# An Adaptive Cyberinfrastructure for Threat Management in Urban Water Distribution Systems

Kumar Mahinthakumar[1], Gregor von Laszewski[2], Ranji Ranjithan[1], Downey Brill[1], Jim Uber[3], Ken Harrison[4], Sarat Sreepathi[1], and Emily Zechman[1]

[1]North Carolina State University, Raleigh, NC, USA
{gmkumar, ranji, brill, sarat_s, emzechma}@ncsu.edu

[2]University of Chicago, Chicago, IL, USA
gregor@mcs.anl.gov

[3]University of Cincinnati, Cincinnati, OH, USA
jim.uber@uc.edu

[4]University of South Carolina, Columbia, SC, USA
harriskw@usc.edu

**Abstract.** Threat management in drinking water distribution systems involves real-time characterization of any contaminant source and plume, design of control strategies, and design of incremental data sampling schedules. This requires dynamic integration of time-varying measurements along with analytical modules that include simulation models, adaptive sampling procedures, and optimization methods. These modules are compute-intensive, requiring multi-level parallel processing via computer clusters. Since real-time responses are critical, the computational needs must also be adaptively matched with available resources. This requires a software system to facilitate this integration via a high-performance computing architecture such that the measurement system, the analytical modules and the computing resources can mutually adapt and steer each other. This paper describes the development of such an adaptive cyberinfrastructure system facilitated by a dynamic workflow design.

## 1 Introduction

Urban water distribution systems (WDSs) are vulnerable to accidental and intentional contamination incidents that could result in adverse human health and safety impacts [1]. The pipe network in a typical municipal WDS includes redundant flow paths to ensure service when parts of the network are unavailable, and is designed with significant storage to deliver water during daily peak demand periods. Thus, a typical network is highly interconnected and experiences significant and frequent fluctuations in flows and transport paths. These design features unintentionally enable contamination at a single point in the system to spread rapidly via different pathways through the network, unbeknown to consumers and operators due to uncertainty in the state of the system. This uncertainty is largely a function of spatially and temporally varying water usage. When a contamination event is detected via the first line of defense, e.g., data from a water quality surveillance sensor network and reports from consumers, the municipal authorities are faced with several critical questions as the contamination event unfolds: Where is the source of contamination? When and for how long did this contamination occur? Where should additional hydraulic or water quality measurements be taken to pinpoint the source more accurately? What is the current and near future extent of contamination? What response action should be taken to minimize the impact of the contamination event? What would be the impact on consumers by these actions?

Real-time answers to such complex questions will present significant computational challenges. We envision a dynamic integration of computational approaches (the conjunctive use of simulation models with Bayesian and optimization methods) and spatial-temporal data management methods, enabled within a grid-based software framework. Each component in the cyberinfrastructure can be represented at multiple scales within a hierarchy, and must mutually adapt according to the levels within the hierarchy of the associated components. For example, the level of accuracy of the chosen model must appropriately match the degree of refinement in the available data or steer the measurement system to obtain additional data. Similarly, the degree of sophistication of the search method must appropriately match the available computing resources on the grid or steer the grid brokering system to obtain more resources. Thus the cyberinfrastructure must facilitate dynamic interactions among its components through a mutually adaptive and steerable software system. This cyberinfrastructure will be enabled by recently developed grid-based resources and high performance networks. Developing and demonstrating such a cyberinfrastructure for adaptive contaminant assessment in municipal WDSs is the overall goal of this research.

## 2 Technical Objectives

The primary technical objectives of this DDDAS [2] research are to:

- Develop simulation procedures and search algorithms that can adapt to changing conditions including data and computational resources. These procedures are part of a strategic decision-making framework for source characterization, hydraulic control, and activation and deployment of sensors for threat management in urban water distribution systems.
- Implement a grid-enabled dynamic work flow engine that can adaptively assemble and drive various data, computational, and computer resource components for changing conditions and demands. This engine will enable tractable execution of the algorithms as well as real-time processing of sensed data to help make rapid, tactical decisions related to hydraulic control and confirmatory sampling.
- Test and evaluate the work flow engine and the associated components for hydraulic control, water quality and hydraulic sensor network design and confirmatory sampling procedures for an array of threat management scenarios using computer simulations.

