

Research Summary

Leana Golubchik

March 27, 2010

My research interests are in the area of computer systems modeling and performance evaluation. My earlier work largely focused on (a) development of models and efficient methodologies for evaluation of systems at design time - specifically, on techniques that lead to reasonably simple *abstractions* (needed for ease of applicability) and *efficient* solution techniques (needed for rapid evaluation of design ideas), and (b) quality-of-service-based design of large-scale continuous media (CM) *storage systems* and their subsequent performance evaluation - efficient storage system designs are fundamental to the viability of a broad range of systems and applications.

More recently, one aspect of my research focused on design and evaluation of large-scale peer-to-peer (p2p) and overlay systems. Addressing challenges in p2p and overlay systems can lead not only to significant improvements, but also to significant insight - not only into p2p and overlay systems but into distributed systems in general. Such understanding and subsequent ability to provide quality-of-service (QoS) and performance prediction relies on thorough evaluation. Doing such evaluation through *analytical modeling* can lead to (a) quick evaluation of design-time choices and (b) insight through better understanding of systems. Hence, my work in this area has largely focused on (i) development of accurate analytical models and (ii) design of approaches that can lead to QoS predictions.

Another aspect of my research has been reliability evaluation of *software* systems. High level abstractions that allow for estimating reliability of software *architectures* (rather than implementations) can provide insight into the reliability characteristics of a software's design while facilitating timely and less costly design improvements. Given the complexity of modern software, scalability of the evaluation methodology is a critical concern. Moreover, wide-spread adaptation of such techniques by software engineers (largely) hinges on them "fitting into" existing design processes. Hence, I work on analytical reliability evaluation methodologies that (a) are easily derivable from architectural designs and (b) lead to *efficient* solution techniques, needed for rapid evaluation of design ideas.

I have also focused on analysis of data generated by wireless sensor systems. Such systems can facilitate scientific studies by instrumenting the real world and collecting corresponding measurements, with the aim of detecting and tracking phenomena of interest. Given the huge volume of measurements such deployments produce, accurate and robust *automated* processing of data is fundamental to its meaningful use. Thus, I expect our work to be of use to a broad range of current and future sensor systems and their applications.

1 Large-scale P2P and Overlay Systems

As the demand on content delivery systems grew, p2p-based architectures emerged as viable and cost-effective approaches to satisfying such systems' requirements¹. Given the significant interest from researchers and Internet users alike, improvements in p2p systems' performance is likely to

¹Some estimates put p2p traffic at 75% of the bytes flowing on the Internet today.

have an impact on a large community. Since such architectures need to be highly efficient, highly scalable, and highly distributed, appropriate design and evaluation of p2p-type systems is fundamental to the future of distributed systems in general.

This part of my research has focused on the design and evaluation of highly distributed p2p systems as well as overlay systems. To focus this research further, I have concentrated my efforts on the following content delivery applications: (a) *file downloads*, i.e., applications where the entire content is delivered before it is consumed (e.g., software upgrade downloads); (b) *live streaming*, i.e., applications where continuous media (CM) is being viewed as it is being generated (e.g., live sporting events); and (c) *video-on-demand*, i.e., applications where CM was recorded at an earlier time, and it can either be viewed as it is being delivered, or large portions of it can be downloaded before viewing begins. Many such applications require high QoS in order to be commercially viable - inability to control the resulting QoS is an important barrier for their wide-spread development and use (and is the reason commercial p2p systems include dedicated servers).

Many challenges in such systems can be viewed in light of two fundamental issues: (1) *incentives*, i.e., provision of appropriate incentives for peers to contribute their resources such that the entire system performs well according to metrics appropriate for each application; and (2) *quality-of-service*, i.e., designing protocols and architectures that are amenable to provision of QoS and performance prediction. My work focuses on gaining insight into these fundamental issues, which I believe will lead to the development and proliferation of applications that are able to provide high QoS that is currently lacking. Challenges to being able to address these issues include: (1) dynamic and heterogeneous nature of p2p and overlay environments; (2) ability to use resources efficiently with little or no centralized control; (3) design and analysis of models that are able to capture sufficient aspects of such systems to be able to understand their “interesting” behavior and to facilitate better understanding of how such systems work and where improvements are needed; and (4) design and analysis of protocols and architectures that are robust to the dynamic and heterogeneous nature of such environments and that are subsequently able to provide QoS.

