How Dynamic is the Grid?
Towards a Quality Metric for Grid Information Systems

Laurence Field
CERN, Geneva, Switzerland
Laurence.Field@cern.ch

Rizos Sakellariou
The University of Manchester, Manchester, UK
rizos@cs.man.ac.uk

***Abstract*—Grid information systems play a core role in today's production Grid Infrastructures. They provide a coherent view of the Grid services in the infrastructure while addressing the performance, robustness and scalability issues that occur in dynamic, large-scale, distributed systems. Quality metrics for Grid information systems are required in order to compare different implementations and to evaluate suggested improvements. This paper proposes the adoption of a quality metric, first used in the domain of Web search, to measure the quality of Grid information systems with respect to their information content. The application of this metric requires an understanding of the dynamic nature of Grid information. An empirical study based on information from the EGEE Grid infrastructure is carried out to estimate the frequency of change for different types of Grid information. Using this data, the proposed metric is assessed with regards to its applicability to measuring the quality of Grid information systems.**

***Keywords*-Grid Information Systems, Information Freshness**

I. Introduction

Grid information systems enable users, applications and services to discover which Grid services exist in a Grid infrastructure and further information about their structure and state [5]. Information describing each Grid service is provided by the service itself. Hence, the Grid service, in terms of Grid computing, is the primary information source. The large number of distributed information sources in a typical Grid infrastructure and the requirement from Grid information systems to resolve queries that may take into consideration multiple information sources make it difficult for Grid information systems to find information on-the-fly, that is only upon request. For this purpose, implementations of Grid information systems [7] employ caching [15] and/or quasi-copies [2], [12] in order to address performance, robustness and scalability issues. Using such techniques introduces of a level of uncertainty, as it is not possible to know if the result of a query, as provided by the Grid information system, actually reflects the content of the information source without contacting the information source itself.

In existing literature [4], [5], [13], [14], the concept of *freshness* is used to give an indication of this uncertainty regarding the values maintained by a Grid information system. Thus, a cache (or quasi-copy) is considered to be *fresh* the instant it has been synchronized with the information sources. Over time, as some information sources change and the cache is not refreshed, the contents of the cache become less fresh until the cache is eventually considered to be *stale*. Elaborating further on the concept of freshness, a particular value in a cache (or quasi-copy) is fresh if it coincides with the value on the information source. The percentage of all values in the cache that are fresh indicates the degree of freshness of the cache.

Measuring the degree of freshness of a cache (or quasi-copy) at any given moment is not an easy task. This would require checking each value stored in the cache and whether the copy in the cache coincides with the real value of the information source. Given the large number of information values normally stored in a cache, an exhaustive check would incur a high overhead and would be impractical. Thus, to come up with a quantitative description of freshness for a cache (which is essentially a quality metric for a Grid information system) one has to resort to models that capture information source changes.

One possible approach to model freshness is to use *age*; the time elapsed since the last refresh of the cache (or quasi-copy). The older the copies are the less fresh they will be. However, this by itself is not enough. Since the early days of the Grid, it has been recognized that not all Grid information changes with the same frequency; some information has been described as *static* and other information as *dynamic* [5]. This affects the temporal relationship with freshness. Early work has used simulation to model freshness in relation to the frequency of changes and updates of the cache [13, Sec. 4.7.1], but no previous work has tried to model freshness in the context of actual Grid information.

In this paper, we carry out an empirical study, which is trying to understand the rate of change for different types of Grid information and use this understanding in a model for freshness. For the latter, we turn to related work that has been done to measure how up-to-date a Web search engine is. In [3], the concept of (α, β)-currency has been defined to provide such a measure. Roughly speaking, this provides a probability, α, that a search engine is current (up-to-date or fresh) relative to a grace period, β, for a given web page. Estimating this probability for different types of Grid information could be used to indicate freshness and, hence, provide a quality metric for Grid information systems. In our study, we use the EGEE project infrastructure [9] as our experimental testbed and we carry out an empirical estimation of the frequency of changes for different types of Grid information, which we then feed into a model for (α, β)-currency. To the best of our knowledge, this is the first time that such an empirical study is carried out and used to provide a quality metric for Grid information freshness and Grid information systems.

The rest of the paper is structured as follows; Section II provides some background on the impact of the Grid information model, it illustrates the type of changes for Grid information and describes a model for the (α, β)-currency; Section III presents an empirical study using the EGEE project Grid infrastructure, which calculates the frequency of change for the different types of Grid information; Section IV evaluates evaluates these results with respect to (α, β)-currency and its model, and some concluding remarks are given in Section V.

