The Moving Cloud: Predictive Placement in the Wild

Azarias Reda and Brian Noble
{azarias,bnoble}@umich.edu
University of Michigan

1 Introduction

Latency presents an enduring—and worsening—challenge to mobile systems designers. With the increasing adoption of cellular devices as the primary avenue of network connectivity for many users around the world, the "reach" of latency as a first-class concern is extending. Bandwith grows roughly as the square of latency, while storage capacity is growing faster still [16].

There are several reasons for this. The instantaneous nature of latency makes it hard to improve the metric by simply packing more bits on the wire. To make things worse, additional devices on the network, such as firewalls and switches, add more delay to network packets while minimally impacting the aggregate bandwidth. Finally (and perhaps most importantly), bandwidth is simply easier to sell in the marketplace.

Unfortunately, even in systems with an adequate balance between latency and bandwidth—and such systems will become increasingly rare—humans are acutely sensitive to delay and jitter. Performance analyses of interactive applications, ranging from video streaming to interactive web services, show a modest increase in latency can make a session noticeably annoying or unusable [14]. Even at a few hundred milliseconds of latency, all too common in cellular networks that often power developing region connectivity, users reported many applications start to become noticeably annoying [23]. For highly interactive applications, user experience degrades significantly much sooner.

The latency problem is even more pronounced in challenged network environments endemic to developing regions, where resources are limited to begin with. With cellular links and shared dial up connections in internet kiosks as the typical ways to connect to the internet [17], developing countries face significant challenges in network access. This makes even simple network tasks unpleasant, and rich media prohibitively difficult. Working through an interactive session in one of these kiosks can be charitably described as frustrating [19]. Provisioning data and computational support as close to demand as possible is the key to solving this problem [20].

To address this provisioning problem, we propose *the moving cloud*, a framework for proactive data delivery. The moving cloud leverages route fingerprints in individual mobility and users' contextualized behavior of data access for predictive data placement. Essentially, we are trading bandwidth and storage for latency, exchanging resources that grow more quickly for the one that grows most slowly. The moving cloud alleviates the latency problem by proactively placing content where it needs to be in the near future, so that resources are closely and readily available when requested by the user. This paradigm enables a number of networking scenarios including bulk data access, mobile resource augmentation, on-demand social networks and personal content distribution.

People are creatures of habit, and move in repeated patterns that can be probabilistically learned. As such, several models have been suggested for predicting human mobility [8, 21, 22]. Given the history of locations visited, these models typically predict the next location for a user. Our framework employs a new approach for augmenting these predictions with time bounds, producing actionable information for data placement. This temporal component of mobility is crucial as data delivery often has some freshness constraint. Our approach can be combined with most existing location predictors, and enhance them with an expected-time dimension. For example, coupled with a second order Markov model [22], our system was able to predict time of arrival within an hour in more than 90% of the hits in a sample dataset.

Another important observation is the contextual nature of data use. Not only do people move in repeated patterns, they also access data in a habitual manner. Data accessed in one context, say during school hours, is often different from data accessed in another, say while at an internet kiosk. As a result, individual mobility can also provide an informed data selection policy in content placement. We combine these complementary observations in building a secure, generic, proactive and predictive data delivery framework.

1.1 Case Study: Cyber Foraging

Cyber foraging [4, 7] is a decade old idea of augmenting the computational and storage capacities of mobile devices, not with remote machines located in the cloud, but with locally available surrogates. These surrogates can be commodity computers provided for use in enterprises or public locations. Cyber foraging enables mobile devices to perform computationally intensive activities, while circumventing the latency problem by limiting their communication to a local computer—often a single hop away.

The latest in this line of research is the idea of cloudlets—'datacenters in a box' [20]. Cloudlets could be deployed alongside wireless access points, and provide on demand augmentation to mobile devices in the vicinity. Like most cyber foraging solutions, cloudlets also use virtual machines for encapsulating and moving applications between surrogates. While virtual machines provide an elegant solution to problems like process migration and configuration in transporting services, their big size is often problematic. As a result, starting up a new surrogate with the user's stateful virtual machine downloaded on demand incurs a significant delay.