In the context of large-scale overlays, I have also continued my efforts on scalable *upload* applications over wide-area networks, where we have focused on design of an application layer infrastructure that improves *scalability*, *security*, *fault tolerance*, and *performance* of current and future Internet-based services. To the best of my knowledge, our work was the first to focus on scalability and efficiency of upload applications.

1.1 P2P Systems

Incentives. We explore incentive provision in the context of BitTorrent (BT) [D12]. Although, BT “forces” peers to share their resources *while* attempting to complete their downloads (through the tit-for-tat (TFT) mechanism) there are still opportunities for *free-riders* (i.e., peers who do not contribute resources) to download files through: (1) capacity provided by *leechers* (nodes who have not completed their download yet) through the optimistic unchoking mechanism (used by leechers to probe for new downloading opportunities), and (2) capacity provided by *seeds* (nodes that have completed their download but continue to contribute their uploading capacity to the system). Sufficient evidence exists that: (1) opportunities for free-riding hurt the system’s performance, and (2) a non-negligible number of nodes in BT participate as seeds, which significantly contributes to providing reasonable performance for the free-riders. Moreover, a leecher’s progress through the downloading process is slower at the beginning (when having few chunks prevents a node from participating in TFT) and at the end (when it is more difficult to find peers with the few remaining pieces of needed

data). In both cases, seeds can contribute significantly, as they have complete data and are no longer playing the TFT game. Since, the seeds' behavior is an important factor in facilitating free-riding and in aiding contributing leechers, our main idea was to explore alternative approaches for seeds to contribute their upload capacity to the system, with the goal of degrading free-riders' performance while (if possible) improving other leechers' performance. To the best of our knowledge, no other study before ours explored this.

Although empirical evidence suggests that many nodes participate in *multiple torrents*², little research literature exists on this topic. This can be exploited to increase the number of seeds in a given torrent, which in turn contributes to (a) helping newly joined nodes ramp up so that they can become contributing peers faster, (b) helping nodes nearing the end of their downloads find the last few file chunks faster, and (c) keeping a torrent "alive" longer (i.e., making sure that all file chunks are available in the system). Thus, our work [D8,A4] focused on (i) what incentives could be provided for nodes to contribute resources as seeds in a multi-torrent environment, and (ii) what are the resulting performance consequences, both, for the nodes willing to be seeds and for the overall system. We found that the current system lacks incentives for nodes to stay around as seeds in a multi-torrent environment. Motivated by that, we proposed a cross torrent based method and illustrated, through an extensive performance study, the benefits of such an approach.

QoS/performance prediction. We have explored the different aspects of QoS and performance prediction through the use of analytical models and protocol design. For instance, we have developed a simple, yet extensive analytical model of BT [D7]. To the best of our knowledge, it is the first (*steady state*) analytical performance model of a *heterogeneous* BT system that includes the behavior of seeds and free-riders (as noted earlier, both are important). The accuracy of our model can be attributed to including important BT characteristics, namely (i) imperfect clustering in regular (TFT-based) unchokes and (ii) bias in optimistic unchokes. This accuracy is validated through simulations and PlanetLab experiments, which also demonstrate the importance of including imperfect clustering and biased optimistic unchoking in accurately predicting nodes' download times. Our model is quite extensible and can be used to evaluate future protocol modifications - e.g., we demonstrated our model's extensibility by explicitly modeling two variations of BT: (a) the well known large view exploit, and (b) our proposed fix for the large view exploit (the novel seeding approach described above [D12]).

In a p2p-based streaming system, packets can arrive at a node in any order, but they must be played back in a specific order (and at a specific rate), corresponding to the original recording rate of the stream. We can view the subset of the edges (between nodes in a p2p overlay) used for packet delivery as forming a mesh. The system's performance is largely a function of the algorithms used to construct and maintain this mesh as well as the algorithms used for scheduling packet delivery over the mesh. Our work [D6] focused on approaches to constructing such meshes, where the primary goal was to develop an understanding of two related quantities – playback delay and buffering requirements. To this end, we explored two approaches to mesh construction and subsequent streaming, one based on multi-trees and the other based on hypercubes. We used these approaches to explore the resulting playback delay, buffer space, and communication requirements³. In our work we focused on a "structured" approach to mesh construction (i.e., the set of edges used for delivery of packets was fixed by our algorithms) as compared to an "unstructured" approach (i.e., the edges used for delivery are determined on a per packet basis, essentially on the fly when the

²One study indicates that > 85% participate in multiple torrents.

