Science of Computer Programming 98 (2015) 159–183

Contents lists available at ScienceDirect

Science of Computer Programming

www.elsevier.com/locate/scico

Parallel gesture recognition with soft real-time guarantees ☆

Stefan Marr *, Thierry Renaux, Lode Hoste, Wolfgang De Meuter

Software Languages Lab, Vrije Universiteit Brussel, Belgium

HIGHLIGHTS

- First parallel soft real-time Rete rule engine.
- Designed for gesture recognition and complex event processing.
- Scalability evaluated for up to 64 cores.

ARTICLE INFO

Article history:
Received 7 March 2013
Received in revised form 23 January 2014
Accepted 5 February 2014
Available online 21 February 2014

Keywords:
Gesture recognition
Rete
Soft real-time guarantees
Complex event processing
Multicore

ABSTRACT

Using imperative programming to process event streams, such as those generated by multi-touch devices and 3D cameras, has significant engineering drawbacks. Declarative approaches solve common problems but so far, they have not been able to scale on multicore systems while providing guaranteed response times.

We propose PARTE, a parallel scalable complex event processing engine that allows for a declarative definition of event patterns and provides soft real-time guarantees for their recognition. The proposed approach extends the classical Rete algorithm and maps event matching onto a graph of actor nodes. Using a tiered event matching model, PARTE provides upper bounds on the detection latency by relying on a combination of non-blocking message passing between Rete nodes and safe memory management techniques.

The performance evaluation shows the scalability of our approach on up to 64 cores. Moreover, it indicates that PARTE's design choices lead to more predictable performance compared to a PARTE variant without soft real-time guarantees. Finally, the evaluation indicates further that gesture recognition can benefit from the exposed parallelism with superlinear speedups.

1. Introduction

In order to improve how users and computers interact, multi-touch input, gesture recognition, and speech processing are becoming common in consumer hardware. To power more natural user interfaces, primitive sensor readings from multiple input devices need to be correlated to form higher-level events. A wide range of applications has been proposed to utilize such sensors by extracting meaningful information from the raw data, commonly called gesture recognition. Examples

☆ This paper is an extension of: Renaux, T., Hoste, L., Marr, S. and De Meuter, W. (2012), Parallel gesture recognition with soft real-time guarantees, in: Proceedings of the 2nd edition on Programming Systems, Languages and Applications Based on Actors, Agents, and Decentralized Control Abstractions, pp. 35–46.

* Corresponding author.
E-mail addresses: smarr@vub.ac.be (S. Marr), trenaux@vub.ac.be (T. Renaux), lhoste@vub.ac.be (L. Hoste), wdmeuter@vub.ac.be (W. De Meuter).

http://dx.doi.org/10.1016/j.scico.2014.02.012
0167-6423/© 2014 Elsevier B.V. All rights reserved.

include discovering when a phone is dropped, detecting whether the user is "throwing" data to another device,[1] and performing multi-touch gestures to quickly access information.[2]

In such multimodal systems with many possible interactions, the required computational power easily outgrows what today's processors provide in terms of sequential performance. This is problematic for real-time server-sided pattern recognition, for instance to process surveillance camera-input, as well as for embedded devices and mobile phones with various sensors such as an accelerometer, gyroscope, multi-touch, proximity-sensor, and a microphone. Fusing these primitive and higher-order events easily becomes excessive for a single processing unit, such that utilizing the steadily rising degree of available parallelism becomes a necessity to provide the required degree of real-time interactivity to the users.

Another common problem is that imperative programming languages are considered to be cumbersome, error-prone, and lacking flexibility [1,2] when complex event correlations need to be implemented. General purpose imperative languages do not provide the necessary abstraction to help the programmer to express event patterns conveniently. Hammond and Davis [3], Scholliers et al. [4], and Hoste et al. [1] demonstrate that declarative definitions for sketch recognition, multi-touch gestures, or multimodal correlation provide the necessary language constructs and improve over imperative approaches by providing better software engineering abstractions.

Unfortunately, automatized recognition with machine learning techniques has drawbacks as well. On the one hand, it requires large sets of training data to build a statistical model of a gesture. Gathering such data and annotating it for the training process can be prohibitively time intensive. On the other hand, machine learning techniques are typically black-box processes that do not give developers the necessary information to debug and change the way gestures are recognized directly. Consequently, declarative approaches are a better choice today.

Declarative approaches are based on inference engines that process incoming sensor events using the declarative rules that describe the gestures. The Rete algorithm [5] is one possible foundation for such an inference engine. It is a forward-chaining, state-saving algorithm that is used to build rule-based expert systems. Declarative gesture approaches benefit from it because the state-saving approach optimizes the execution of rules so that only the program part relevant to an incoming event is executed, which minimizes the necessary computation that has to be performed whenever a new event takes place. As such, it reduces the computational overhead of continuous pattern matching enabling the use of a wide range of input sources and complex patterns without causing unacceptable performance overheads.

We present here a variation of the Rete algorithm called PARTE, built on a graph represented by a set of actors, providing both scalability and responsiveness. The contributions of our work are:

PARTE: design and implementation techniques tailored towards parallel recognition of user interaction patterns with soft real-time[3] guarantees.

Validation of PARTE's real-time guarantees by analyzing the execution properties of the implemented algorithm, which ensures freedom of unbounded loops and uses only non-blocking concurrent interactions. Furthermore, an empirical evaluation assesses the soft real-time properties of PARTE and compares it with a non-real-time variation.

Validation of PARTE's practicality by demonstrating the scalability of the parallel implementation on up to 64 cores and demonstrating that the sequential overhead compared to CLIPS,[4] a highly optimized sequential Rete implementation, can be overcome in the parallel case.

The remainder of this paper is structured as follows: first, in Section 2 we provide a detailed discussion of the context of multimodal input systems, their requirements and constraints, and the assumptions we can make. Section 3 describes the design principles, implementation techniques, and the parallel Rete algorithm of PARTE in detail. Section 4 evaluates PARTE in the context of gesture recognition and characterizes its execution semantics both with respect to non-blocking behavior and with respect to unbounded loops. PARTE's performance is evaluated in Section 5. Finally, we contrast our approach with the related work in Section 6 and summarize our conclusions and future work in Section 7. The used benchmarks are characterized in Appendix A as a foundation for the evaluation.

2. Context and requirements

The domain of gesture recognition comes with a set of properties that is different from many domains in which inference engines are commonly used. Since we rely on these particularities of the problem domain in the design of PARTE, we sketch the domain briefly and distill a list of requirements for inference engines in this domain.

[1] *Hoccer, exchanging data using gestures*, Art+Com Technologies, access date: July 8, 2013, http://www.youtube.com/watch?v=eqv8Q6M1O6Y.

[2] *Gesture Search for Android*, Google Inc., access date: July 8, 2013, http://www.google.com/mobile/gesture-search/.

[3] In *soft* real-time systems, the usefulness of results degrades past their deadline, whereas in *hard* real-time systems the usefulness drops to zero on a missed deadline. Hence, delays in a soft real-time system undermine the system's quality of service, while delays in hard real-time systems undermine the system's correctness.

[4] *CLIPS: A Tool for Building Expert Systems*, Gary Riley, access date: January 25, 2013, http://clipsrules.sourceforge.net/.