Some solutions have been suggested to this problem, including splitting virtual machines to a 'base' component that is common among many users, and an 'overlay' for transient customization of the base. Since chances of finding an exact base VM at all surrogates is slim, users are expected to maintain multiple overlay VMs. This introduces a number of problems when it comes to updating components (due to the dependency cycle among many people) and synchronizing state between multiple overlays (due to overlay incompatibility).

The moving cloud provides a natural solution to this problem by leveraging repeated patterns in human mobility, as well as contextual data access, for predictively placing dynamic content in the wild. Our framework places personalized cloudlet virtual machines in a user's near-future vicinity, allowing their use with minimal initialization costs. This in turn enables users to almost entirely bypass the latency problem without going through the messy detail of assembling and disassembling pieces. The moving cloud approach becomes even more feasible as the available bandwidth improves, allowing users to ride on the winning side of the network innovation wave.

Our key contributions in this work are:

- A pluggable framework for predictive data placement
- An algorithm for time prediction in mobility models
- And a method for utilizing mobility for contextual data selection

2 Design

With strict constraints on energy efficiency, size and usability, mobile devices are at a perpetual disadvantage to their stationary counterparts in computational power. Nonetheless, consumers still want to run intensive and demanding applications on these devices [5, 18]. Luckily, the network can be leveraged to bridge this gap, enabling mobile devices to compute in harmony with more capable machines in the cloud. Yet, this is also difficult in many scenarios where mobile network connectivity is limited—which is especially true in developing regions. Therefore, it would be preferable for mobile devices to leverage computing resources available on machines in their vicinity where the access latency is low. To accomplish this, however, devices need to quickly initiate and customize external resources to their preferences. Predictive placement enables this scenario, realizing the notion of a readily available cloud that moves as the user moves—keeping the needed resources close, and the latency for reaching them low.

Sensor-rich mobile devices make it easy to garner detailed location and contextual information as the user moves in the environment. As this information accumulates over time, it often defines a pattern of mobility and data access behavior that can be probabilistically learned. Combined with a framework for securely delivering data, this information can enable many new networking opportunities, significantly improving how mobile devices interact and take advantage of their surrounding.

The key principles in designing this framework were:

- Proactive placement: utilizing individual mobility history to build a model for placing data where it needs to be in the near future
- Default security provisions: supporting secure data distribution out of the box, and allowing developers to modify it as needed
- Contextual feedback: notifying applications about contextual patterns in data access for fine tuning future requests
- Simple customization: providing a set of programming interfaces for taking advantage of the framework in developing new services.

The moving cloud is concerned with placing content in the user's future vicinity for enabling crisp user experience by reducing interaction latency. It provides mechanisms for location based node discovery and secure data delivery, allowing applications to focus on their business logic. A node is a publicly available computational resource that mobile devices have access to. Such augmentation nodes could be standalone machines available in the wild such as at schools or libraries, or deployed alongside access points, as discussed in several projects [7, 20]. The moving cloud has three main pieces that deal with modeling mobility, establishing access context and securely delivering data. Figure 1 shows the common usage scenario for the moving cloud.

A typical application using our framework will have two components: a server based component where high fidelity, first level replica of its data lives and a mobile component that interacts with augmentation nodes and executes its user-facing operations. An application simply hands data from its server component to our framework, which selects and places the data predictively at a destination node in the user's future vicinity. This is accomplished by recording the user's mobility history over time, and building a model for data placement. As updates to the data happen in the wild, the changes are propagated back to the application's server component, which merges them and prepares future placement requests. The user's contextual access patterns that get established over time are also provided back to the application, which can inform these future requests. The framework supports simple interfaces for pushing data into the service, providing contextual access feedback, harvesting state from nodes, as well as versioning multiple transfers.