II. Background

A. Information Model

Information in a Grid information system conforms to an information model. This is described using an entity-relationship model where the entities are different objects each one comprised of a set of attributes. The information model is a key part of any study, such as the one described in this paper, as it describes the information that changes. Whereas a Web page is considered to have changed if it has been updated, the information model used in Grid computing enables changes to be observed with more granularity as the frequency of change per object type in the entity-relationship model can be measured. These measurements would be valid only for that specific information model and could not be used to compare with measurements obtained for a different information model. Although it may sometimes be possible to translate information from one model into another model [8], the composition of the objects may alter and hence the frequency of change for those objects may be affected.

In one of the early papers on Grid information systems [5], information was described as being relatively static or more dynamic. The concept of static and dynamic information has since been widely used by the Grid community even though a detailed definition does not exist and its use may be misleading. In this paper, we assert that all information is dynamic and it is the frequency of change that defines the relative dynamic nature of the object types.

B. Taxonomy of Changes

If the information model describes object types, a taxonomy is required to describe the types of change that an object may undergo. An instantiation of an object type, such as a description of a Grid service, can undergo three transitions. A new Grid service joining the infrastructure would result in an object being added; the decommissioning of a Grid service would result in the object being deleted; and, during the lifetime of a Grid service, the object may be modified to reflect changes to the service. Objects that are modified can be modified in three ways. A new attribute can be added to the object, an attribute can be deleted from the object, or an attribute's value can be modified. One additional transition that may need to be considered is the re-addition of an object. This can occur when a Grid information system implementation experiences failures, such as broken network connections or power outages at computing centres. Another reason for such behaviour is during a planned (or unplanned) intervention to the service or information source. In such a scenario, the service still exists but may not be usable, and whether the object should be removed from the information system or its status should be updated is an implementation detail. However, it is important to monitor such transitions in order to avoid double-counting of the added and deleted transitions. By considering all these possible changes, a classification of changes can be seen in Figure 1.

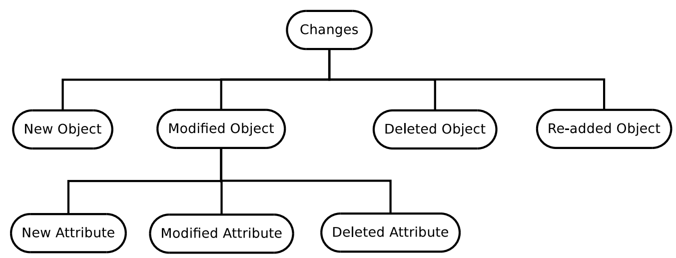

Figure 1. A taxonomy of possible changes to objects in a Grid information system.

Using the taxonomy described in Figure 1, the frequency of change for each type of change and object type can be calculated. The frequency, f, is defined as the number of changes, n, over a period of t time units, that is,

$$f = \frac{n}{t}. \quad (1)$$

C. Measuring information decay

As already mentioned, the concept of (α, β)-currency was introduced in [3] as a measure of how up-to-date a search engine is with respect to a changing collection of Web pages. The cached data for a given Web page that is used by a search engine data is said to be β-current if the Web page has not changed between the last time it was indexed and β time units ago. A search engine for a collection of pages is then said to be (α, β)-current if a randomly chosen page in the collection has a search engine entry that is β-current with probability at least α. To quote an example from [3], a daily newspaper is (0.90, 1 day)-current when it is printed, meaning that the newspaper has at least 0.9

probability of containing one day current information on topics of interest to its readers. By considering the analogy between a Web page and a Grid information source (they are both information sources) and a Web search engine and a Grid information service (they are both snapshots; the former of the Web, the latter of the Grid), it should be possible to apply the concept of (α, β)-currency on the Grid to describe how up-to-date (or fresh) the information for a given object of a Grid information service is.

Applying the concept of (α, β)-currency, requires us to specify an approach to calculate the probability α, that what is contained in the snapshot for some information source is current, relative to a grace period β. In [3], the changing nature of web pages is modeled using the same ideas underpinning reliability theory [11] in industrial engineering. When a component in a system fails, it is replaced and the period between when it was replaced and when it failed is defined as the lifetime.

For Web pages, lifetime, defined as the period between changes of a web page, can be easily found by using the LAST-MODIFIED attribute in the HTTP header. However, as no such attribute is available in Grid information models or Grid information system implementations, this approach cannot be used on the Grid. Instead, on the Grid, we need to infer lifetimes (and, essentially, rate of change) by observing the number of changes for a specific object over a period of time. Then, by feeding this information into an exponential decay model, as described in [3] and imminently, we can approximate the probability α that some information (i.e., a specific object type) obtained from the Grid information system is current with respect to the information source, as a function of a grace period β. The model assumes that changes are events controlled by an underlying Poisson process, where the probability of observing a change at any given time does not depend on previous changes. Given a time period t and a change rate λ, the probability that a change is observed is given by

$$P(\text{change observed}|\lambda, t) = 1 - e^{-\lambda t}. \tag{2}$$

Equation 2 expresses the probability that a change has occurred. Therefore, the probability that a change has not occurred is given by

$$P(\neg(\text{change observed})|\lambda, t) = e^{-\lambda t}. \tag{3}$$

In order to apply the same technique to Grid information systems, it must be understood if they can be modelled using the same process.

