RedBlue Consistency - A Survey

John Smith,[1*] Jane Doe,[1] Joe Scientist[2]

[1]Department of Chemistry, University of Wherever,
An Unknown Address, Wherever, ST 00000, USA
[2]Another Unknown Address, Palookaville, ST 99999, USA

[*]To whom correspondence should be addressed; E-mail: jsmith@wherever.edu.

njnjnjnjn This document presents a number of hints about how to set up your *Science* paper in LaTeX . We provide a template file, `scifile.tex`, that you can use to set up the LaTeX source for your article. An example of the style is the special {`sciabstract`} environment used to set up the abstract you see here.

Introduction

As the user base of the internet is increasing, providing good quality of user experience is becoming a challenging task. To deal with this,various service providers are replicating their system states across various geographically diverse sites and when an user accesses the service, it is directed to the closest or the least loaded site. Such applications are now very common. Most of the leading companies like Google,Amazon and EBay - all claim that a slight increase in user-perceived latency translates into concrete revenue loss. So to provide better availability and disaster tolerance, applications replicated user data replicated across various sites. When an user accesses the service, it is forwarded to the closest or the least loades site.

This paper tries to solve the inherent tension between performance between performance and meaningful consistency. The first step towards this goal is to allow various levels of consistency to coexist. Some operations can be executed optimistically,without synchronizing with concurrent actions at other sites, while others require a stronger consistency level and thus require a cross-site synchronization. This paper proposes a novel consistency technique called RedBlue consistency is defined in which the blue operations are executed locally and lazily replicated in an eventual consistent manner. Red operations on the other hand are serialized with respect to each other and require cross site co-ordination. In such a unique consistency scheme, causality is maintained by ensuring that the dependencies established at the primary site when an operation is invoked is preserved when the operation is incorporated at some other site. As a developer, the first concern with this form of consistency is that one needs a clear cut strategy under which operations should be red or blue so that it is ensured that the application invariants are never violated and all replicas converge on the same final step. Since there are a few operations(Blue) which are lazily replicated, it might so happen that they are not always consistent. To address this, operations were basically divided into two components - a generator operation that identifies changes the original operation should make but has no side effects itself and the shadow operations that performs the identified changes and is replicated to all sites.

Properties of Geo-Replicated Systems

Before we go into a broad description of how consistency is maintained in a geo-replicated system, one must understand the properties that a geo-replicated system should have. They are:

- Low Latency - A geo-replicated system should always provide low latency responses. Providing a good user experience by providing low latency access implies that the operation should proceed after contacting a small number of replicas. But this is at odds with the other requirements that a geo-replicated system should satisfy.

- Causality - Causality is an important property that should be maintained both within a session and also across different replicas.
- All replicas should execute the same set of operations and converge to the same final state.
- All operations to all replicas should return the same value.
- All replicas should have the same stable history.
- In maintaining the above the geo-replicated system should not restrict any operation.
- All operations in any of the replicas should maintain the set of invariants. For e.g. No two users should accessing two different replicas should never have the same user id.
- Eventual Propagation - All operations at a single replica should be propagated to other.

Levels of Consistency

1. Strong Consistency - For strong consistency systems, replicated systems behaves like a single server that serializes all the operations. So the coordination among replicas lean to an overhead which is the delay in response.

2. Timeline/Snapshopt Consistency - Here there are two sites - a primary site that handles updates of all the replicas and a number of secondary states that acts as a read-only copy of the data. This kind of system although handles faster reads but writes always have an overhead.

3. Fork Consistency - This system relaxes the strong consistent system by allowing users to observe distinct causal histories. So the problem is once the replicas are forked, they cannot be reconciled.

4. Eventual Consistency - As the name suggests, all replicas diverge in short term as long as divergence is eventually repaired.

5. Multi Consistency - This kind of consistency eexposes multiple values exposed from divergent branches in operation.

6. RedBlue Consistency - RedBlue consistency level is one where the above issues are somewhat taken care of.

Other Work

1. Consistency Rationing - Consistency rationing allows consistency guarantees to be associated with data instead of operations and in it consistency levels automatically swith at runtime between weak consistency and serializability based on specific policies

2. TACT - TACT bounds the amount of inconsistency of data items in an application specific manner based on numeric errors,order errors and staleness of data.

