A Latency Hiding Framework for Enhanced Ubiquitous Access to Big Data in a Constrained Digital Ecosystem: Application to Digital Medical Archives.

Dessalegn Mequanint
Addis Ababa University
IT Doctoral Program
Ethiopia
Dessalegn.Mequanint@aau.edu.et

Lionel Brunie
Institut National des Sciences Appliquées(INSA)
France
Lionel.Brunie@insa-lyon.fr

Mulugeta Libsie
Addis Ababa University
Dept. of Computer Science
Ethiopia
Mulugeta.Libsie@aau.edu.et

David Coquil
University of Passau
German-French Doctoral College- MDPS
Germany
david.coquil@uni-passau.de

ABSTRACT
This paper presents our latency hiding framework for access to big data in a constrained digital ecosystem with application to digital medical archives. Aiming to enhance ubiquitous access of big data such as patient-oriented access of medical archives, we apply complex/multi-context prefetching to reduce latency thereby improving response time. We propose a formal model for prefetch requests rate and network workload or stress bound that takes into account a diverse set of constraints a digital ecosystem could be in. In addition to that, components of our latency hiding framework such as a generic multi-context functional architecture, use case model, medical database model with emphasis on API (abstracted patient information) and a high-level system architecture have been designed. The development of a complex or multi-context prefetch algorithm that uses a patient's chief complaints, slackness sensitivity, popular content tag, user specified contexts and constraints is underway. A prototype system will also be developed to validate the proposed solutions. Moreover, input and output metrics will be developed to gauge the efficiency and effectiveness of the prefetch algorithm under development.

Categories and Subject Descriptors
D.4.8[Performance]: Modeling and Prediction, Queuing theory

General Terms
Algorithms, Performance and Design.

Keywords
Latency Hiding, Multi-context Prefetching, Ubiquitous Access to Big Data, Digital Medical Archives.

1. INTRODUCTION
Moore's law of how computing power doubles every two years and Gilder's law of how communication power doubles every six months have been unable to fix the ongoing and perennial problems of asymmetric wired and wireless devices. As such the laws do not offer much benefit in constrained nodes, networks, and digital ecosystems. With high proliferation of smart and more constrained devices, the asymmetric nature of network bandwidth capacity, client devices' computing power, memory and display area size have continually become more pressing issues that need addressing. Thus, in order to improve quality of services as well as quality of users experience, enhanced techniques or approaches that take into account service bottlenecks in a constrained digital ecosystem that manifest because of low network bandwidth, CPU capacity, memory size, display area, and link unreliability are highly required.

In an attempt to soothe problems of latency, several techniques such as caching, prefetching and multithreading have been applied in various contexts almost ever since bottlenecks in computing have been noticed [10, 11, 12, 15]. The motivation in the past has been mainly the lack of symmetry in devices computational capacity or network bandwidth, and in this digital age and era the asymmetry issue has taken new dimensions on top of what already exists. Latency manifests in various forms mainly as a result of the asymmetric nature of the growth of data communication elements capacity [1, 2, 3, 4, 5, 6, 7, 9, 13]. With emerging trends such as cloud computing and virtualization taking strong foothold, the need to hide latency has become even more pressing. In the cloud, on top of latencies that can be caused by inadequacy of network bandwidth and other constraints, virtual machines can also introduce additional latency because of the time-sharing nature of the underlying hardware.

Thus, latency is a preeminent issue that cuts across computing paradigms and needs addressing. The healthcare system is one of the most information intensive sectors. A single patient visit could result in the creation of massive amount of data in the form of imageries-X-rays, CTs, MRIs, reports, laboratory test results, etc. Healthcare providers shall have access to historical data such as reports, imaging findings and laboratory results in an integrated manner in the course of treating current ailments and to be able to facilitate comparison of a patient's current status with findings from past examinations towards accurately evaluating the temporal evolution of a medical condition. Ideally, all the patient data need to be quickly available to healthcare practitioners involved at the point of care. However, this in most cases is not possible due to constraints on network bandwidth and processing and storage capacity of intermediary nodes, and most importantly data origin servers could be remotely placed. Traditionally prefetching of medical records has been restricted to restoring of archived images from picture archiving and communication systems to diagnostic imaging workstations [16]. This work takes a different approach from the problem-oriented trend in that, among other things, it uses multi-context for a patient-oriented prefetching.

In this work, pertinent issues that surround latency hiding in a constrained digital ecosystem with an emphasis on e-health ecosystem where healthcare providers such as primary, secondary and tertiary levels are interconnected to a data center towards the provision of integrated care in the prevalence of network bandwidth and devices capacity constraints will be investigated.

