determine these values and update the specification.

5 Other Resource Usage Not Covered Here

The plan presented here ignores the resource usage of the servlets (Server 1 in Figure 1) and the Web Service itself (Server 2 in Figure 1). This policy is based on the belief that these servers are not expensive and would not be worth the effort required to account for usage. In addition, it is not clear what metrics should be used on these systems. This decision may be revisited as we gain the operational experience.

This plan does not specify how external resources will be accounted. We assume that such resources have their own accounting processes, which will be used to charge the itineraries as needed. For example, an Oracle server might be configured to charge users for access. In this case, the itinerary will need to connect with the appropriate account, and the account usage will be reported via Oracle.

We have proposed to charge for wall-clock time rather than ‘CPU usage’ or memory use. This is justified because the servers are (usually) dedicated resources, running only one job at a time. In this case, the wall clock is a valid measure of overall usage.

In addition, the design of the D2K Server (Server 2 in Figure 1) makes it difficult to collect ‘per job’ CPU and memory usage. The user’s job runs in one or more threads of the D2K Server, so it would be necessary to measure processor time and memory for the set of threads associated with the job. This is not easy technically. Furthermore, it is difficult to clearly identify the work that is done on behalf of a given job, e.g., for I/O, class loading, etc.. If we need to have more detailed accounting in the future, it may require redesign of the D2K Server.

6 Future Developments

This note has focused on the initial deployment of the D2K Web Service. We anticipate further development of the D2K Server and Web Service.

6.1 Authentication

Future services will require the ability to authenticate users, either community IDs or individual accounts. This will be done by integrating NCSA portal technology, which will enable, for example, the use of credentials from MyProxy. This will open the way to the use of HPC and Grid resources.

6.2 Scheduler

The current implementation of the D2K Web Service and D2K Server have rudimentary scheduling policies. In the future, it may be necessary to implement priority scheduling (e.g., certain users may preempt others), and limits (e.g., certain users may only run for 3 minutes or less). This will address the needs of communities, and also may deal with denial of service and other security issues.

6.3 Uploading

The initial deployment allows users to run itineraries, but they cannot modify or install itineraries. It is likely that future applications will be designed to let users “up load” modules or itineraries, and perform other administrative operations.

The D2K service can support this operation today, but we do not know all the implications for security. One likely approach would be to give a PI or community admin certain rights, while ordinary community members may only run.

7 Summary

This note summarized the proposed resource accounting for the D2K Web Service. The D2K Web Service will be responsible for managing the usage records for each job, and for reporting. The D2K Web Service will be responsible for obtaining the correct NCSA IDs and PSNs, according to standard mechanisms, TBD.

We propose to use elapsed time on the D2K server as the measure of usage. The collected data will be logged or reported in a standard way, TBD.

In the initial deployment, the services will be open to any user without registration or authentication. In future development, we will add options to authenticate to a group or individual account. This may require additional log records, e.g., to track group and individual user id. These developments are yet to be determined.

The scheduling capabilities of the D2K Web Service may need to support new features, such as priorities and resource limits. This will require development.

8 References

1. NCSA, "NCSA Account Usage Data Format; Version 1.01", May 2006.