Consider object x at the information source and its image x' in the Grid information system. When the information is obtained from the information source at time $t = 0$, it is certain that image x' is identical to the original object x. Assuming that all information will change if the observation period is sufficiently large, we can also assert that image x' is not identical to the original object x when $t = \infty$. During the period $0 < t < \infty$ the probability P that image x' is identical to the original object x is described by the function $f(t)$. These boundary conditions can be expressed as follows:

$$P(x' = x, t) = \begin{cases} t = 0 & P = 1 \\ 0 < t < \infty & P = f(t) \\ t = \infty & P = 0 \end{cases} \tag{4}$$

Assuming that each object x is of the same type, the probability that the object will experience a certain change type in time period Δt is identical for all objects, that is,

$$P(x \to \neg x', \Delta t) = constant \tag{5}$$

As the probability of change is constant for all objects of the same type, the change events can be modelled by a Poisson process. The probability of an object changing in time period $\triangle t$ can be found by dividing the total number of changes n during that period by the total number of objects N, that is,

$$P(x \to \neg x', \Delta t) = \frac{n}{N} = \lambda \tag{6}$$

Combining Equation 1 with Equation 6 we obtain

$$\lambda = \frac{f}{N} \tag{7}$$

assuming that the rate refers to the same period of time used to calculate frequency (this means that $t = 1$ in Equation 1).

Using Equation 3 and the value for λ given by Equation 7, it is possible to find the probability α that Grid information is current, for a specific grace period β for a randomly selected object by knowing the expected frequency of change. The value of β is equivalent to t in Equation 3 and the probability α is equivalent to $P(\neg(\text{change observed})|\lambda, t)$. Hence, the fundamental equation underpinning our model is

$$\alpha = e^{-\lambda\beta}. \tag{8}$$

III. Inferring the Frequency of Change for Grid Information

As mentioned before, to infer frequency of change for Grid information, we need to measure the number of changes over a (possibly long) period of time. For this purpose, we used real data from the Enabling Grids for E-Science (EGEE) project infrastructure [9] which operates the largest multi-disciplinary Grid infrastructure in the world.

The EGEE Grid information system has adopted the GLUE 1.3 schema [1] as its information model. This model is composed of 15 object types that are split into three main groups: core objects (e.g., Site and Service); compute service related objects (e.g., Cluster); and, storage service related objects. In addition, two relationship objects also exist that describe the relationship between the compute service and the storage service. A list of these objects, listed by group, is given in Table I.

Table I
DESCRIPTION OF THE 15 OBJECT TYPES IN THE GLUE 1.3 INFORMATION SCHEMA.

Object Type	Description
Site	The organization that installed and manages the Grid service
Service	An abstracted, logical view of a Grid service
ServiceData	Key/value pair extension for the service object
Cluster	A computing cluster managed by a single batch system
SubCluster	A homogeneous set of machines within the cluster
CE	A single queue within the batch system with associated policies and state
VOView	The policies and state of a queue from the VO's perspective
Location	The name, version and path of installed software
SE	A file-based storage system
SA	A logical storage area within the storage system
VOInfo	VO specific information for a storage area.
SEAcessProtocol	A file transfer protocol supported by the storage system
SEControProtocol	A control protocol supported by the storage system
CESEBind	An association object to describe the CE, SE relationship
CESEBindGroup	A grouping object for all SE relationships with a CE

The EGEE information system has a hierarchical structure with three levels. The fundamental building block used in this hierarchy is the Berkeley Database Information Index (BDII) [6]. The BDII can be visualized as an LDAP database that is periodically refreshed. The first level of the hierarchy is the resource-level which is co-located with the Grid service and provides information about that service. The second level is the site-level which aggregates the information from all the resource-level BDIIs running at a particular site. The third level is the top-level where all the information from all the site-level BDIIs is aggregated and hence contains information about all Grid services in the infrastructure. As the top-level BDII contains the global view of the infrastructure, it can be used to provide a snapshot of the information in the information system.

Snapshots can be obtained periodically. However, the use of snapshots to observe changes based on periodical sampling has a potential limitation. Snapshots cannot detect changes within the sampling period if such changes revert back to the same value. For example, if a new object was created and then deleted between two consecutive snapshots, this would be missed. An indication of whether sampling is affected by such scenarios can be obtained empirically. If the (empirically estimated) lifetime of the objects is significantly greater than the sampling period, then the chance of such a scenario should be negligible. For objects where this is not the case, a shorter sampling period is required. For this reason, two sampling methods were used in our study, with sampling periods determined empirically, as we also elaborate later. In the first method, daily snapshots were obtained throughout a month, by querying the top-level BDII for all objects and storing the output as a file. In the second method, snapshots every 5 minutes for the duration of a single day were obtained by enabling the archive function of BDII which stores a snapshot every time the database is refreshed.