The rest of this paper is organized as follows. Section two summarizes the state-of-the-art. Section three outlays the motivation that underpins the need for latency hiding for

ubiquitous data access in a constrained digital ecosystem. It further provides a brief account of the problem context and the research questions along with the research objectives and functional goals. Section four presents the various complementary models of our latency hiding framework. Finally, section five gives a summary of what has so far been done and outlines the immediate priorities of the research in progress.

2. Related Work

In this section prominent techniques on latency hiding in regards to the use of prefetching are reviewed. An attempt has been made to put into perspective as to where the status of the state-of-the-art lies with respect to how latency related problems are addressed. In doing so, the scientific contributions of this research have been highlighted.

Most of the prefetching techniques proposed in the literature can be characterized as either aggressive [10, 11, 12] in that they are meant only for the prominently adage of overlapping communication with computation, or pattern based [1, 2, 3, 4, 5, 6, 7] in that the prefetching is carried out based on a user's past access patterns of web pages. These approaches cannot be adopted or improvised for prefetching of digital medical records, as the notion of prefetching based on access patterns makes no sense. Moreover, prefetching aggressively in a constrained digital ecosystem is also impossible.

Prefetching as a latency hiding technique is not a new idea. It has been and still is researched in several settings, particularly to increase parallelism of tasks so as to keep the CPU busy by fetching the data before it is required. Prefetching again has come into the fore because of several factors. Chief among them are repositories are getting disproportionately huge more than ever, and exciting and at the same time challenging requirements are continuously popping up [13, 14]. Most importantly, the unparalleled growth of data communication elements capacities is showing no sign of change. Moreover, emerging trends such as putting data and services into the cloud means remote access to data and services will suffer from latency and hence may affect users' quality of experience. Knowing how cloud computing is young and evolving, how quickly latency has become an issue may not surprise many as it is being said that cloud's computing value proposition is generally not one of improved application performance, but flexibility and cost savings. In [8], a real-time data prefetching algorithm based on sequential pattern mining in a cloud setting has been proposed.

Prefetching has been used mainly in three flavors, such as in the form of web prefetching for hiding web access latency, in the form of data prefetching for hiding data access latency and a non-data prefetching which is also known as DNS prefetching for hiding latency that manifests in connection setup. Prefetching based on estimated network bandwidth and a miss strategy to decide what to request from data sources in case of cache miss based on real time network conditions has been extensively researched. However, the prefetching methods proposed in the literature in recent years have been concentrated on the semantic web [1, 2, 3].

Different methods have been proposed to mine user's sequence of page views and subsequently predict what next page the user is likely to view so that it can be prefetched. One of the mechanisms has been to use association rules as a result of sequential pattern mining [4]. The second and predominantly used mechanism to mine user's sequence of page views has been modeling user's accessed web pages as a Markov process with states representing the accessed web pages and edges representing transition probabilities between states computed from the given user sequence in the web log. There have been also recent efforts to integrate semantic information in the mining process in order to reduce the state complexity while maintaining high accuracy of the Markov model [1, 2, 3, 6, 7]. Research efforts to streamline CPU usage and response time via prefetching that result in overlapping of data communication with computation have brought in significant achievements [10, 11, 12]. However, their aggressiveness restricts their usefulness in constrained settings.

Furthermore, a semantic-based effort using a next page prediction method based on domain ontology for semantic web usage mining is proposed in [3]. However, its application is limited only to web access. A prefetching technique based on the notion of complex or multi context and fine-grained data access other than pages in the web and most importantly for access to digital medical archives is needed in constrained settings where network bandwidth and client devices capacities are in short supply. Moreover, the prefetching technique proposed for access to digital medical records has shortcomings in that it is only meant for problem-oriented access as opposed to patient-oriented, and can only be used in local area network settings [16]. This is where the envisaged research comes in; the use of complex or multi context prefetching with coherent caching strategy to hide latency for patient-oriented access to digital medical archives in constrained digital ecosystem settings which typically has a request and reply model built in the shape of client/server or similar models.

In summary, in our research additional requirements such as discriminating prefetch requests as slacked (not urgent) and non-slacked (urgent), and the use of multi-context in general are considered for preetching digital medical records means, a clean state approach towards developing a suitable multi-context prefetching technique is desirable.