2.1 Mobility

Various models try to capture patterns in human mobility [8, 15, 21, 22]. Some of these models are quite successful in predicting a user's next location, sometimes with accuracies as high as 90%. We take these results further by augmenting location predictions with time predictions, allowing us to produce actionable information for predictive and proactive data dissemination. We do this by using fingerprints for identifying distinct routes in the user's mobility history, and probabilistically analyzing associated route times.

A route is defined as a recorded mobility edge between two locations with an identifying precedent (its fingerprint) and measured route time. This can be easily constructed from GPS traces, associate cell towers etc. The model, M, for each route r consists of:

$$M[r] = \{\mu_r, \psi_r, \sigma_r, e\} \tag{1}$$

Where μ_r is the average route time, $\psi_r$is the percentile distribution of route times, σ_r is the standard deviation and e is the error rate in prediction. This model is maintained for observed distinct routes over the history of mobility. The statistical information for route times is used in conjunction with a second order Markov model for route fingerprints. The error rate is used to capture mispredictions over time and adapt accordingly.

The fingerprint for identifying a route is the previous two locations the user visited. This information is recorded in a second order Markov chain. However, this fingerprint is not completely unique, as a preceding context can lead to several routes. We capture this divergence in a *fingerprint matrix*. This sparse matrix has the feature that routes with a shared fingerprint have a common entry that holds row level information such as the last time the fingerprint was observed and the bias for route selection. For each route in the matrix, we maintain the state given in equation 1. Route time predictions are then made as follows.

When a user advances to a new location, a fingerprint is produced by concatenating this location with its previous location. Meanwhile, this new mobility history is used to slightly modify the route selection bias for the *previous* fingerprint, as well as the statistical model for the travelled route. The row level storage for the new fingerprint is updated to reflect the current occurrence of the context. Then, the route selection bias for this new fingerprint is consulted for deciding which route to select for prediction. For the route chosen, the timestamp for its fingerprint, along with the statistical model of the route, is used to bound the estimated arrival time at the route's end point. Each time prediction is also associated with a confidence score that measures how well the model did in the last few predictions. Coupled with a second order Markov model, our system was able to predict time of arrival within an hour in more than 90% of the hits in the CRAWDAD human mobility dataset [22].

2.2 Contextual Access Feedback

Data access patterns have been used to inform system designs. For example, temporal and spacial locality in file I/O is used to optimize memory and storage devices. On the other hand, there are several systems that take advantage of clusters that get formed when users often access some files together [7, 13]. We posit that data access patterns could also be correlated with the context of access, such as location and time, which provides further guidance in data placement. These contextual patterns emerge over time, and our framework enables applications to use this feedback for fine tuning future placement requests. Applications can combine the contextual feed-

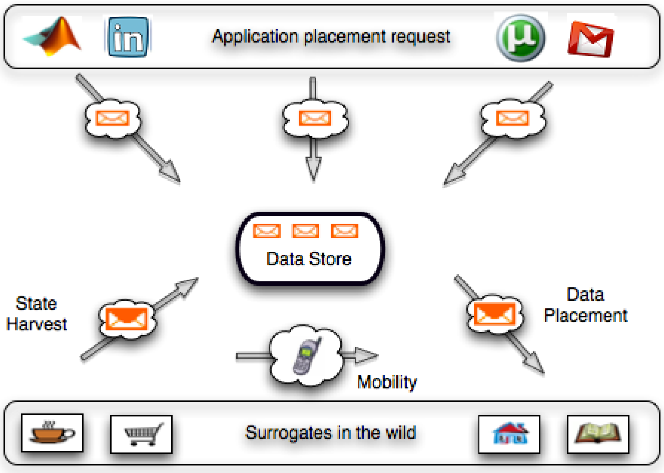

Figure 1: As the individual (and by extension the mobile device) moves in its environment, the moving cloud captures and extracts patterns in mobility and contextual data access. This information is used for predictively placing relevant content in the users future vicinity. When the state of a placed content it altered, the new state is harvested so that future placements could be up to date.

back with their knowledge of file dependency in leveraging the moving cloud.