RedBlue Consistency is better than these related work in the sense that it tries and handles consistency at the operation level and its main focus is not on adapting the consistency level of particular data items at runtime but on partitioning the operations according to their different actions.

RedBlue Consistency

Assumptions

Before defining what a RedBlue Consistent system is, it would be important to define a few assumptions that will in turn help us define the problem.

1. A distributed system with state fully replicated across k sites denoted $site_0 \ldots site_{k-1}$

2. S $\in S$ denotes a system state and $u, v \in O$ a set of operations.

3. Initial State - S_0. When operation u is applied it goes to state S'. So $S' = S + u$

4. $\forall$ S $\in S$, S$+u + v =$ S $+v + u$

5. A state S is valid if it satisfies all these invariants.

6. Each u is submitted to one site which is called $u's$ primary site and denoted by $site(u)$.

7. The system later replicates u to the other sites.

Defining RedBlue Consistency

The definition of RedBlue consistency has two components -

Definition 0.1. RedBlue order : Given a set of operations $U = B \cup R$,where $B \cap R = \emptyset$, a RedBlue order is a partial order $O = (U, \prec)$ with the restriction that $\forall u, v \in R$ such that $u \neq v, u \prec v$ or $v \prec u$ ($i.e.$ red operations are totally ordered).

Definition 0.2. Causal Serialization : Given a site i, $O_i = (U, <)$ is an i-causal serialization(or short, a causal serialization) of RedBlue order $O = (U, \prec)$ if

1. O_i is a linear extension of O ($i.e$, ¡ is a total order compatible with the partial order $\prec$)

2. for any two operations $u, v \in U$, if $site(v) = i$ and $u < v$ in O_i then $u \prec v$

Definition 0.3. RedBlue consistency : A replicated sytem is O-RedBlue consistent(or short, RedBlue consistent) if each site i applies operations according to an i-causal serialization of RedBlue order O

Formatting Citations

Citations can be handled in one of three ways. The most straightforward (albeit labor-intensive) would be to hardwire your citations into your LaTeX source, as you would if you were using an ordinary word processor. Thus, your code might look something like this:

```
However, this record of the solar nebula may have been
partly erased by the complex history of the meteorite
parent bodies, which includes collision-induced shock,
thermal metamorphism, and aqueous alteration
({\it 1, 2, 5--7\/}).
```

Compiled, the last two lines of the code above, of course, would give notecalls in *Science* style:

> . . . thermal metamorphism, and aqueous alteration (*1, 2, 5–7*).

Under the same logic, the author could set up his or her reference list as a simple enumeration,

```
{\bf References and Notes}

\begin{enumerate}
\item G. Gamow, {\it The Constitution of Atomic Nuclei
and Radioactivity\/} (Oxford Univ. Press, New York, 1931).
\item W. Heisenberg and W. Pauli, {\it Zeitschr.\ f.\
Physik\/} {\bf 56}, 1 (1929).
\end{enumerate}
```

yielding

References and Notes

1. G. Gamow, *The Constitution of Atomic Nuclei and Radioactivity* (Oxford Univ. Press, New York, 1931).
2. W. Heisenberg and W. Pauli, *Zeitschr. f. Physik* **56**, 1 (1929).

That's not a solution that's likely to appeal to everyone, however — especially not to users of BIBTeX (***?***). If you are a BIBTeX user, we suggest that you use the `Science.bst` bibliography style file and the `scicite.sty` package, both of which we are downloadable from our author help site (http://www.sciencemag.org/about/authors/prep/TeX_help/). You can also generate your reference lists by using the list environment {`thebibliography`} at the end of your source document; here again, you may find the `scicite.sty` file useful.

Whether you use BIBTeX or {`thebibliography`}, be very careful about how you set up your in-text reference calls and notecalls. In particular, observe the following requirements:

1. Please follow the style for references outlined at our author help site and embodied in recent issues of *Science*. Each citation number should refer to a single reference; please do not concatenate several references under a single number.

2. Please cite your references and notes in text *only* using the standard LaTeX `\cite` command, not another command driven by outside macros.

3. Please separate multiple citations within a single `\cite` command using commas only; there should be *no space* between reference keynames. That is, if you are citing two papers whose bibliography keys are `keyname1` and `keyname2`, the in-text cite should read `\cite{keyname1,keyname2}`, *not* `\cite{keyname1, keyname2}`.