3. Motivation, Problem Context, Research Questions and Functional Goals

3.1 Motivation

The size of data being generated every minute is getting bigger and bigger ever. In particular the Internet has become a place where massive amount of information and data are being generated every day. However, ubiquitous access to these data being generated is constrained in many ways due to factors mentioned in the previous sections. The following excerpt gives a snapshot of the amount of data generated on the Internet every minute [14]. YouTube users upload 48 hours of video, Facebook users share 684,478 pieces of content, Instagram users share 3,600 new photos, and Tumbir sees 27,778 new posts published. There are also other sites which people use on a regular basis and will likely continue to use in the future which are known to generate big data every minute. Emerging requirements which seek to harness today's digital data and real-time analytics for global development [14] also give more weight to the urgency and at the same time the critical need of the development of latency hiding techniques. The healthcare sector is also one of those known to generate massive amount of data. Table 1 is a summary of what a typical patient medical record encapsulates.

Table 1: Descriptions of Medical Data Items.

Data Item Descriptions	Estimated Size
Patient Demography	200-300 Bytes
Prior studies (Imageries-X-rays, CTs, MRIs, textual descriptions, etc.)	24-200 MB per image
Medicine and allergy lists, and immunization status	200-300 Bytes
Laboratory test results, vital signs	200-500 Bytes
Medication information including side-effects and interactions	200-300 Bytes
Evidence-based recommendations for specific medical conditions	1KB-1 MB
A record of appointments, reminders, physician and nursing notes	500-1K Bytes
Total estimated size of a single patient record could get	**>200 MB**

Transmitting or accessing data of several patient medical records in slow networks with bandwidth capacity of 512Kbps or fewer megabytes, which is not uncommon in developing countries, takes too much time which has been the most critical challenge that impedes access to data and services. The review patient history use case where prior studies of a patient need to be transported to the point of care ahead of a patient's scheduled visit is one which highlights the need for a tailored latency hiding mechanism.

3.2 Problem Context

In many cases a patient's medical records exhibit temporal relationships and as a result prior studies are critically required for current illness examinations. To this effect, relevant patient's medical history needs to be available for review at the point of care with sufficient lead time which is challenging to achieve in a constrained environment such as in networks with restricted bandwidth capacity. Thus, a better scheme or protocol tailored to the constraints that a given e-health ecosystem is under is desirable.

3.2.1 Use case Model

The two use cases depicted in Figure 1, i.e., the patient history review and the perform pharmacology preparation use cases, are the primary targets of the research in order to demonstrate the workability and significance of the latency hiding technique under development. A suitable use case for demonstrating the significance of delaying content transcoding for enhancing cache hit rates is under development.

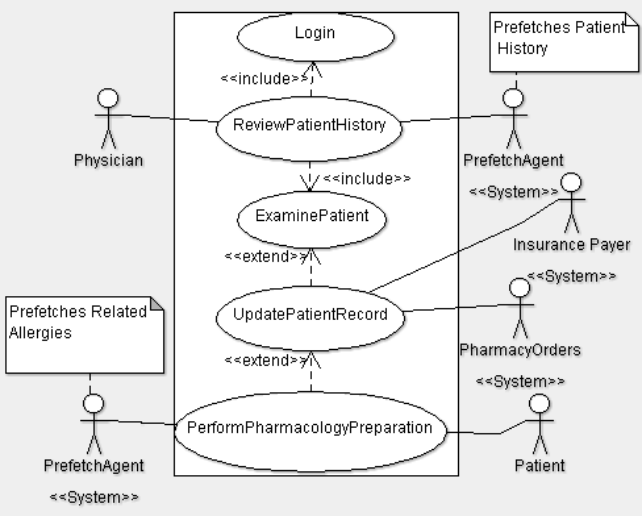

Figure 1. A Use Case Model for Multi-Context Based Digital Medical Records Prefetching.

3.3 Research Questions and Functional Goals

Hypothesis: The use of complex/multi-context for guiding prefetching decisions for ubiquitous access to data in a constrained e-health ecosystem is possible. It would improve effectiveness of pre-fetching and response time.

Complex/Multi-context represent contexts of generic as well as domain specific nature. The term encapsulates constraints of data communication elements such as network bandwidth threshold, devices display area, CPU capacity, memory size, and domain context (for instance in the medical domain, a patient's chief complaints), request context such as slacked or non-slakced in addition to other user specified contexts or constraints.

In line with the problem context described above, this research is therefore set out to realize the hypothesis by answering the following questions:

- How best the full range of constraints that impact prefetching decisions in digital medical archives can be modeled?
- To what extent the use of complex/multi-context based predictions improve cache hit rates?
- To what extent dynamic complex/multi-context prefetching strategies impact cache utilization?
- To what extent data staging of popular contents to the cache server for small and mobile client devices maximizes cache hit rates?