Daily snapshots were recorded throughout March 2010. During this period, there was an average of about 100,900 entries (i.e., different objects) in each snapshot, which represented an average data size of about 76.1Mb in LDIF [10]. The composition of this information per object type is shown in Figure 2. It shows that most of the information, 45.3% by number of entries and 35.1% by data size, describes installed software (Location). The second largest, 9.5% by number of entries and 10.7% by data size, is the VOView, which describes the state of a queue in a batch system from the VO's perspective. The information describing services, 3.8% by number of entries and 6.9% by data size, and sites, 0.35% by number of entries and 0.39% by data size (included under 'Other' in the figure), is by comparison quite small.

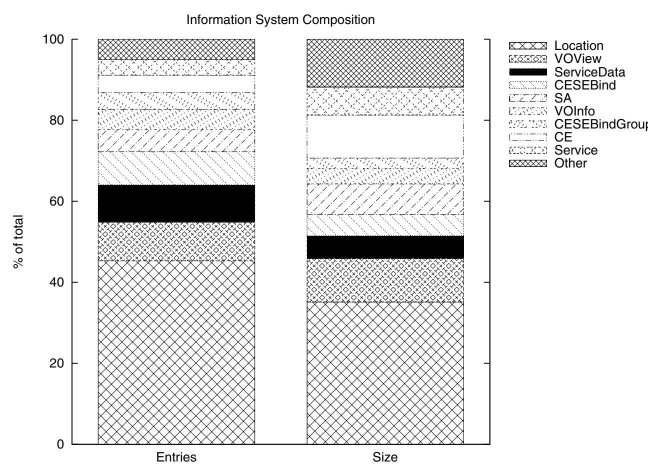

Figure 2. The composition of information (for different object types) in the Grid information system by number of entries and data size.

The daily snapshots were compared to find differences and, hence, calculate the number of changes that had occurred. These changes were grouped according to each major type of change (added, deleted, modified), as outlined in the taxonomy in Section II, and object type. The average number of objects changed per day for each object type and each of the three types of change is shown in Table II. The second column of the table shows the total number of objects in each snapshot on average. The next three columns show, for each object type, the number of objects that changed each day on average for each different type of change. Clearly, by using Equation 1, the frequency of change for different object types can be calculated. However, before doing so, it is useful to check whether, for some object types and types of change, there are indications that the sampling period of a day may be too high and a smaller sampling period may be necessary to obtain more precise numbers.

Table II
AVERAGE NUMBER OF OBJECTS CHANGED PER DAY FOR EACH OBJECT TYPE AND TYPE OF CHANGE.

Object Type	ΣObjects	Added	Deleted	Modified
CE	4271	11.20	7.43	3484.0
CESEBind	8254	22.23	13.80	8.10
CESEBindGroup	4318	13.93	7.33	2.13
Cluster	599	1.33	0.74	1.83
Location	45699	320.20	174.06	302.27
SA	5470	24.83	26.41	2571.57
SE	456	1.03	1.07	171.67
SEAccessProtocol	2347	13.37	11.47	6.07
SEControlProtocol	703	2.13	2.24	0.20
Service	3851	15.83	11.60	819.73
ServiceData	9313	49.37	27.37	1.60
Site	351	0.53	0.43	1.10
SubCluster	657	1.97	1.00	129.67
VOInfo	5011	38.87	35.97	1.67
VOView	9603	23.70	15.90	6930.17

For this purpose, we are going to display the results in Table II in a different way by considering the number of objects that changed as a percentage of the total number of objects. The number of objects added and deleted per day as a percentage of the total number of objects for each type is shown in Figure 3. For all object types, this percentage is less than 1% and there are no big variations for different object types. In fact, the slightly higher values for some object types are also due to the fact that there exists an one to many multiplicity between objects in the information model. For example, if a VOView object is added for each VO that is enabled to use a specific Cluster, the addition of one new Cluster object will also result in the addition of multiple VOView objects. Overall, the small percentages suggest that there is a more than 99% chance that an object will not be added or deleted within a day, which, in turn, is an indication of a lifetime significantly longer than the sampling period.