## 3 DDDAS Architecture

The integrated adaptive cyberinfrastructure will contain several components that communicate with each other (Fig. 1). These components can be abstracted at the highest level into three categories: sensors &amp; data, algorithms &amp; models, and middleware & resources. These components will be assembled and driven by an adaptive workflow engine and a web portal. Each component is described below.

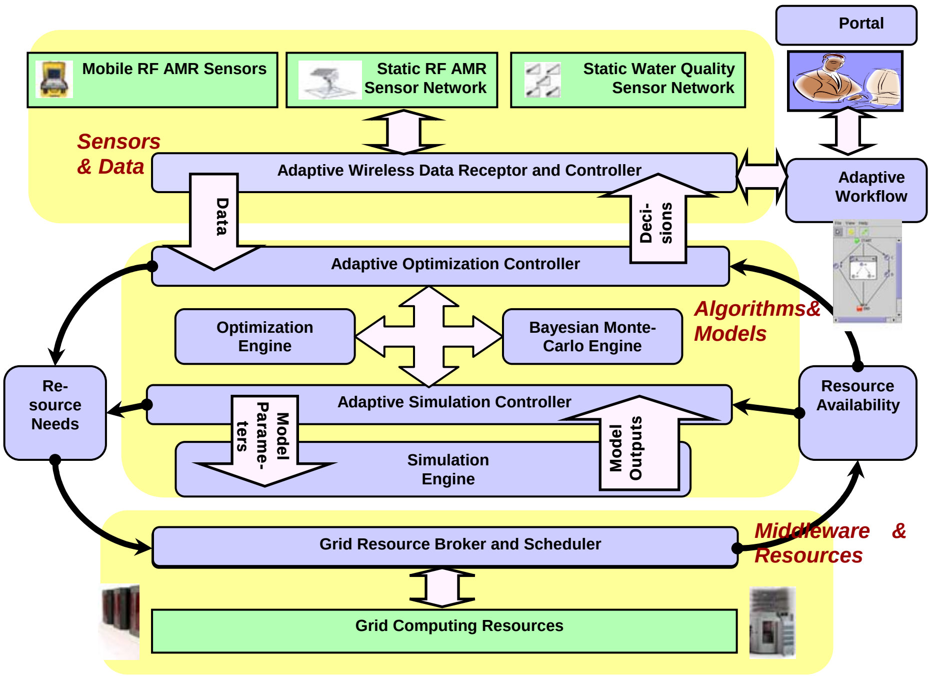

**Fig. 1.** Basic Architecture of the Cyberinfrastructure

### 3.1 Sensors & Data

Part of the infrastructure relies on the continuous ingestion of sensor data to drive the simulations. In case of an incident, it is planned to run several alternative instantiations in parallel and present the observing team with statistical augmentations in relationship to their accuracy. Our framework will furthermore determine areas of interest and suggestions for locating additional sensors. The sensor data is gathered from sensors supplied by the Neptune Technology Group (NTG). These second generation sensors can be integrated into a wireless RF (Radio Frequency) network. The data can either be read by drive-by readings or a communication station placed in the vicinity of a number of fixed transmitters. The sensors are one-way communication devices that transmit data using frequency hopping spread-spectrum technology enabling data security and reliability. The frequency for data reporting of the sensors is between 11 and 14 seconds. The fixed transmitters are configured to report 15 measurements per hour back to the utility center. The data gathered at the data center will be one way forwarded to a specialized and secured data store that is used by our software in order to avoid interference with the sensor network.