³By communication requirements we mean the number of neighbors with which a node must communicate.

data is needed). Although unstructured techniques allow the system to more easily adapt to node churn, existing unstructured approaches to *streaming* (rather than file downloading) are mostly “best effort”, and little exists in the way of formal analysis of resulting QoS prediction. In contrast, an advantage of our approach is that it has *provable quality-of-service (QoS)*, where we provide analysis of corresponding performance characteristics.

While many efforts focused on the common challenges faced by live and pre-recorded (Video-on-Demand) streaming applications, our efforts focused on some of the open questions in designing p2p-based Video-on-Demand (VoD) systems specifically [D4]. We considered a BT-like VoD system and focused on: (i) how the lack of load balance, which typically exists in a p2p-based VoD system, affects performance and what steps can be taken to remedy that, and (ii) is a FCFS approach to serving requests sufficient or whether a Deadline-Aware Scheduling (DAS) approach can lead to performance improvements. Motivated by the lack of high QoS in streaming systems, we proposed several practical schemes aimed at addressing the above stated questions and illustrated the benefits of our techniques through an extensive performance study.

1.2 Multi-path Streaming

We have also been carrying over our efforts on QoS-oriented design from storage systems [A5,A8, D23,D17] to delivery of CM data over the Internet. The poor QoS in streaming over the Internet is partly due to variations in delays, bandwidth limitations, and packet losses. Although streaming applications can tolerate some missing data, non-recoverable information loss degrades these applications’ QoS. Consequently, a number of application areas have backed away from streaming of their content over the Internet. Inability to control the resulting visual and auditory quality of the streamed presentation is an important reason for such a trend.

We believe that this trend can be reversed. To this end, our work focused on exploring high quality streaming through the exploitation of *multiple paths* existing in the network (between a set of senders and a receiver). By high quality, we mean with significant bandwidth requirements, of relatively long duration, and without information loss or hiccups in data delivery.

Our goal was to design an *application-level* approach, i.e., one that can be easily deployed over the Internet, without requirements for support from/modifications to the lower layers of the network. Our focus was largely on providing a fundamental understanding of the benefits of using *multiple paths* to deliver CM data destined for a particular receiver. (This data is fragmented into packets and the different packets take alternate routes to the receiver, e.g., by streaming the data through an overlay of relays distributed over a wide area network. As each packet is sent on only one of the paths, our approach does not increase the overall workload on the network.)

Existence of multiple paths (with *disjoint bottlenecks*⁴) includes the following potential benefits: (a) reduction in correlations between consecutive packet losses; (b) increased throughput; (c) improved robustness and ability to adjust to variations in congestion patterns and failures. Our work focused on the loss characteristics aspects. Using a Gilbert model (GM), we gave [A14] an analytical characterization of when a multi-path (MP) approach is beneficial, as compared to a single path (SP) approach. Our results indicated that: (1) in general, MP streaming exhibits better loss characteristics than SP streaming, (2) use of an erasure code may not necessarily improve data loss characteristics in the case of SP streaming, while MP streaming (with or without use of an erasure code) can improve data loss characteristics, and (3) lag1-autocorrelation of MP streaming is usually

⁴Such paths do not have to be completely disjoint; it is sufficient for them to have disjoint bottlenecks.

closer to zero than that of SP streaming, which should also result in a higher viewing quality of the received data. It was also indicated that the average error burst length of MP streaming is statistically shorter than that of SP streaming.

Given these benefits, another important question is how to distribute the load among multiple paths so as to optimize viewing quality. (The setting and possible adaptation of what fraction of data is sent on each path can be done by the receiver, based on its perceived QoS.) We studied [D20] the problem of assigning CM data to the multiple paths, according to the path characteristics, such that a certain specified performance metric is optimized. This was done under the GM, while considering appropriate optimization objectives, with the goal of improving the perceptual quality of the streamed media. Determining and gaining insight into suitable optimization objectives is also non-trivial and important (e.g., optimizing the lag-1 autocorrelation may result in higher loss rates observed at the receiver, when the paths are not homogeneous). Since there is a fundamental tradeoff between the frequency of losses and the corresponding loss correlations, it is natural to consider optimization objectives that encompass both metrics.

A great deal of our work was done under a “conventional Gilbert model”. A limitation of using such a model is that the loss process of a path is *independent* of the bandwidth requirements of the streaming application. Our later effort [A9] proposed the use of a *functional Gilbert model* (FGM) as a more general approach to characterizing the bursty loss nature of a path as well as its dependency on an application’s bandwidth requirements.