Towards enabling ubiquitous patient-oriented access to digital medical archives in a constrained e-health ecosystem, the following two functional goals are sought after:

- **Improve response time or hide latency**: improving users quality of experience through multi-context prefetching that brings patient medical records which are large in size to the site where they are needed in order for a physician to undergo a review with sufficient lead time before the scheduled visit of a patient,

- **Improve resilience**: the caching server(s) that need(s) to be placed close to nodes or client devices is/are capacitated to ease temporary glitches on the remote data origin server to some extent.

3.4 Research Objectives

The general objective of the research is to design a latency hiding framework to take explicit account of asymmetries among data communication elements in order to achieve the twin goals of complex/multi-context based prefetching- reducing the peaks of bandwidth demand or hiding latency and providing fault tolerance to some extent.

Derived from the general objective, the specific objectives to achieve are the following:

- To enhance users quality of experience in accessing patient medical records ubiquitously in a constrained e-health ecosystem;
- To model the arrival of prefetch requests, the workload and multi- contextness in connection to prefetching of patient medical records;
- To develop simple/composite service metrics for gauging the accuracy level of the envisaged complex/multi-context algorithm; and
- To showcase the benefits and the soundness of the proposed solutions through a prototype system.

4. Proposed Preliminary Models

In this section preliminary models along with ongoing modeling activities are presented. At this stage of the research, models such as functional model/architecture, patient or medical database model with particular emphasis on patient medical histories, prefetch requests model and network workload bound model have been designed.

4.1 Functional Model/Architecture

This section presents the possible general architecture that can effectively support the functional goals of the complex/multi-context prefetch. The various elements that constitute the prefetching process are represented in the form of layers as depicted in Figure 2, with each layer responsible for a designated role as described below. A suitable mapping of the layers or components of the architecture into client or server nodes that takes in to account performance related issues and the functional goals of the architecture will be decided during the design and development of the prototype system.

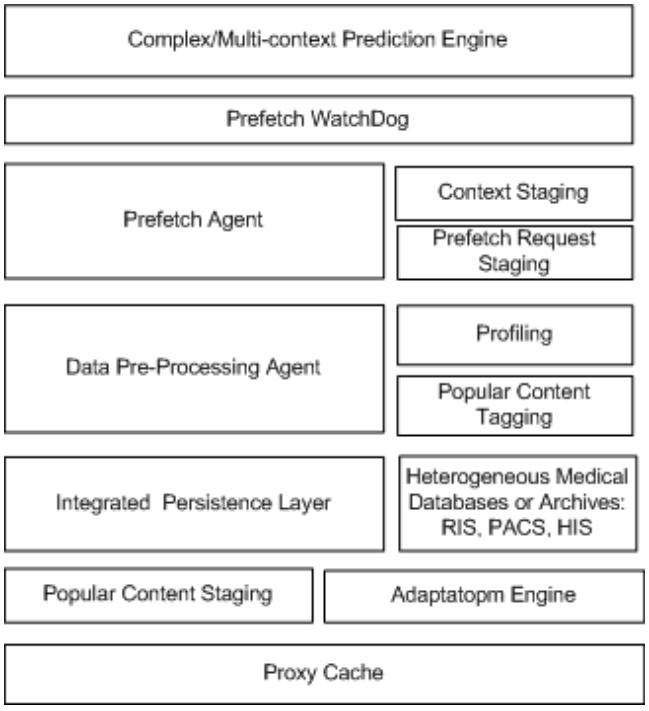

Figure 2. A Functional Model or Architecture for Prefetching.

Complex/Multi-context Prediction Engine- multi-context as has been defined in section 3, represents context of generic and domain specific nature. This layer is responsible for making prediction with regard to what content to prefetch using the complex/multi-context as hint. It also stages context to the next layer. Slackness sensitivity is low means the prefetch request is non-preemptive and needs to be served immediately with the resources available, and it is high means the prefetch request is preemptive and prefetching can take place later when the minimum required resources are available.

Prefetch WatchDog: this layer is responsible for monitoring bandwidth and cache consistency. It monitors the fulfillment of the necessary resources to trigger prefetching. In addition, it performs traffic control by adjusting the ongoing prefetching to the capacity of the network bandwidth available. Throughout the prefetching, the rate can fluctuate as a result of an increase or decrease of the available network bandwidth. Thus, the Prefetch WatchDog has to take care of the fluctuation in bandwidth. By doing so it tries to reduce the idle time of the network to the minimum possible thereby making the traffic rate constant. Computing the optimal data size to prefetch in the subsequent proxy server's idle time is also done by this layer. Moreover, it also decides when to prefetch.

