Optimizing the Update Packet Stream for Web Applications

Muthuprasanna Muthusrinivasan\textsuperscript{1} and Manimaran Govindarasu\textsuperscript{2}

\begin{itemize}
\item \textsuperscript{1} Google Inc., Mountain View CA 94043, USA \texttt{muthup@google.com}
\item \textsuperscript{2} Iowa State University, Ames IA 50011, USA \texttt{gmani@iastate.edu}
\end{itemize}

\textbf{Abstract.} The Internet has evolved to an extent where users now expect any-where any-time and any-form access to their personalized data and applications of choice. However providing a coherent (seamless) user experience across multiple devices has been relatively hard to achieve. While the \textit{how to sync} problem has been well studied in literature, the complementary \textit{when to sync} problem has remained relatively un-explored. While frequent updates providing higher user satisfaction/retention are naturally more desirable than sparse updates, the steadily escalating resource costs are a significant bottleneck. We thus propose extensions to the traditional periodic refresh model based on an adaptive \textit{smart sync approach} that enables variable rate updates closely modeling expected user behavior over time. An experimental evaluation on a size-able subset of users of the GMAIL web interface further indicates that the proposed refresh policy can achieve the best of both worlds - limited resource provisioning and minimal user-perceived delays.

\textbf{Keywords:} data synchronization, web applications, cloud computing.

\section{Introduction}

The World Wide Web has seen tremendous growth since its early days at CERN \cite{1}, and in the past few years has witnessed a steady shift away from the traditional desktop computing paradigm. The rapid emergence of cloud computing \cite{2} has given rise to - \textit{service providers} who build/manage universally accessible, massively scalable, highly reliable compute infrastructure as an utility/commodity, \textit{software vendors} who host their applications in this cloud avoiding capital expenditure and instead paying only for their resource utilization, and \textit{end users} who can now access technology-enabled applications easily without knowledge of the underlying infrastructure. While cloud/service providers \cite{3} benefit from multi-tenancy and economies of scale, the software vendors benefit from \textit{on-demand} access to resources for their SaaS \cite{4} deployments worldwide.

The greatest beneficiaries though have been the end users - the use of open standards, technological convergence \cite{5} and pervasive computing \cite{6} have enabled users to access information through a multitude of devices - instant
messaging through desktop clients, web browsers, mobile phones, interactive TV etc. is now possible. Users can thus access their data in the cloud using any device, at any time, anywhere in the world, and in any desired form, with no restrictions whatsoever. The widespread adoption of these ubiquitous applications is now primarily governed by natural expectations of a coherent user experience. In applications where data reconciliation can be quick and easy, seamless on-access device sync would suffice. However, it is untenable for applications requiring longer synchronization cycles or needing a complex update mechanism, due to undesirable user-perceived delays and slow application response times.

The traditional approach towards data reconciliation has been to execute mutually agreed update protocols at regular intervals. However, high frequency updates for devices having low user activity leads to a large number of redundant null updates. Such a non-optimal use of resources does not scale well for the cloud managing millions of users each possibly using tens of remote devices. Hence we propose a smart sync approach that exploits user behavior (past access patterns) to determine the likelihood of an impending user access to trigger a pro-active update. This would not only consume far fewer resources due to throttling of updates during periods of expected user inactivity, but also provide maximal data coherence across devices due to pro-active data synchronization. In this paper, we analyze a sizeable subset of GMAIL user interactions with the cloud, to determine optimal user behavior models for likelihood estimation (update interval adaptation) and to also study the resulting benefits for the users and the application in the cloud.

2 Related Work

Continuous harmonization of data over time across multiple remote sources, or data synchronization [7], has been a well studied topic [8], and has been explored in multiple contexts such as database design [9], secure communications [10], memory architectures [11], distributed computing [12], etc.

The research on data synchronization for ubiquitous computing has proceeded mainly along two lines, namely the how to sync and when to sync paradigms. The ‘how to sync’ issue deals with designing optimal data sync protocols that are scalable with network size and the resulting bandwidth/storage considerations. The traditional approach has relied on the use of timestamps [13] for version control, and delta compression [14] for minimal data transmission. Recent approaches relying on robust mathematical [15], information-theoretic [16], and probabilistic [17] techniques have provided greater flexibility.

Also, commercial solutions including Palm’s HotSync [18], Microsoft’s ActiveSync [19], Nokia’s IntelliSync [20], etc. are in wide use today. This proliferation of proprietary sync protocols and their mutual incompatibilities led to the