2.3 Data delivery

With actionable mobility prediction at hand, our framework provides a secure infrastructure for disseminating data from the data source to destination nodes. Since most nodes are shared by multiple users, some possibly malicious, the moving cloud uses end-to-end encryption by default. Encryption keys and hash values for content verification can be directly retrieved from the data source when the user arrives and after proper authentication. Applications can tighten or loosen these security provisions based on their expected operating environment.

In the common use case for the moving cloud, it is often necessary to harvest any residual state at nodes once the user has processed delivered content. This allows future placements to be up-to-date with the most recent state generated by the user. Our framework provides the necessary primitives for harvesting state, and moving data around in the system. Since the platform is content agnostic, applications are responsible for managing the internal semantics of their data.

3 Applications in Challenged Networks

The serious lack of network resources in many developing countries makes accessing data significantly difficult. With very low bandwidth and extremely high latency [17], even simple network tasks become unpleasant. Various forms of caching and prefetching have been suggested to help with this situation [3, 19]. These solutions allow users to cache commonly requested data, or prefetch content for future access upon request.

The nature of these solutions makes them ideal candidates for adopting the moving cloud. Since humans are habitual, and have repeated patterns in their day to day mobility, we can use this behavior in automating data distribution in challenged environments. Even in the absence of widely available GPS capable devices, we can use location information at the granularity of internet kiosks or libraries visited by the user for predictive delivery, thereby eliminating most of the user wait time during data access. For a kiosk operator, predictive placement can allow for better scheduling and resource utilization, as kiosk capacity often varies throughout the day. Lets consider how such a service can be built.

As the user moves around her environment, accessing data from various locations, the moving cloud could capture and extract repeated patterns. Rather than reactively waiting for the users request, often leading to long access delays while data is downloaded over a slow link, future data access can be automated through the moving cloud. We have modified Sulula [19], a data prefetching solution for challenged environments, to take advantage of the moving cloud for predictively placing content. In the modified system, data in no longer requested on-demand, but predictively placed using our framework, realizing users significant time savings in a typical email and news reading session. For rich media content, the savings can be upwards of an order of magnitude.

4 Applications for Augmented Computation

As briefly discussed in the section 1.1, augmenting the storage and computational capacity of mobile devices with surrogates in the vicinity holds a viable promise for overcoming the latency problem [4, 7, 20]. A predictive moving cloud helps in these cloudlet implementations by making customized virtual environments, needed for resource augmentation, available ahead of request time. The benefits of this approach are three fold. First, the delay in transferring (or assembling) these computational and storage environments is eliminated, allowing users on the go to have access to readily waiting resources. Second, it avoids the difficulty of synchronizing state between various incompatible copies of an otherwise functionally similar environment. Finally, these computational environments could easily be upgraded to their latest versions without causing a brittle dependency cycle among multiple building pieces.

5 Challenges

A complete implementation of the moving cloud will require overcoming a number of technical as well as service model challenges. We will elaborate on some of these challenges in this section, and provide our thoughts on addressing them.

Phase changes in human mobility: a key element in human mobility is eventual divergence from the established mobility pattern. For example, people take vacations, or change cities altogether. Mobility models need to cope with these changes in order to provide a reasonable performance in transition, as well as in the new phase. One approach for tackling this challenge is implementing a quick learning scheme based on prediction error rates. A sudden increase in mispredictions can often be a good clue for identifying a new phase. In these situations, past mobility history becomes less relevant in making future predictions. As a result, avoiding to use this outdated history is just as important as learning the new pattern. On the other hand, the mobility model needs to have enough "memory" for it to recognize only temporary shifts in phase.