Prefetch Agent: this layer is responsible for staging both context and prefetch requests. The cache proxy intercepts client device prefetch requests and stages the request along with context information to the data origin server via the Prefetch Agent. It triggers the Data Pre-Processing Agent to start prefetching.

Data Pre-processing Agent: this layer is responsible for profiling in addition to filtering tagged popular contents. A popular content is one which has the potential to be accessed by several devices of different capacity. This layer does not perform content adaptation. However, during a prefetch request of a popular content,

adaptation could be delayed in order to maximize hit rates. To this effect, staging and placing the prefetched popular content at the proxy cache is done so that the popular content can be used to serve future prefetch requests of diverse client devices. In prefetching medical archives, the Data Pre-Processing Agent layer performs pre-processing activities such as patient identification and the subsequent patient history or prior studies mapping before a prefetch begins.

Integrated Persistence Layer: this layer is responsible for providing access to persistent historical data of heterogeneous nature. For instance, in prefetching medical archives, it provides access to HIS (Health Information System), RIS (Radiology Information System), and PACS (Pictures Archive and Communication System) which are critically needed to facilitate comparison of a patient's current status or illness with findings from past examinations.

Popular Content Staging and Adaptation Engines: the proxy cache stores roughly two forms of content. For instance, in digital medical archives, the first one could be patient-oriented data required by healthcare professionals for review, and the second could be individual-specific data prefetched transcoded, or prefetched and staged (or pulled down) to the proxy cache from the data origin server without being transcoded. The Adaptation Engine at the proxy cache is required for adapting staged or pulled down popular contents to the capacity of constrained devices. This improves cache hit rates in addition to keeping long distance communication to the minimum possible.

4.2 Data Model for a Patient-Oriented Access Medical Database

A data model for a patient-oriented medical database with emphasis on entities or objects related to visits that a patient makes in the course of his/her illness history is depicted in Figure 3. The emphasis is on entities/objects that are traditionally known by healthcare professionals as SOAP (Subjective Objective Action and Plan). Ideally the entire patient history needs to be prefetched for review. However, due to the prevailing constraints explained in previous sections, only selected ones need to be prefetched, and therefore those selected have to be prefetched with maximum accuracy. Thus, only a summary (most relevant) of patient history and prior studies are needed for current diagnosis or treatment and hence the acronym API (Abstracted Patient Information) is used. The efficacy of the proposed latency hiding technique is gauged with respect to the degree of accuracy of prefetching the relevant APIs related to a patient's current illness for which s/he is scheduled to visit a physician or healthcare professional. The data model is subjected to further evolvement as the prototype development gets underway.