We verified our analytical and simulation results under real world conditions, by building a p2p-based multi-path streaming prototype [A11] and using it for streaming video experiments between UMD, CUHK, and USC. The choice of a p2p-based architecture was motivated by the parallelism between simultaneous downloads in p2p systems and multi-path streaming.

1.3 Scalable Infrastructure for Wide-area Uploads

Hotspots are a major obstacle to achieving scalability in the Internet; they are usually caused by either high demand for some data or high demand for a certain service. At the application layer, hotspot problems have traditionally been dealt with using some combination of increasing capacity, spreading the load over time and/or space, and changing the workload. These classes of solutions have been studied in the context of one-to-many, many-to-many, and one-to-one type communication. However, to the best of our knowledge ours was the first effort [D37,D33,B1,A13], on making applications using *many-to-one* communication scalable and efficient. This corresponds to an important class of *upload* applications (e.g., submission of income tax forms to IRS, submission of papers to conferences, Internet-based storage, certain aspects of cloud computing, and many more).

The main focus of our early work was on scalable infrastructure design for wide-area upload applications, where we proposed *Bistro* [D37], an architecture which employs an overlay of (untrusted) hosts for improving the efficiency and scalability of uploads. Our basic idea was based on the observations that (a) existence of hotspots in many upload applications is due to approaching deadlines and long transfer times and (b) what is actually required by many upload applications is an assurance that specific data was submitted before a specific time, and that the transfer of the data needs to be done in a timely fashion, but does *not* have to occur by that deadline. Thus, our approach was to break the original deadline-driven upload problem into: (1) a real-time *timestamp* subproblem, where we ensure that the data is timestamped and that the data cannot be subsequently tampered with; (2) a low latency *commit* subproblem, where the data goes “somewhere” (to an intermediary) and the user is assured that the data is safely and securely “on its way” to its destination;

and (3) a timely *data transfer* subproblem, which can be carefully planned (and coordinated with other uploads) and results in data delivery to the original destination.

Bistro's ability to share an infrastructure, such as an infrastructure of proxies, between a variety of wide-area applications has clear advantages over the more traditional solutions. In [B1] we conducted a performance study which demonstrated the potential performance gains of the *Bistro* framework as well as provided insight into the general upload problem. Since confidentiality of data as well as other security issues are especially important in upload applications and in our solution where we introduced untrusted (public) intermediaries (i.e., *bistros*), we also developed [D33] a secure data transfer protocol within the *Bistro* framework, which not only ensures the privacy and integrity of the data but also takes scalability considerations into account.

In later efforts we focused on performance and reliability improvements to the basic *Bistro* design along the subproblems outlined above [D19,D21,A10,D27,A1,D29,D30]. For instance, we obtained [A10,D27] interesting results in the context of the timely data transfer subproblem (described above), which can be viewed more generally as the problem of collecting a large amount of data from a set of hosts to a single destination in a wide-area network. Often, due to congestion conditions, the paths chosen by the network may have poor throughput. By choosing an alternate route at the application level (e.g., by using relays in an overlay), we may be able to obtain substantially faster completion time. This data collection problem is a non-trivial one because the issue is not only to avoid congested link(s), but to devise a *coordinated* transfer schedule which would afford maximum possible utilization of available network resources. In our work, we made no assumptions about knowledge of the topology of the network or the capacity available on individual links of the network, i.e., we only used end-to-end information. We used a network flow based solution, utilizing *time-expanded* graphs, which gave us an *optimal* data transfer schedule with respect to the makespan metric, given the constraints on knowledge of the topology and capacity of the network. Experimentally, we established that the lack of knowledge of the paths, provided by the network to send data, are not a significant barrier. Our approach to computing coordinated data collection schedules resulted in significant performance improvements. In general, our approach can be used for solving arbitrary data movement problems over the Internet, and we used the *Bistro* platform to illustrate one such application.

Another direction where we considered a general problem, motivated by the *Bistro* architecture, is that of digital signature schemes' performance [A1,D29]. Digital signature schemes are widely used in many applications; however, the performance of an Internet server computing digital signatures online is limited by the high cost of modular arithmetic. One way to improve the server's performance is to reduce the number of computed digital signatures by combining a set of documents into a batch (in a smart way) and signing each batch only once. This approach could reduce the demand on the CPU but requires additional network bandwidth (for sending extra information to clients). Our work provided a framework (using semi-Markov models) for studying as well as analyzing performance of a variety of online digital signature batching schemes. We extensively validated our framework's accuracy and effectiveness. Our results showed that substantial computational benefits can be obtained from batching without significant increases in the amount of additional information that needs to be sent to the clients.