Data consistency: like most distributed systems, the moving cloud faces the challenge of providing a consistent view to data that flows through it. The problem is even more interesting in the moving cloud because its consistency model needs to consider data access in predictively delivered content. Data that was consistent at time of delivery is not necessarily consistent at the time of access. In addition, when considering delivery nodes distributed in the wild with numerous administrative domains, network partition is a given. The moving cloud needs to have a robust versioning mechanism coupled with an appropriate consistency model to enforce horizontal as well as vertical integrity. By facilitating version updating among delivery nodes, as well as with the cloud data store, the moving cloud can provide some consistency guarantees.

Degree of certainty in data delivery: at the core of probabilistic mobility models is some level of uncertainty about the future. These models in general try to manage this uncertainty by using some confidence score based on established patterns, past prediction hits and surrounding contexts. Still, acting upon these predictions needs to be informed by the resources utilized in doing so. This is especially important in environments where resources are limited to begin with. As a result, it becomes necessary to define some threshold for predictive delivery based on our confidence on mobility predictions. Setting this threshold requires understanding the operating environment and the available resources. The moving cloud will need to tailor this decision, often on the fly, based on active or passive observations of the environment.

While we have focused on some of the technical challenges for implementing the moving cloud, a practical deployment also involves several interesting service model challenges. These include node availability, administration and maintenance, revenue models and application availability.

6 Related Work

Deeper analysis of individual mobility records reveals that patterns emerge over time. Several approaches have been suggested for capturing these patterns, ranging from Levy flights [11] to Markov [6, 22] and Lempel-Ziv [25] parsing models. A levy flight is a type of random walk such that the step size of a walk follows a power law distribution. Markov model based algorithms assume the probability of an individual visiting a particular location only depends on the most recent locations visited, and this probability is the same anywhere the context is the same. On the other hand, Lempel-Ziv based approaches use Markov like models whose historical length of a context is variable, and is allowed to grow without a limit as needed. These models have the common goal of predicting future locations for an individual based on history. Our approach improves these models by adding time bounds to their location prediction.

Automatic placement of application data has been suggested in various contexts. Systems such as Emerald [10] and Globe [24] focused on providing programming abstractions and migration mechanisms for moving data and computation. LAN level data placement was tackled in several systems [2, 9] that facilitate application partitioning among a set of machines. Volley [1] is a data center level solution that analyzes logs of datacenter requests by users and recommends data placement among a set of geographically distributed datacenters, so that user access time can be reduced. The moving cloud shares the goal of reducing latency, but its approach is much more fine grained. We analyze personal mobility patterns and contextual data access behavior for predictive data placement. This allows mobile devices to access data from nearby machines where the roundtrip latency is much lower.

7 Conclusion

As available bandwidth increases, the role of latency as the system bottleneck becomes even more pronounced [12]. This is particularly concerning as pervasive computing becomes the norm, and cellular networks provide the first order of connectivity for devices on the go. In challenged networks where resources are limited, the situation is even worse. This paper presented the moving cloud, a proactive data delivery framework that leverages route fingerprints in individual mobility with users' contextualized behavior of data access. The moving cloud trades bandwidth and storage for latency by predictively placing content where it needs to be in the near future. This paradigm enables a number of networking scenarios ranging from on-demand social networks to mobile resource augmentation and personalized content distribution networks.

References

[1] S. Agarwal, J. Dunagan, N. Jain, S. Saroiu, A. Wolman, and H. Bhogan. Volley: Automated data placement for geo-distributed cloud services. In *NSDI*, 2010.

[2] K. Amiri, D. Petrou, G. R. Ganger, and G. A. Gibson. Dynamic function placement for data-intensive cluster computing. In *USENIX Technical Conference*, 2000.

[3] A. Badam, K. Park, V. S. Pai, and L. L. Peterson. Hash-Cache: cache storage for the next billion. In *NSDI'09*.