Failure to follow these guidelines could lead to the omission of the references in an accepted paper when the source file is translated to Word via HTML.

Handling Math, Tables, and Figures

Following are a few things to keep in mind in coding equations, tables, and figures for submission to *Science*.

In-line math. The utility that we use for converting from LaTeX to HTML handles in-line math relatively well. It is best to avoid using built-up fractions in in-line equations, and going for the more boring "slash" presentation whenever possible — that is, for `$a/b$` (which comes out as a/b) rather than `$\frac{a}{b}$` (which compiles as $\frac{a}{b}$). Likewise, HTML isn't tooled to handle certain overaccented special characters in-line; for $\hat{\alpha}$ (coded `$\hat{\alpha}$`), for example, the HTML translation code will return [^(α)]. Don't drive yourself crazy — but if it's possible to avoid such constructs, please do so. Please do not code arrays or matrices as in-line math; display them instead. And please keep your coding as TeX-y as possible — avoid using specialized math macro packages like `amstex.sty`.

Displayed math. Our HTML converter sets up TeX displayed equations using nested HTML tables. That works well for an HTML presentation, but Word chokes when it comes across a nested table in an HTML file. We surmount that problem by simply cutting the displayed equations out of the HTML before it's imported into Word, and then replacing them in the Word document using either images or equations generated by a Word equation editor. Strictly speaking, this procedure doesn't bear on how you should prepare your manuscript — although, for reasons best consigned to a note (*?*), we'd prefer that you use native TeX commands within displayed-math environments, rather than LaTeX sub-environments.

Tables. The HTML converter that we use seems to handle reasonably well simple tables generated using the LaTeX `{tabular}` environment. For very complicated tables, you may want to consider generating them in a word processing program and including them as a separate file.

Figures. Figure callouts within the text should not be in the form of LaTeX references, but should simply be typed in — that is, `(Fig. 1)` rather than `\ref{fig1}`. For the figures themselves, treatment can differ depending on whether the manuscript is an initial submission or a final revision for acceptance and publication. For an initial submission and review copy, you can use the LaTeX `{figure}` environment and the `\includegraphics` command to include your PostScript figures at the end of the compiled PostScript file. For the final revision, however, the `{figure}` environment should *not* be used; instead, the figure captions themselves should be typed in as regular text at the end of the source file (an example is included here), and the figures should be uploaded separately according to the Art Department's instructions.

What to Send In

What you should send to *Science* will depend on the stage your manuscript is in:

- **Important:** If you're sending in the initial submission of your manuscript (that is, the copy for evaluation and peer review), please send in *only* a PostScript or PDF version of the compiled file (including figures). Please do not send in the TeX source, `.sty`, `.bbl`, or other associated files with your initial submission. (For more information, please see the instructions at our Web submission site, http://www.submit2science.org/ .)

- When the time comes for you to send in your revised final manuscript (i.e., after peer review), we require that you include all source files and generated files in your upload. Thus, if the name of your main source document is `ltxfile.tex`, you need to include:

  - `ltxfile.tex`.

- `ltxfile.aux`, the auxilliary file generated by the compilation.
- A PostScript file (compiled using `dvips` or some other driver) of the `.dvi` file generated from `ltxfile.tex`, or a PDF file distilled from that PostScript. You do not need to include the actual `.dvi` file in your upload.
- From BIBTEX users, your bibliography (`.bib`) file, *and* the generated file `ltxfile.bbl` created when you run BIBTEX.
- Any additional `.sty` and `.bst` files called by the source code (though, for reasons noted earlier, we *strongly* discourage the use of such files beyond those mentioned in this document).

1. We've included in the template file `scifile.tex` a new environment, {`scilastnote`}, that generates a numbered final citation without a corresponding signal in the text. This environment can be used to generate a final numbered reference containing acknowledgments, sources of funding, and the like, per *Science* style.

Fig. 1. Please do not use figure environments to set up your figures in the final (post-peer-review) draft, do not include graphics in your source code, and do not cite figures in the text using LaTeX `\ref` commands. Instead, simply refer to the figure numbers in the text per *Science* style, and include the list of captions at the end of the document, coded as ordinary paragraphs as shown in the `scifile.tex` template file. Your actual figure files should be submitted separately.