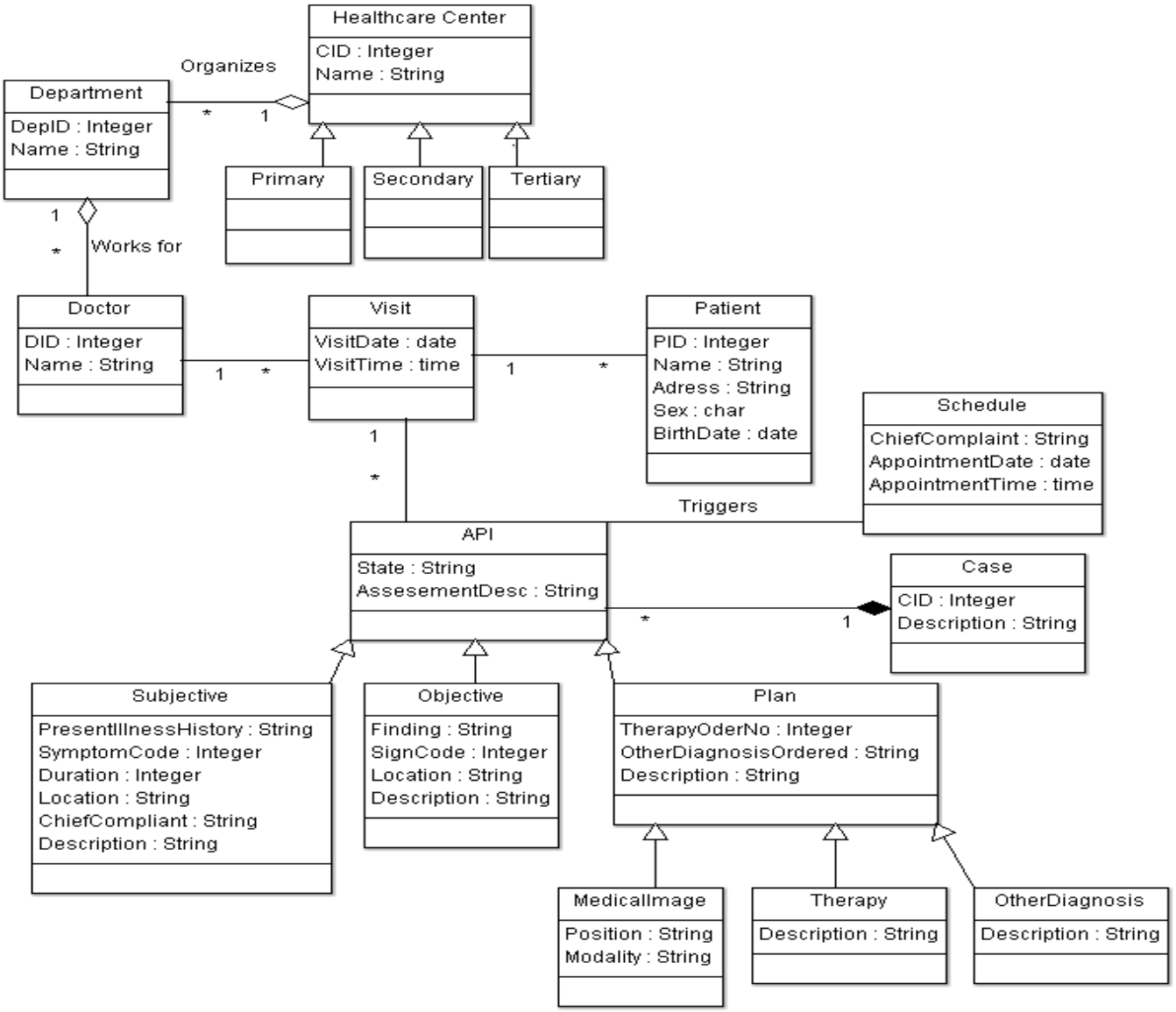

Figure 3. A Data Model for a Medical Database.

4.3 Request Arrivals Model

The effectiveness of the proposed latency hiding technique hinges on how best the prefetch activities are controlled and coordinated. Scheduling generated prefetching requests with respect to dynamically changing server resources and network workloads needs control and request type categorization such as by urgency or slackness. To this effect, a queuing model to characterize the nature of prefetch transactions entertained by the data origin server where medical records are archived is needed. The goal in using a queuing model is in order to be able analyze server resources as well as bandwidth or service channel capacity utilization and the average response time for prefetch requests differing in both arrival rate and type(slackness sensitivity). This would enable us the design of a prefetching scheme both adaptable and responsive to the level of slackness and aggressiveness expressed by the prefetch requests.

The following are the essential features for developing a mathematical queuing model for the queuing system under consideration:

- **Arrival Process:** the probability distribution of prefetch requests arrival in the envisaged prefetch system;
- **Service Process**: the probability distribution of prefetch service times in the envisaged prefetch system;
- **Number of Servers:** two servers are working in series, client prefetch requests are directed to the server which serves as a proxy cache, which at the same time serves as an agent to relay prefetch requests to the data origin server. The second server, the data origin server, receives prefetch requests and processes the requests for providing best or accurate responses.

Prefetch requests flow in steady state and are generated randomly. Random arrival is used because we assumed that the arrival of prefetch requests at a given point in time is completely

independent of the number of prefetch requests arrived during any previous interval. The number of prefetch requests arrival (number of prefetches) is a discrete variable and hence perfectly fit in the queuing model with Poisson distribution. At this stage of the research, a preliminary model is developed in order to approximate the expected number of prefetch requests within an interval of a given time. It is subjected to further evolvement or fine-tuning as the research progresses further. The following equation can be used to calculate the expected or mean (μ) prefetch requests arrival rate:

$$\mu = \lambda_i P(T) \qquad (4.1)$$

where λ_i is the prefetch request arrival rate in an interval of time, and P(T) is the probability that there will be an arrival of prefetch requests of μ at the data origin server in the time interval 0 to t, which is given by:

$$P(T)= 1- e^{-\lambda_i t} \qquad (4.2)$$

In the setting under consideration, the prefetch requests of clients will be serviced by a shared link capacity of **LC** connecting the data origin server and the proxy or cache server. The capacity of the link is shared among clients whose prefetch requests are serviced quasi-simultaneously. Each prefetch request gets a fraction of the entire link capacity, thus for an attainable peak rate **PR** of a connection, the arrival rate of prefetch requests can be approximated as:

$$\lambda_i = LC/PR \qquad (4.3)$$

The entire link capacity is segmented into a number of separate channels. However, for small number of prefetch requests, each request could be served with relatively larger channel capacity or higher peak rate **PR,** and for large number of requests, the prefetch requests are bound to get a fraction of the entire link capacity **LC** much less than the peak rate.

The possibility of modeling the arrival rate using Gaussian distribution for large number of prefetch requests is also under consideration. The queue discipline as to how prefetch requests need to be serviced given they could be preemptive and non-preemptive on top of commonly used queue disciplines such as FIFO, LIFO and Priority is also under development.

4.4 Network Workload Bound Model

Prefetching can be an overhead if cache misses soars which is a result of low accuracy. In general estimating workload gives a flavor of the accuracy level of the prefetching and the soundness of the proposed algorithms or latency hiding scheme. On the basis of the proposed functional architecture presented in section 4.1, the extent of the stress or workload the network is going through can be gauged. This would be pivotal to address performance related issues.

4.4.1 Workload Description

The data origin server handles many online and offline (i.e., responses can be delayed or may not be instantly delivered) patient data prefetching requests that come from several clients. Based on the settings of the system architecture depicted in Figure 4, the nature of traffic flows to and from the proxy/cache(s) and the data origin server(s) are identified for modeling the work load or network stress under the slacked (offline) mode of operation. Prefetch requests originating from thick or thin client devices are intercepted by the Proxy Cache to stage or forward them to the data origin server. Once prefetching takes place, the prefetched data are stored at the Proxy Cache and subsequently delivered to the data sinks (thick or thin client devices) initiating the request(s).

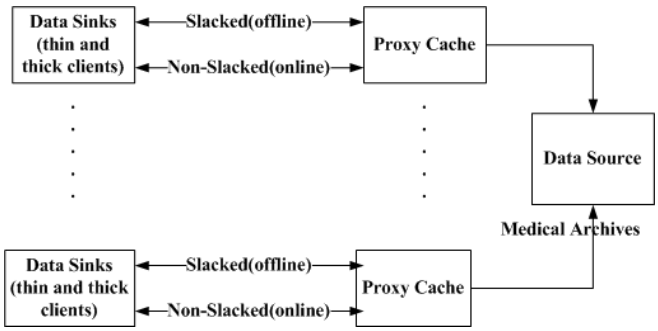

Figure 4. A High-Level System Architecture for the Settings of the Complex/Multi-context Prefetching

A mapping can be associated with the functional architecture proposed in section 4.1 and the system architecture of Figure 4 in a way that the Complex/Multi-context Prediction Engine including the Prefetch Watchdog and the Adaptation Engine can be deployed in the proxy cache, and the rest such as the Prefetch Agent, Data Pre-Processing Agent, the Integrated Persistence Layer and the Adaptation Engine can be deployed in the data origin server(s). The Adaptation Engine needs to be deployed at both ends. The Adaptation Engine at the proxy cache performs content adaptation or transcoding of popular contents.

4.4.2 Network Workload/Stress Modeling

The following is a preliminary workload model developed to gauge the amount of network workload or stress generated for the review patient history use case. It is subjected to further evolvements. The network workload bound NWB_n for prefetching n records or tuples for slacked (for prefetching that can take place latter when the required minimum resources are available) mode of operation can be modeled as follows:

$$NWB_n = \begin{cases} 0 & if\ K = N \\ \frac{LC}{PR}(N-K)S & if\ 0 < K < N \\ \frac{LC}{PR}(NS) & if\ K = 0 \end{cases} \qquad (4.4)$$

where the ratio **LC/PR** gives the virtual channels generated out of dividing the entire capacity of the link into virtual separate channels by the peak rates that constitute the prefetch request flow rates, K stands for the number of unwanted or unrelated patient's medical records for review, N stands for the total number of a patient's medical records and S stands for the average size of a medical record or data.

The goal is to increase the accuracy of prefetching by filtering abstracted patient's medical records that are only important for review. Thus, K=N means there are no related patient's medical records, and K=0 means all the patient's medical records are needed for review. The development of a similar workload model for the non-slacked (online) traffic is underway.