However, the situation is different when we consider the number of modified objects per day as a percentage of the total number of objects for each object type. This is shown in Figure 4 for different object types (x-axis) sorted in descending order of the percentage value. The graph clearly suggests that different object types can be grouped in two categories; those that experience (on average) less than 1% changes per day (which we can call *low frequency object types*) and those that experience over 1% changes per day (*high frequency object types*). A line separating the two categories can be drawn between the Location object type, where 0.66% of objects were changed, and the SubCluster object type, where 19.7% of objects were changed (note that

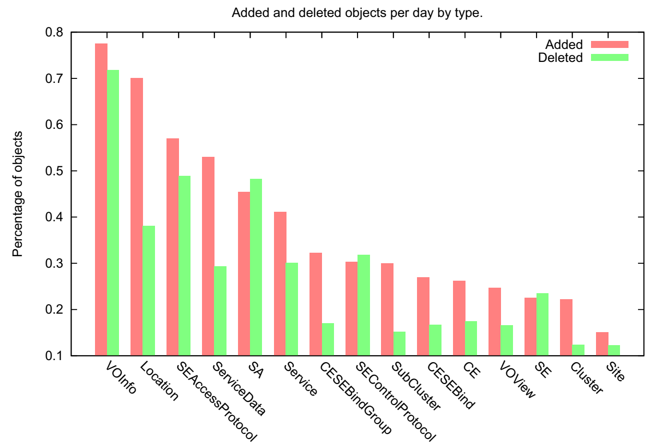

Figure 3. Percentage of added and deleted objects per day by object type.

these percentages can easily be derived from the values in Table II). This separation suggests a possible (measurable) way to differentiate between what has been termed as static and dynamic type of information in the past [5], as noted in Section I, even though we would rather avoid the use of the word ’static’, as it may imply that the information does not change (which is not true). Regarding the category of the so-called high frequency object types, the rather high percentage of objects changed (which at one extreme approaches 82.7% for the CE object type) suggests that the expected lifetime for objects of this type may be smaller than the sampling period. Therefore, it would be prudent to observe the number of changes for objects in this category with a smaller sampling period.

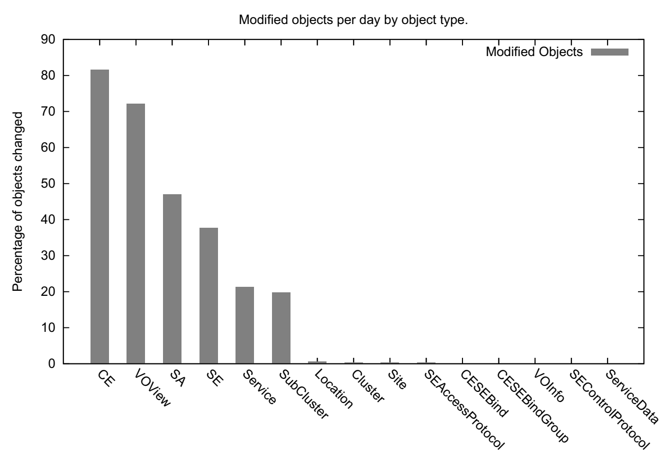

Figure 4. Percentage of modified objects per day by object type.

For the high frequency object types, snapshots, taken every 5 minutes over a single day, were used to investigate their number of changes in more detail. Data from the day of the 28th March 2010 was chosen for further analysis. The same approach as before is used to present the results in Table III. As can be seen, the average number of additions and deletions is negligible. However, even with a 5 minute sampling period, there are still many modifications for these object types (shown as a percentage of the total number of objects for each object type in Figure 5). Looking at the CE and VOView objects, in 5 minutes, 57.4% and 44.3%, respectively, of objects had changed within that period. By comparison, with the daily snapshots this rises to 86.7% and 76.7% for the CE and VOView, respectively.

Table III
AVERAGE NUMBER OF OBJECTS CHANGED EVERY 5 MINUTES FOR EACH HIGH FREQUENCY OBJECT TYPE AND TYPE OF CHANGE.

Object Type	ΣObjects	Added	Deleted	Modified
CE	4307	0.03	0.13	2470.09
SA	5406	0.06	0.32	829.29
SE	450	0.00	0.02	65.49
Service	3900	0.06	0.09	568.19
VOView	9736	0.06	0.25	4319.95

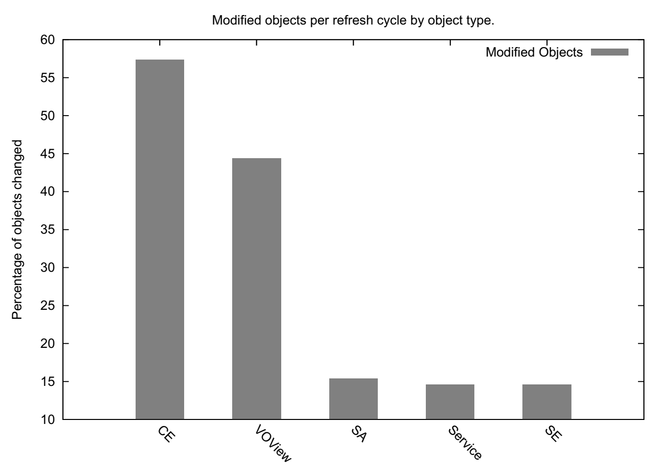

Figure 5. Percentage of modified objects every 5 minutes by object type.