2 Estimating Reliability of Software Architectures

Early prediction of reliability⁵ is important in building dependable software systems, as problems discovered in later stages (e.g., after implementation), can be quite costly to address. Thus, it is important to estimate reliability at the *software architecture* level. A number of efforts (prior to ours) focused on system-level reliability prediction, essentially assuming that reliabilities of components (making up the system) are known. Our work [D11] in this area strived to remedy the shortcomings of previous approaches by proposing a framework for predicting reliability of software *components* during architectural design. Important challenges in component reliability prediction at architectural design time are due to uncertainties and lack of information in the early stages of design and development. Our work focused on dealing with two such uncertainties: (1) lack of a component’s operational profile (since it has not been implemented yet) and (2) its failure information (since systems are designed for correct behavior, failure modes are not typically part of an architectural specification). Our evaluation and validation efforts (as compared to implementation-level techniques) indicated that our framework can meaningfully assess a component’s reliability.

Predicting a component’s reliability was an important first step to system reliability prediction, on which we focused next. Our work on system-level reliability [D2] is distinct as it focused on (a) predicting reliability of *concurrent* software systems in a *scalable* yet accurate manner, while (b) taking *contention between software resources* into account. The improved scalability is achieved through a *hierarchical approach*. Most existing techniques assume a sequential system. In predicting reliability of concurrent systems, one typically needs to keep track of the status of all components. Often, such approaches essentially generate reliability models in a brute-force manner and thus suffer from scalability (“state explosion”) problems, making them often prohibitively costly (for generating and solving models). To this end, we proposed a hierarchical reliability prediction framework, where we approached the scalability problem by generating and solving submodels that each capture a part of the system’s functionality. In doing so, we leveraged system use-case *scenario* models (e.g., such as UML sequence diagrams). Specifically, we solved finer granularity scenario-based submodels and a coarser granularity system model; we then combined their results to obtain a system reliability estimate. The motivation here is that the submodels are expected to be relatively small and that solving a number of smaller submodels (rather than one huge model) results in space and computational savings.

We were able to achieve significantly better scalability without sacrificing the level of detail modeled. That is, some existing techniques model a component as being either on or off; doing so gives information about which component failed but not about what causes the failure. By using a hierarchical approach, we are able to retain significant level of detail about the system being modeled, while modeling it in a scalable manner. Moreover, our approach also allows for modeling of *contention* between software components. (In a concurrent system, components share resources, e.g., services provided by other software components.) We validated the accuracy of our approach and illustrated that in practice significant space and computational benefits are obtained. Our evaluation also indicated that it is important to model contention in predicting a concurrent system’s reliability, as the resulting accuracy is improved.

⁵Here, reliability refers to the probability that a system will perform its intended functionality.

3 Wireless Sensor Systems

Wireless sensor systems have significant potential for aiding scientific studies by instrumenting the real world and collecting measurements, with the aim of observing, detecting, and tracking scientific phenomena that were previous only partially (or not at all) observable or understood. Such technology can facilitate environmental protection, public health, and prevention of economic losses and life. However, one obstacle to achieving the full potential of such systems, is the ability to process, in a timely and meaningful manner, the huge amount of measurements they collect. Given such large volumes of data, one natural question might be: *Can we devise an efficient automated approach to identifying the “interesting” parts of these data sets?* We can view identification of such “interesting” or “unexpected” measurements (or events) in collected data as anomaly detection (we use the generic term “anomaly” for all interesting (typically, other-than-normal) events occurring either on the measured phenomena or the measuring equipment). Automated online (or real-time) anomaly detection in measurements collected by sensor systems has been the topic of my recent work.