5. Conclusion, Future Works and Challenges

Latency is a preeminent issue that cuts across computing paradigms, and hence needs addressing. In this paper the notion of complex or multi-context prefetching has been introduced. Artifacts such as use case diagram, functional architecture, prefetch requests rate model, a medical data model and a high level system architecture have been developed. Several pertinent issues in relation to ubiquitous access to big data, particularly digital medical archives in a constrained digital ecosystem are also under investigation. The immediate priority of the research is to enhance or evolve the proposed models into the next level. Subsequently, activities such as the development or adoption of prefetching algorithm for the use cases under consideration, the development of input and output metrics, modeling and formalization of complex or multi-contextness (which includes domain specific and general contexts that can serve as hints for prefetching), service time modeling and proof of concept development in the form of a prototype system will be carried out.

REFERENCES

[1] Mabroukeh, N.R. and Ezeife, C.I.;, "Semantic-Rich Markov Models for Web Prefetching", *Data Mining Workshop, 2009. ICDMW '09. IEEE International Conference on*,, pp. 465-470, 6 Dec. 2009.

[2] C.-Z. Xu and T.I. Ibrahim: "Semantics-Based Personalized Prefetching to Improve Web Performance," Proceedings of the IEEE 20th International Conference on Distributed Computing Systems, Taiwan, April 2000, pp. 636-643.

[3] Nizar R. Mabroukeh and Christie I. Ezeife: "Using domain ontology for semantic web usage mining and next page prediction," Proceedings of the 18th ACM conference on Information and Knowledge Management, November 02-06, 2009, Hong Kong, China.

[4] Nizar R. Mabroukeh and C. I. Ezeife: "A taxonomy of sequential pattern mining algorithms", ACM Computing Surveys (CSUR), Vol. 43 No. 1, pp. 1-41, November 2010.

[5] J. Pitkow and P. Pirolli: "Mining longest repeating subsequences to predict web surfing", In Proceedings of the 2nd USENIX Symposium on Internet Technologies and Systems 2, pp. 13–21, October 1999.

[6] Xin Jin and Huanqing Xu. "An Approach to Intelligent Web Pre-fetching Based on Hidden Markov Model", Proceedings of the IEEE Conference on Decision and Control, 2003, USA.

[7] C.Z Xu and T. I. Ibrahim: "A Keyword-Based Semantic Prefetching", IEEE Transactions On Knowledge and Data Engineering, Vol. 16, No. 4, April 2004.

[8] J. Li and S. Wu: "Real-time Data Prefetching Algorithm Based on Sequential Pattern Mining in Cloud Environment", American Journal of Engineering and Technology Research, Vol. 11, No. 9, September 2011, China.

[9] Peitso, L.: "Communications latency hiding for distributed SoS", *System of Systems Engineering (SoSE), 2011 6th* International Conference, pp. 113-118, 27-30 June 2011.

[10] Ramos, L.M., Briz, J.L., Ibáñez, P.E, and Viñals. *"Multi-level Adaptive Prefetching based on Performance Gradient Tracking"*, The Journal of Instruction-Level Parallelism, January 2011. Vol. 13, pp. 1-14.

[11] U. Yoon, H. Kim, and J. Chang: "Intelligent Data Prefetching for Hybrid Flash-Disk Storage Using Sequential Pattern Mining Technique", IEEE/ACIS 9th International Conference on Computer and Information Science, pp. 280-285, 2010.

[12] Yong Chen, Huaiyu Zhu, and Xian-He Sun. "An Adaptive Data Prefetcher for High-Performance Processors," In proceedings of the 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing, pp. 155-164, 2010.

[13] Visual News, "How much Data are Created Every Minute", DOI=http://www.visualnews.com/2012/06/19/how-much-data-created-every-minute/.

[14] UN Global Pulse, "Big Data for Development: Challenges and Opportunities", White Paper, http://www.unglobalpulse.org/sites/default/files/BigDataforDevelopment-UNGlobalPulseJune2012.pdf.

[15] Boothe, B. and Ranade, A.: "Improved Multithreading Techniques for Hiding Communication Latency in Multiprocessors", Proceedings of,the 19th Annual International Symposium on Computer Archityecture, pp. 214-223, 1992.

[16] Alex A., Michael F., Jonathan G., Alfonso F. and Denise R., "Problem-oriented Prefetching for an Integrated Clinical Imaging Workstation," J Am Med Inform Assoc. 2001; 8:242–253.