*Adaptive Data Receptor and Controller*

We will construct a stochastic network simulation and data communication engine to replace the physical radio frequency data communication layer. Working with NTG, their EZNET™ server software will be modified to communicate directly with a defined software API that simulates the physical behavior of installed RF meter interface units. At the heart of this new capability is simulation of the NTG E-Coder™ and ProRead™ encoder meter register, which has a physical interface with the customer water meter. In the simulation environment, these meter registers will be populated by network simulation models driven by stochastic models of individual customer water usage. The stochastic simulation models will be based on an existing framework for Monte-Carlo simulation of network hydraulics and water quality [3] and published Poisson-process stochastic water usage models (e.g.,[4]). The general testing environment will also simulate – as it would be configured in a real-time installation – contaminant introduction and the response of installed water quality sensors and data communications, thus providing a flexible testbed for evaluating the performance of our cyberinfrastructure.

### 3.2 Models & Algorithms

Our cyberinfrastructure is designed to tackle two kinds of problems: (i) contaminant source characterization problem, (ii) contaminant control problem. Immediate focus within this project will be on the source characterization problem which can be posed as parameter estimation or an inverse problem. The goal is to use the time-series of sensor data to recover the likely locations of the contaminant sources and their release histories that minimize the error between the predictions and the observations. The control problem, to be considered in subsequent projects, can be posed as a strategy optimization problem where the goal is to identify an optimal schedule for activating a set of control choices to meet threat management objectives. Choices for controlling the hydraulics in the network include pumping rates, valve adjustments, and flow releases from fire hydrants.

*Optimization Engine*

The optimization engine will consist of a suite of optimization procedures that will be used to search for the most likely source characterization to fit the measurements, calibrate the model in real time as new data are streamed-in, and to search for the most effective control strategy to manage network contamination. To facilitate use of these methods within the adaptive cyberinfrastructure, methods with varying degree of sophistication are needed to appropriately match the data and computing resource availability as well as the required accuracy at different stages of solution. The search methods will incorporate incoming data from sensors as they are streamed in real time to improve prediction efficiency. We are currently investigating an adaptive evolutionary algorithm for conducting search under dynamically varying systems [5]. The promising preliminary results indicate that this approach significantly improves prediction efficiency with dynamic infusion of data. An associated investigation of another search procedure shows promise in assessing the degree of non-uniqueness in solutions as more data are incorporated into the prediction process [6].

### *Bayesian/Monte-Carlo Engine*

The Bayesian/Monte Carlo engine will consist of procedures to identify adaptive and cost & time-efficient sampling plans (locations and frequencies) to reduce model-based prediction uncertainties and therefore improve confidence in control strategy performance. A computing resource-adaptive Bayesian data worth framework will be implemented. A data worth framework is a sequential process where in each iteration the best location to next sample is identified - sampling at that location will maximize the expected worth of the sample. The incorporation of Bayesian analysis allows the accounting, and updating with new sampling data, of uncertainties in parameters that are not directly observable, e.g., the true location and injection rates of contaminants in the water distribution network. Markov chain Monte Carlo (MCMC) methods for Bayesian analysis [7] will be used for computational efficiency. We will research and implement MCMC methods that can adapt to resource fluctuations in the grid computing environment: the number of parallel MCMC chains and tolerances for stopping criterion will be controlled. Approaches for reconciling the tradeoff between accuracy and run time will be explored. The accuracy in the identification of the best next sampling location will be balanced against the need to quickly zero in on the contaminant source characterization.

