

Toward Large-scale Situation Awareness Applications on Camera Networks

Kirak Hong
 College of Computing
 Georgia Institute of Technology
 hokira@gatech.edu

Abstract—Ubiquitous deployment of cameras and recent advances in video analytics enable a new class of applications, *situation awareness* using camera networks. The application class includes surveillance, traffic monitoring, and assisted living that autonomously generate actionable knowledge from a large number of camera streams. Despite technological advances, developing a large-scale situation awareness application still remains a challenge due to the programming complexity, highly dynamic workloads, and latency-sensitive quality of service. To solve the problem, my research topic concerns developing a programming model and a runtime system to support large-scale situation awareness applications on camera networks. The programming model requires a minimal set of domain-specific handlers from the domain experts, allowing them to focus on the algorithmic aspect of situation awareness applications rather than the details of distributed programming. The runtime system provides automatic resource management using smart cameras and the cloud for handling dynamic workloads and ensuring latency-sensitive quality of services.

Keywords—programming model; runtime system; situation awareness; camera network; cloud computing;

I. INTRODUCTION

Sensors of various modalities and capabilities, especially cameras, have become ubiquitous in our environment. Technological advances and the low cost of such sensors enable deployment of large-scale multi-modal camera networks in large metropolises such as London and New York. Various algorithms for analyzing and drawing inferences from video and audio have also matured tremendously in recent times. For example, face recognition and target tracking algorithms have been successfully demonstrated in many small-scale settings. Such technological advances have led to the development of a new class of applications called *situation awareness*. Situation Awareness is both a property and an application class that deals with recognizing when sensed data could lead to actionable knowledge [1]. Common examples in this class include intelligent surveillance, autonomous traffic monitoring, assisted living, and many more.

However, large-scale situation awareness applications using several thousand sensor streams are not yet common and are still challenging to develop. This is because such applications bring many challenges to the forefront: the general complexity of sensor-based distributed programming, the dynamism in terms of resource requirements and application

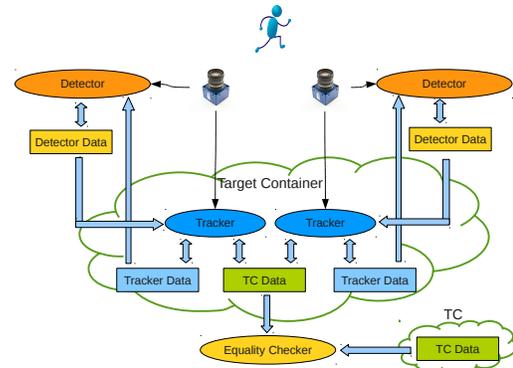


Figure 1. Surveillance Application using TC Model

context, and the need to meet the latency constraints in converting sensed data to knowledge.

To address these challenges, I take a two-pronged approach in my dissertation. First, I am exploring high-level programming abstractions that would allow domain experts to focus on application-specific analytics and not worry about the details of distributed programming. Second, to deal with the dynamic workloads while meeting the latency constraints of situation awareness applications, I am developing runtime systems that exploit the elasticity of the cloud and efficient event filtering at smart cameras. Detailed approaches for the programming models and runtime systems will be discussed in the following sections.

II. PROGRAMMING LARGE-SCALE SITUATION AWARENESS APPLICATIONS

One of the biggest challenges in developing large-scale situation awareness applications is the complexity of programming. Specifically, the developers of such applications are experts in their own domain (e.g., computer vision) and not necessarily well trained to deal with the myriads of details associated with sensor-based distributed programming such as synchronization, buffer management, and meeting real-time guarantees. To facilitate domain experts to write large-scale situation awareness applications, the right level of programming abstraction should be provided.

As a first step towards such a programming abstraction, I have developed together with colleagues from Rutgers University *Target Container* (TC) [2], a target-centric parallel programming abstraction for video-based surveillance. The

key insight of this programming abstraction is to allow the domain expert (e.g., a vision researcher) to only focus on the algorithmic details of target tracking and let the system deal with providing the computational resources (sensing, networking, and processing) to enable target tracking. Figure 1 shows the conceptual picture of how a surveillance application will be structured using the programming model. A TC corresponds to a target, possibly tracked from multiple cameras. The domain expert provides the code modules for target tracking as handlers to the TC system. The handlers are invoked dynamically by the TC system to discover new targets (detector), to follow existing targets (tracker) in camera streams, to check equality of existing targets (equality checker), and to merge the targets into the same TC if they are the same (merger). In essence, the TC programming model and the runtime system virtualize the sensor sources so far as the application is concerned. For example, when a target moves from one camera to another (Figure 1), the runtime system provides the new camera stream to the tracker without any application intervention.