[4] R. Balan, J. Flinn, M. Satyanarayanan, S. Sinnamohideen, and H.-I. Yang. The case for cyber foraging. In *EW 10: Proceedings of the 10th workshop on ACM SIGOPS European workshop*, pages 87–92, 2002.

[5] N. Dell, N. Breit, T. Chaluco, J. Crawford, and G. Borriello. Digitizing paper forms with mobile imaging technologies. In *Proceedings of the 2nd ACM Symposium on Computing for Development*, ACM DEV '12, pages 2:1–2:10, 2012.

[6] N. Eagle, A. Clauset, and J. A. Quinn. Location segmentation, inference and prediction for anticipatory computing. *AAAI Spring Symposium*, 2009.

[7] J. Flinn, S. Sinnamohideen, N. Tolia, and M. Satyanaryanan. In *FAST '03*, pages 15–28.

[8] M. C. Gonzalez, C. A. Hidalgo, and A.-L. Barabasi. Understanding individual human mobility patterns. *Nature*, 453(7196):779–782, June 2008.

[9] G. C. Hunt and M. L. Scott. The coign automatic distributed partitioning system. In *OSDI '99*, pages 187–200.

[10] E. Jul, H. Levy, N. Hutchinson, and A. Black. Fine-grained mobility in the emerald system. *ACM Trans. Comput. Syst.*, 6(1):109–133, 1988.

[11] J. Klafter, M. F. Shlesinger, and G. Zumofen. Beyond brownian motion. *Physics Today*, 49(2):33–39, 1996.

[12] L. Kleinrock. The latency/bandwidth tradeoff in gigabit networks. *Communications Magazine, IEEE*, 30(4):36 – 40, apr. 1992.

[13] G. H. Kuenning and G. J. Popek. Automated hoarding for mobile computers. In *SOSP '97*, pages 264–275.

[14] H. A. Lagar-Cavilla, N. Tolia, E. de Lara, M. Satyanarayanan, and D. O'Hallaron. Interactive resource-intensive applications made easy. In *Middleware '07*, pages 143–163.

[15] A. J. Nicholson and B. D. Noble. Breadcrumbs: forecasting mobile connectivity. In *MobiCom '08*, pages 46–57.

[16] D. A. Patterson. Latency lags bandwith. *Commun. ACM*, 47(10):71–75, 2004.

[17] B. Petrazzini and M. Kibati. The Internet in developing countries. *Commun. ACM*, 42(6):31–36, 1999.

[18] N. Rangaswamy and E. Cutrell. Anthropology, development and icts: slums, youth and the mobile internet in urban india. In *Proceedings of the Fifth International Conference on Information and Communication Technologies and Development*, ICTD '12, pages 85–93, 2012.

[19] A. Reda, B. Noble, and Y. Haile. Distributing private data in challenged network environments. In *WWW '10*, pages 801–810.

[20] M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies. The case for vm-based cloudlets in mobile computing. *IEEE Pervasive Computing*, 8:14–23, 2009.

[21] C. Song, Z. Qu, N. Blumm, and A.-L. Barabasi. Limits of Predictability in Human Mobility. *Science*, 327(5968):1018–1021, 2010.

[22] L. Song, D. Kotz, R. Jain, and X. He. Evaluating location predictors with extensive wi-fi mobility data. *SIGMOBILE Mob. Comput. Commun. Rev.*, 7(4):64–65, 2003.

[23] N. Tolia, D. G. Andersen, and M. Satyanarayanan. Quantifying interactive user experience on thin clients. *Computer*, 39:46–52, 2006.

[24] M. van Steen, P. Homburg, and A. S. Tanenbaum. Globe: A wide-area distributed system. *IEEE Concurrency*, 7:70–78, 1999.

[25] J. Ziv and A. Lempel. Compression of individual sequences via variable-rate coding. *IEEE Transactions on Information Theory*, 24(5):530–536, 1978.