The results of a more in-depth study considering modifications of the VOView object's attributes are displayed in Figure 6. The top three attributes with the highest frequency are the FreeJobsSlots, FreeCPUs and EstimatedResponseTime. These three attributes are all related to the state description for the cluster and are required for workflow scheduling. It may be interesting for future work to consider frequency of changes for specific attributes, but this is beyond the scope of the present study.

This investigation has provided evidence to support a number of remarks with respect to Grid information from the EGEE infrastructure. These remarks can be summarized as follows: (i) Information describing the main entities in a Grid infrastructure, the site and service objects, represents a small proportion, 4.17%, of the total information. (ii) The number

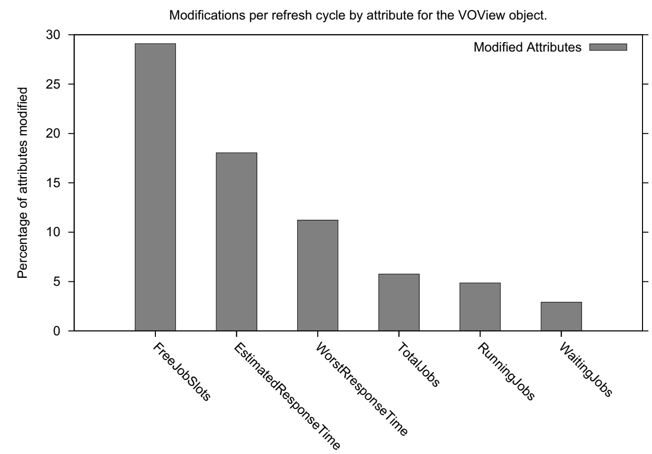

Figure 6. Percentage of modified attributes every 5 minutes for the VOView object type.

of adds and deletes per day is quite low in comparison to the modifications, even for the Location object (recall that most of the information in the snapshots relates to this). (iii) It is misleading to characterize some Grid information as static, as all objects types do experience change; however, the rates of change vary for each object type. (iv) Looking into more detail into high frequency object types, their actual frequency of change may be so high that a 5 minute sampling period may not be small enough to estimate it with good precision. (v) Within these highly dynamic objects, the attributes that mostly contribute to the dynamic nature of the object are related to state information.

Finally, it should be stressed that as this investigation relied on data from a Grid infrastructure following a specific information model (hence, specifying a particular type of objects), these results would not be directly comparable with results obtained from platforms where a different information model was used.

IV. MODELING THE DECAY OF A SNAPSHOT

Using the data for the number of changes obtained in the previous section, the frequency of change for different object types can be calculated using Equation 1. The value for frequency can be used to obtain a value for λ using Equation 7. Then, the value of λ allows us to use Equation 8, which models the probability, α, that a randomly selected object of a specific object type is current after a grace period β.

To support the validity of the decay curve produced by Equation 8, we considered three different cases for further study, which can be regarded as representative. The rationale for choosing these was as follows. First, with respect to changes due to additions and deletions, as we noted from Figure 3, the percentage of changes is similar for all object types. As such, the Service object was chosen as a representative of all object types with additions/deletions. For changes due to modifications, we chose the Site object as a representative of a low frequency object and the VOView object as a representative of a high frequency object. The values of f and λ for these object types and change types were calculated and the results are shown in Table IV. We note that the values for the total number of objects and number of changes for the first two rows are given in Table II, and for the third row in Table III.

Table IV
THE VALUE OF λ FOR THE OBJECT TYPES AND CHANGE TYPES DESCRIBED.

Object	ΣObjects	Changes	f (sec^{-1})	λ
Service	3851	11.6 / day	1.34×10^{-4}	3.5×10^{-8}
Site	351	1.1 / day	1.27×10^{-5}	3.6×10^{-8}
VOView	9736	4320 / 5 mins	14.4	1.48×10^{-3}

The decay curve produced by Equation 8 models the change in probability α that a randomly selected object of a specific object type is current with respect to the grace period β. To compare this with the data obtained from the snapshots, we need to measure the decay of a snapshot over time by comparing subsequent snapshots to the first snapshot. Then, the probability that a randomly selected object of a specific object type is current at a given time t (that is, at time t, it has not changed with respect to the first snapshot) is given by

$$P(x' = x, t) = 1 - \frac{n}{N}, \tag{9}$$

where n is the number of objects (of this type) that are still current at time t and N is the total number of objects (of this type) in the original snapshot.