Based on the previous version that was mainly designed for smart cameras, I am extending the handlers and API to support multi-modal sensors such as motion, light and RFID sensors. Supporting the multi-modal sensors allows situation awareness applications to perform more accurate analytics using various modalities of sensor data. Such a programming framework will be applicable to many application domains including physical security of office buildings, airports, and neighborhoods.

III. AUTOMATIC RESOURCE MANAGEMENT FOR LARGE-SCALE SITUATION AWARENESS

Another challenge in large-scale situation awareness applications is resource management for handling dynamic workloads and ensuring latency-sensitive quality of service. Imagine an airport security system that performs video analytics on live video streams to infer the identities of observed individuals. To ensure real-time processing of the camera streams, one could associate sufficient computational resources with each camera to deal with the worst-case demand for video analytics. However, this will be wasteful since some cameras may have more traffic (e.g., those near a security gate) than others, leading to gross under-utilization of the computational resources. Further, the overall resource requirement changes with time. During peak time when there is a significant amount of people traffic in the airport the resource requirement reaches a maximum, while it wanes during the night when there is not much traffic. Therefore, it is natural to use the elasticity of the cloud to deal with dynamic workloads. However, thus far cloud infrastructures have been designed primarily for throughput-oriented applications (e.g., MapReduce applications), not for applications with latency-sensitive quality of service and highly dynamic workloads. Exploring the right interfaces between the sens-

ing infrastructure and the cloud to meet such constraints is another aspect of my dissertation research.

Currently, I am investigating the effect of dynamic workloads on the quality of service using a specific situation awareness technique called *spatio-temporal analysis*. The technique infers each occupant's location at a given time by generating an event based on a captured biometric signature (e.g., a face image), and updating the application state based on the generated event. Applications using this technique involve highly dynamic workloads since the workloads of event generation and state update depend on the real world situation such as number of faces captured per second. Plus, the quality of service of the applications are specified as end-to-end latency from signature detection to state update. I have developed a distributed framework [3] for providing automatic resource management for the spatio-temporal analysis technique with colleagues from SUNY-Buffalo who have developed algorithms for the technique [4]. The key objective of the framework is to ensure the user-provided end-to-end latency using elastic computing resources in the cloud while minimizing the performance degradation during sudden changes of workloads.

IV. CONCLUSION

With the ubiquity of cameras and the technological advances in video analytics, it is clear that the demand for large-scale situation awareness applications on camera networks will be commonplace in the near future. Thus my research focus is finding the right distributed programming model and the runtime system for supporting such applications.

ACKNOWLEDGMENT

The work documented in this extended abstract is joint with my advisor Professor Umakishore Ramachandran, and colleagues from Rutgers University and SUNY-Buffalo.

REFERENCES

- [1] U. Ramachandran, K. Hong, L. Iftode, R. Jain, R. Kumar, K. Rothermel, J. Shin, and R. Sivakumar, "Large-scale situation awareness with camera networks and multimodal sensing," *Proceedings of the IEEE*, vol. 100, no. 4, pp. 878–892, april 2012.
- [2] K. Hong, S. Smaldone, J. Shin, D. J. Lillethun, L. Iftode, and U. Ramachandran, "Target container: A target-centric parallel programming abstraction for video-based surveillance," in *ICDSC*, 2011, pp. 1–8.
- [3] K. Hong, M. Voelz, V. Govindaraju, B. Jayaraman, and U. Ramachandran, "A distributed framework for spatio-temporal analysis on large-scale camera networks," Georgia Institute of Technology, Tech. Rep. GT-CS-12-10.
- [4] V. Menon, B. Jayaraman, and V. Govindaraju, "The Three Rs of Cyberphysical Spaces," *Computer*, vol. 44, pp. 73–79, 2011.