Our initial efforts [D14,A3] focused on what can be viewed as a special case of anomaly detection, namely *fault detection*. Sensor network measurement studies have reported instances of transient faults in sensor readings; thus, we sought to answer a simple question: How often are such faults observed in real deployments? We explored and characterized four qualitatively different classes of fault detection methods: (1) rule-based methods that leverage domain knowledge to develop heuristic rules for detecting and identifying faults; (2) estimation methods that predict “normal” sensor behavior by leveraging sensor correlations, flagging anomalous sensor readings as faults; (3) time series analysis based methods that start with an *a priori* model for sensor readings and compare a sensor measurement against its predicted value to determine if it is faulty; and (4) learning-based methods that infer a model for the “normal” sensor readings using training data, and then statistically detect and identify classes of faults. We found that these classes of methods sit at different points on the accuracy/robustness spectrum. Rule-based methods can be highly accurate, but their accuracy depends critically on the choice of parameters. Learning methods can be cumbersome to train, but can accurately detect and classify faults. Estimation methods are accurate, but cannot classify faults. Time series analysis based methods are more effective for detecting short duration faults than long duration ones, and incur more false positives than the other methods. We applied these techniques to real-world sensor data sets and found that all methods were qualitatively consistent in identifying sensor faults and that the prevalence of faults and their type varied with data sets. This work was an initial step towards automated on-line fault detection and classification.

Recently, we have expanded our efforts to more general anomaly detection [F1]. A good anomaly detection methodology for sensor systems should be able to accurately identify many types of anomalies, be robust, require relatively little resources, and perform detection in (near) real-time. Hence, we focused on an approach to *online* anomaly detection in measurements collected by sensor systems that exhibits these characteristics. Our goal was to detect anomalies without “pre-defining” what is an anomaly⁶. Briefly, our approach leverages temporal and spatial correlations in sensor systems measurements and constructs (what is essentially) a piecewise linear model of sensor data time series. To detect anomalies, we compare the piecewise linear models of sensor data (collected during a time interval) and a reference model (also computed online), where significant differences (as determined by a similarity metric) are flagged as anomalies. Our extensive evaluation study, using real sensor system deployments data, illustrates that our approach is accurate, robust, and efficient. We believe this to be a promising direction as the results also indicate that our *online* approach is

⁶Unlike, for instance, some approaches that view anomalies as outliers.

more accurate than some of the (more computationally intensive) *offline* techniques.

In the context of (multi-hop) sensor networks, we have also studied energy efficient data compression and transmission. In such an environment, data compression is advantageous only when the transmission energy saved (from having to transmit less data) is greater than the energy expended in compressing the data. Thus, the aim of our work [D5] was to design a joint compression, transmission scheduling and transmission power control policy that minimizes the time average energy expenditure while achieving network stability. Our algorithm can achieve time average power expenditure arbitrarily close to the optimal, while adapting to topology, network dynamics, application data rates, and platform differences.

4 Research Directions

The scale and complexity of highly distributed systems as well as the *lack of information* and the *dynamic* nature of environments in which they operate are approaching a point where traditional performance evaluation techniques *alone* may no longer be sufficient.

My long-term goal is to address such problems through the synergy between stochastic performance modeling and machine learning techniques. For instance, machine learning techniques can aid with two difficulties that are typical in performance modeling - model complexity and parameter estimation. Such techniques have not been traditionally employed in this area, but I believe are emerging as a possible direction. In the near future, I plan to focus on the following directions.

Large-scale data centers are becoming a critical part of today's computing infrastructure. Their scale requires intelligent resource management techniques, without which it would not be possible to sustain their growth while achieving desired performance characteristics at low costs. In such environments data placement, management of energy costs, ability to predict level of service delivered to users are all critical issues. These concerns lead to difficult and interesting problems in resource management, capacity planning, detection and anticipation of workload changes, and so on. For instance, we have obtained initial results in workload characterization [D9], an essential component in performance modeling. Categorizing requests and their resource demands often requires significant monitoring infrastructure. Our initial work focused on automatically differentiating and categorizing requests without requiring sophisticated monitoring techniques. Using machine learning, our method requires only aggregate measures and does not assume prior knowledge of request categories or their individual resource demands.

Efficient (real time with little resource requirements) and effective (accurate and robust) anomaly detection techniques can lead to better understanding of sensor systems data, facilitation of scientific phenomena studies, and new methodologies for anomaly detection in a number of applications where large amounts of data, with little structure, are being collected. Thus, another research interest is in continuing our anomaly detection efforts along the directions of approaches that take advantage of spacial correlations across sensor system nodes as well as correlations between data collected by multiple sensors on the same node (as we expect sensor systems nodes, particularly in the future, to hold a number of diverse sensors).

Another area of interest, in the context of software reliability and performance estimation, is carrying over our efforts from the so-called green-field development scenarios (where all components are developed anew) to brown field development scenarios (where third party, off-the-shelf, components are used and details about their design and implementation are typically unavailable). In this context, we plan to focus on web services as a representative brown field scenario application.