### *Simulation Engine*

EPANET [8], a widely used water distribution network hydraulic and water quality modeling tool, will be used as the simulation engine. This model uses known pipe network topology, link/node physical characteristics, and network boundary and initial conditions, to simulate the space-time variation of flows, pressures, and water quality concentrations using well-established principles [9]. The EPANET engine is available as a C language library with a well-defined API [10]. EPANET will be made malleable through adaptive control by the simulation controller. Malleability features to be incorporated are: (i) ability to change the problem configuration at any given time step to meet resource limitations or problem requirements, (ii) ability to change input conditions as new data become available, and (iii) ability to communicate with the controller at any given time. For (i), sensed information and the current solution state would be used to adaptively update the problem configuration. Implementing (ii) is straightforward as new sensor data will replace old values used in the simulation. For (iii), a communication feature using XML-based message passing will be built into EPANET so that it can readily communicate with controller in sending and receiving metadata augmented messages. The messages will include input data (e.g., decision variables) as well as action flags that can steer the simulation. The simulation engine will be coupled with an existing test suite of five real network models, ranging in size from several hundred to over 12,000 nodes. The test suite represents a range of hydraulic behaviors. Thus the test networks will adequately support development of algorithms, as well as challenging their performance and scalability.

### *Optimization and Simulation Controllers*

The optimization and Monte-Carlo controller will have the ability to (i) select appropriate optimization algorithms or Monte-Carlo simulation depending on the problem scenario, data, or computer resources, (ii) communicate with the optimization and Monte-Carlo engines, (iii) communicate with the adaptive data controller, (iv) communicate with the resource broker, and (v) communicate with the global workflow portal/controller.  For (i), a set of predefined rules based on solution time and problem scenario will be formulated for different algorithms. These rules will then be used in an adaptive manner to select the appropriate algorithm.

The simulation controller will have capabilities to (i) communicate with the global workflow controller, (ii) communicate with the optimization and Bayesian/Monte-Carlo engines, (iii) adaptively change model data, (iv) steer and track the model simulations, (iv) communicate with the resource broker. The global workflow controller will relay the baseline model scenario to the model controller. The optimization and Bayesian engines will relay the number of model evaluations to be completed, the decision variables, and any additional data from the sensors to the model controller. The resource broker will relay the available computer resources to the model controller. The model controller will utilize this information to determine the optimal model resolution and the resource requirements. It will then communicate with the resource broker and launch the appropriate simulations. During the simulations, the model controller may change the model data, launch additional simulations, or terminate the simulations depending on the need.

### 3.3 Middleware & Resources

The requirement to manage dynamic processes as part of our implementation strategy calls for a framework that assists the orchestration and coordination of these processes within a dynamically changing infrastructure. One part of such a framework is the design of a workflow management system [11,12] that provides interfaces among the sensor network, the scientific model calculations, the available Grid infrastructure, and the human participants. Such an integrated workflow system allows us to plan and organize the complex experiments in a dynamic changing environment. In contrast to other systems, we project an integrated approach that not only allows us to extend our system, but also allows us to integrate with other systems through the introduction of convenient abstractions. One of the open research issues in the current development of workflow systems is: How do we build a supporting workflow system that not only maps the workflow to an existing static or dynamic environment, but also modifies the workflow at runtime to adapt the workflow itself to the changing environment? This is particularly challenging as we foresee scenarios in which we can for example change the accuracy of calculations performed as part of the workflow while reducing the resource requirements in time and space.

The specification of such workflows must be assisted through a number of frameworks. We need workflow shells for rapid prototyping, a graphical editor to assemble the components by the application user, a programming framework to enhance the workflow management system, and an XML-based specification to assist in interfacing with the Web services community (a language such as XML is intended for automatic interpretation and is too cumbersome for humans to read). We will address these issues by expanding the Java CoG Kit [12] workflow system and explicitly focusing on the needs of this user community.

*Grid resource broker and scheduler*

Our use-cases include two scenarios, a traditional scenario in which we attempt to access a number of resources available to us for running a calculation, and another in which we attempt to predict a very accurate calculation in a case of an emergency. In the first case, we will use facilities of existing Grid middleware, such as Globus Toolkit [13], Condor glide-ins, and the Java CoG Kit to fulfill the users' need for accessing the widest range of resources. We also need to access, however, an advanced reservation system that allows preemptive scheduling based on priority needs in case of an emergency. To achieve this goal we plan to engage with the TeraGrid community in order to evaluate an on-demand emergency ticket system.