For the Service object type, a comparison of the probability computed from the snapshots (raw data) with the outcome of the decay model produced by Equation 8 using $\lambda = 3.5 \times 10^{-8}$ (see Table IV) is shown in Figure 7. We can see that the values predicted by the decay curve do indeed correlate with the observed decay pattern. In fact, the standard deviation of the difference between the two is 0.0145.

The same comparison is made for the Site object type and the VOView object type in Figures 8 and 9, respectively. Again, the values predicted by the decay curve correlate with the observed decay pattern from the snapshots. The standard deviation of the difference between the two is 0.00113 for Site and 0.0145 for VOView. We note here that the values for VOView decay very fast (as remarked in Section 3, a large number of changes was observed for this object type). Thus, the 5-minute snapshots used in the previous section would not show the decay very well (they would correspond to only 3 points in the graph, at 0, 300, and 600 seconds). To compare the decay curve with more raw data, we obtained additional snapshots using a sampling period of 30 seconds.

The above suggest that the decay model for a snapshot given by Equation 8 appears to correlate well with real data

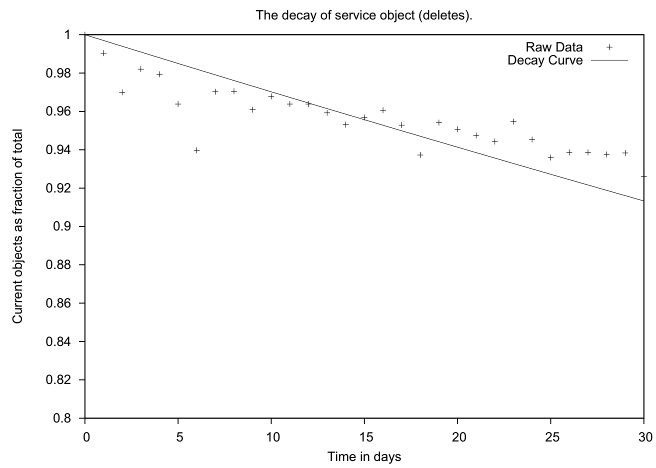

Figure 7. The decay curve for deleted Service objects along with data showing the actual decay of a snapshot.

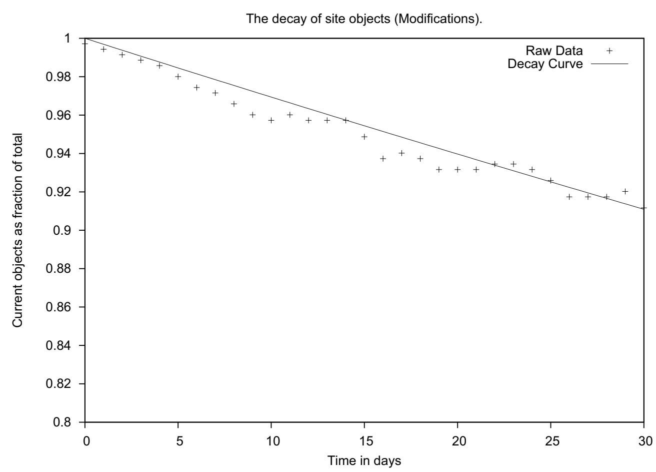

Figure 8. The decay curve for modified Site objects along with data showing the actual decay of a snapshot.

from a production Grid infrastructure. This decay model can therefore be used to describe the (α, β)-currency of Grid information for a specific object type and change type. For example, Grid services are (0.95, 15 day)-current, meaning that a snapshot of Grid services objects has a 0.95 probability of containing current (or fresh) information on Grid services 15 days after it was taken. However, VOView objects are (0.95, 35 second)-current, meaning that a snapshot of VOView objects has a 0.95 probability of containing 35 second current information on the state of a computing service.

Using this metric, information freshness in a Grid information system, and hence the quality of the system itself, can be assessed. By measuring the age of the information, that is, the time period (β) that has passed since the information was refreshed from the information source, the probability (α) that a specific object type is fresh can be calculated using

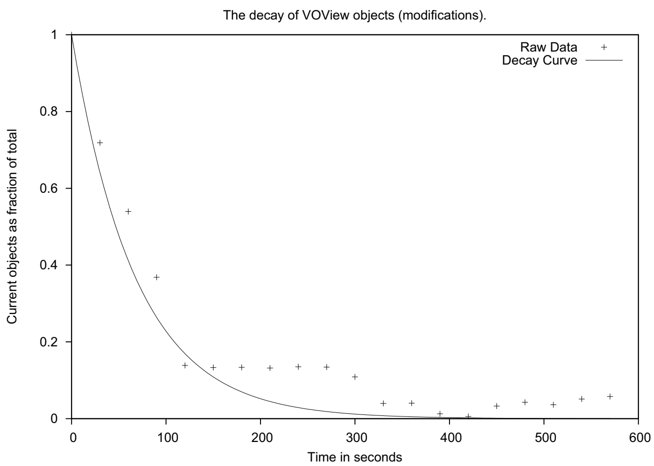

Figure 9. The decay curve for modified VOView objects along with data showing the actual decay of a snapshot

Equation 8. In addition to its value as a quality metric, this probability may also be used to dictate the desired refresh cycle for different types of information.