*Adaptive workflow portal*

One important issue is to enable simple access by the user community through a portal. We will reuse and expand upon portal technology developed as part of the NMI OGCE project [14]. We will enhance this workflow portal with the ability to browse through dynamically changing workflows. We focus on the development of components that foster ease of use, reusability, and computational steering within our framework. A workflow component repository that we have proposed recently [11] will be implemented to allow us to dynamically change components at runtime and develop automatic adaptive algorithms.

## 4 Conclusion

We have presented a novel infrastructure for dynamic data driven analysis for threat management in urban water distribution systems. We believe such an architecture can enable urban authorities to effectively manage water distribution contamination incidents by incorporating and controlling real-time feed of data from automatic meter readers and sensors.

## Acknowledgements

This work is supported by National Science Foundation (NSF) under Grant Nos. CMS-0540316, ANI-0540076, CMS-0540289, CMS-0540177, and NMI-0330545. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the NSF. The Java CogKit development is supported by NSF grant nos. ANI-0540076, NMI-0330545, and by the Dept of Energy under contract no. W-31-109-Eng-38.

## References

1. Water Science and Technology Board, A Review of the EPA Water Security Research and Technical Support Action Plan: Parts I and II, *The National Academies Press*, 2003.
2. Darema, F. Dynamic data driven applications systems: New capabilities for application simulations and measurements. *Lecture Notes in Computer Science 3515*, pages 610–615, 2005.
3. Murray, R., R. Janke, and J. Uber. The threat ensemble vulnerability assessment (TEVA) program for drinking water distribution system security. In Proceedings, *ASCE/EWRI World Water and Environmental Resources Congress*, Salt

Lake City, UT, 2004

4. Buchberger, S., and G. Wells, "Intensity, Duration, and Frequency of Residential Water Demands," *J. Water Resour. Plng. and Mgmt.* 122(11), 1996.
5. Liu, L. and S. R. Ranjithan, Adaptive Niched Co-Evolutionary Algorithm (ANCEA) for Dynamic Optimization, submitted to: *Genetic and Evolutionary Computation Conference (GECCO) 2006*, ACM.
6. Zechman, E. M. and S. R. Ranjithan, Niched Co-Evolution Strategies to Address Nonuniqueness in Engineering Design, submitted to: *Genetic and Evolutionary Computation Conference (GECCO) 2006*, ACM.
7. Gilks, W.R., S. Richardson, and D. Spiegelhalter. *Markov Chain Monte Carlo in Practice*, chapter Introducing Markov Chain Monte Carlo, pages 1–20. Chapman & Hall, London, 1996.
8. Rossman, L.A.., *EPANET Users Manual*. Risk Reduction Engrg. Lab., U.S. Evnir. Protection Agency, Cincinnati, Ohio, 1994.
9. Walski, T., Analysis of Water Distribution Systems, *Van Nostrand Reinhol*d, New York, 1984.
10. Rossman, L.A.. The epanet programmer's toolkit. In Proceedings of Water Resources Planning and Management Division Annual Specialty Conference, ASCE, Tempe, AZ, 1999.
11. von Laszewski, G. and Kodeboyina, D. A Repository Service for Grid Workflow Components. *International Conference on Autonomic and Autonomous System*s. IEEE, 23-28 October 2005. http://www.mcs.anl.gov/~gregor/papers/vonLaszewski-workflow-repository.pdf
12. von Laszewski, G. and Hategan, M. Java CoG Kit Workflow Concepts, in *Journal of Grid Computing*, Jan. 2006. http://www.mcs.anl.gov/~gregor/papers/vonLaszewski-workflow-jgc.pdf.
13. The Globus Project. Web Page. Available from: http://www.globus.org.
14. OGCE. Open Grid Computing Environments. Web Page. Available from: http://www.ogce.org.