V. Conclusion

This paper investigated the possibility of applying the concept of (α, β)-currency, from the domain of web research, to Grid information systems. To apply this concept, the changing nature of information within a production Grid information system needed to be understood. Following an empirical study, we demonstrated that the rate of addition and deletion is similar for different object types with the number of additions slightly exceeding the number of deletions, due to infrastructure growth. Modifications can be grouped into low frequency and high frequency modifications. By using data from a production Grid infrastructure, it is possible to measure the frequency of change for the different object types and change types.

A decay model can be used to calculate the probability α that a randomly selected object of a specific object type is current with respect to a grace period β. The model assumes that change events are controlled by an underlying Poisson process, where the probability of observing a change at any given time does not depend on previous changes. This model can be applied to Grid information as the frequency of change is constant for different object types and change types. Using the frequencies of change found by experiment for different object and change types, a value for the decay constant λ can be calculated for each case. Various snapshots were observed to provide experimental data for the decay which was then compared with the curve given by the decay model. The correlation suggests that the model can indeed be used to predict the decay of a snapshot with respect a specific object and change type. As such it is possible to use the concept of (α, β)-currency as a quality metric for Grid information systems.

Future work could try to extend this work by looking into the possibility of deriving single metrics for collections of different object types and changes, possibly by combining different values. There is also scope for more empirical studies in different Grid settings, including studies at a more fine-grain level of detail (e.g., attributes).

References

[1] GLUE schema specification, version 1.3, Jan. 2007. http://glueschema.forge.cnaf.infn.it/Spec/V13.

[2] R. Alonso, D. Barbará, H. Garcia-Molina, and S. Abad. Quasi-Copies: Efficient data sharing for information retrieval systems. *Lecture Notes in Computer Science*, volume 303, 1988.

[3] B. Brewington and G. Cybenko. How dynamic is the web? *Computer Networks*, 33(1-6):257–276, 2000.

[4] J. Cho and H. Garcia-Molina. Synchronizing a database to improve freshness. *ACM SIGMOD Record*, 29(2):117–128, 2000.

[5] K. Czajkowski, C. Kesselman, S. Fitzgerald, and I. Foster. Grid information services for distributed resource sharing. In *Proceedings of the 10th IEEE International Symposium on High-Performance Distributed Computing* (HPDC'01), 2001.

[6] L. Field and M. W. Schulz. Grid deployment experiences: The path to a production quality LDAP based grid information system. In *Proceedings of 2004 Conference for Computing in High-Energy and Nuclear Physics*, pp. 723–726.

[7] S. Fitzgerald, I. Foster, C. Kesselman, G. von Laszewski, W. Smith, and S. Tuecke. A directory service for configuring High-Performance distributed computations. *Proceedings of the 6th IEEE Symposium on High-Performance Distributed Computing* (HPDC'97), pages 365–375, 1997.

[8] M. Flechl and L. Field. Grid interoperability: joining grid information systems. *Journal of Physics: Conference Series*, 119(6):062030, 2008.

[9] F. Gagliardi, B. Jones, F. Grey, M.-E. Bégin, and M. Heikkurinen. Building an infrastructure for scientific grid computing: status and goals of the EGEE project. *Philosophical Transactions. Series A, Mathematical, Physical, and Engineering Sciences*, 363(1833):1729–1742, August 2005.

[10] G. Good. The LDAP data interchange format (LDIF) - technical specification, 2000.

[11] A. Papoulis. *Probability, random variables, and stochastic processes*. McGraw-Hill, Boston, 4th edition, 2007.

[12] F. Stamatelopoulos and B. Maglaris. A caching model for efficient distributed network and systems management. In *Proceedings 3rd IEEE Symposium on Computers and Communications* (ISCC'98), 1998.

[13] S. Zanikolas. *Importance-Aware Monitoring for Large-Scale Grid Information Services*. PhD dissertation, University of Manchester, 2007.

[14] S. Zanikolas and R. Sakellariou. An Importance-Aware Architecture for Large-Scale Grid Information Services. *Parallel Processing Letters*, 18(3):347–370, September 2008.

[15] X. Zhang, J. L. Freschl, J. M. Schopf. A Performance Study of Monitoring and Information Services for Distributed Systems. *Proceedings of the 12th IEEE International Symposium on High Performance Distributed Computing* (HPDC'03), 2003.