Parallel Gesture Recognition with Soft Real-Time Guarantees

Thierry Renaux
Software Languages Lab
Vrije Universiteit Brussel, Belgium
trenaux@vub.ac.be

Abstract

When dealing with the complex task of extracting meaningful information from multiple continuous sensor streams, declarative rules can be employed to benefit from software engineering principles such as modularization and composition. We propose PARTE, a parallel scalable event processing engine proving predictable response times for a high-quality user experience.

Categories and Subject Descriptors D.1.3 [*Programming Techniques*]: Concurrent programming; D.3.4 [*Programming Techniques*]: Processors; I.5.5 [*Pattern Recognition*]: Implementation

General Terms Algorithms, Design, Performance

Keywords complex event processing, gesture recognition, actors, multimodal interaction, soft real-time guarantees, Rete, nonblocking

1. Introduction

To improve the quality of interactions between users and computers, interest in multi-touch input, gesture recognition, and speech processing on consumer hardware has recently emerged. To power natural user interfaces, primitive sensor readings, which are collected by devices for multimodal input, need to be correlated to create higher-level events. Since hard-coding these complex correlations in imperative programming languages is cumbersome, error-prone, and lacks flexibility, declarative techniques are the preferred solution. Multi-touch interaction frameworks such as Midas [6] and its multimodal successor Mudra [4], use an inference engine, which compares the events based on sensor readings with declarative rules describing the gestures.

SPLASH'12, October 19–26, 2012, Tucson, Arizona, USA.
ACM 978-1-4503-1563-0/12/10.

2. Problem

To provide a high-quality user experience, the inference engine has to correlate events in a timely manner. When a user for instance interacts with a system through a multi-touch interface, changes should be reflected immediately and with a predictable delay to give the user a natural feedback. The same is true for multimodal interaction: When a user gives a series of speech and gesture commands, the right action should be performed without random delays that confuse the user about whether the command has been accepted or not.

Systems such as Midas embed inference engines which only tap the computational power of a single processing unit. However, the rise in sequential processing power offered by single processing units is stagnating, because efforts to increase clock-speed, instruction-pipeline depth, etc. offer diminishing returns. This severely limits the possible number of rules, their complexity, and the rate of events the system can handle. The only way to recognize more complex user interaction patterns without undermining the user experience by increased delays, is to embrace parallel processing power.

In addition to recognizing patterns in a timely manner, the system also needs to guarantee predictable responds times. This ensures that the system feels interactive and responsive. Akscyn et al. [1] show that long delays in interactive systems can distract users, and cause them to stop using the system. Consider for instance a user of a multi-touch gesture recognition system, who taps a certain location. If the user interface does not reflect this change within the timeframe the user has grown to expect, he will assume the command was not received, and may tap again. When the system then finishes processing the overdue gestures, the action will be executed twice. Users will rightfully blame the gesture detection system for this mistake. To prevent such errors, the detection of complex user interaction patterns should happen within a timeframe that can be predicted reliably up front.

However, the requirements of responsiveness and predictable runtime conflict: To offer the best performance on current hardware, the rule engine needs to use the available parallelism, and should provide real-time guarantees to ensure responsiveness. Current rule engines do not combine both requirements. They either are single-threaded in nature, or do not guarantee predictable worst-case execution times.

3. Approach

We propose the Parallel Actor-based ReTe Engine (PARTE), a complex event detection system, which address both the efficiency and the real-time requirement. The Rete algorithm [2] provides the desired efficiency, minimizing the asymptotic complexity.

The declarative rules defining gestures are transformed into a directed acyclic Rete network. In PARTE, each node of the Rete network corresponds to an actor. This enables both pipeline parallelism, as multiple stages of the Rete network can be processed simultaneously, and branch parallelism, as multiple branches in the Rete network can be processed simultaneously. Synchronization between the nodes happens in the form of inter-actor message-passing, where nodes only send messages to their successors in the Rete graph. The structure of the Rete graph in combination with the temporal properties of the events are exploited to guarantee synchronization between the branches, and with that, correctness. The actors' inboxes are represented by nonblocking queues using Michael's Safe Memory Reclamation technique [5].

By using nonblocking queues as the only concurrently accessed shared data structures, and by formalizing the execution steps, PARTE allows the reliable prediction of an upper bound on the execution time of the pattern matching. Our system is designed in such a way that every stage of the processing requires a bounded amount of time, which can be predicted given a set of gesture definitions and an upper bound on the rate of incoming events. Similarly, the number of stages required to process a series of events is bounded and constant for a given set of gesture definitions. To achieve this, we impose the requirement on rulesets that they must be *tiered*, a property that ensures that the rulesets do not express cycles. This approach enables us to guarantee soft real-time detection of gestures, given a set of rules defining gestures and an upper bound on the rate of incoming events.

Our preliminary performance evaluation used a number of microbenchmarks and a set of gesture definitions. The results showed that PARTE in its current unoptimized state is on average about three times slower than the highly optimized implementation of CLIPS [3] for the single threaded case. In the multithreaded case, PARTE was able to scale linearly up to 16 cores on our test machine, as such outperforming the sequential implementation.

4. Contributions and Future Work

The contributions of our work are:

Design and implementation of a parallel Rete engine tailored towards recognition of user interaction patterns with soft real-time guarantees.

Validation of real-time guarantees by evaluating the execution properties of the implemented algorithm, ensuring freedom of unbounded loops, and of blocking concurrent interactions.

Validation of practicality by showing the scalability of the parallel implementation and demonstrating that the overhead compared to CLIPS, a highly optimized sequential implementation, is acceptable.

Some limitations exist in the current version of PARTE. First, since test-expressions are separated from the nodes that join two branches, the full expressiveness of negation-as-failure is not supported in PARTE. Gesture recognition systems can work without a notion of negation, but the addition of it would be worthwhile nevertheless. Second, the coarseness of parallelization can further be tuned, as in some situations, the actor-based approach does not expose all options for parallelism. Inversely, in some situations, the actor-based approach imposes too high an overhead compared to the useful work. Finally, PARTE's current scheduler for the actors focusses only on correctness and real-time properties, but does not do any effort to prevent scheduling actors who do not have any message waiting in their inbox.

PARTE is compatible with existing single threaded inference engines such as CLIPS from NASA [3]. It can be used as a replacement of the core infrastructure of the multimodal Mudra framework [6], but can also be applied in a broader context where responsive and predictable complex event processing is needed. It can for instance be embedded in software monitoring network security, in algorithmic stock trading systems, etc. To conclude, PARTE is a parallel, actor-based variant for the Rete algorithm, offering soft real-time guarantees on the time it requires to process information from various sensor streams.

References

[1] R. M. Akscyn, D. L. McCracken, and E. A. Yoder. KMS: a distributed hypermedia system for managing knowledge in organizations. *Communications of the ACM*, 31(7):820–835, July 1988. ISSN 00010782.

[2] C. L. Forgy. Rete: A fast algorithm for the many pattern/many object pattern match problem. *Artificial Intelligence*, 19(1):17–37, Sept. 1982. ISSN 00043702.

[3] J. Giarratano. *CLIPS User's Guide, Version 6.0*. NASA - Lyndon B. Johnson Space Center, 1993.

[4] L. Hoste, B. Dumas, and B. Signer. Mudra: A Unified Multimodal Interaction Framework. In *Proceedings of ICMI 2011, 13th International Conference on Multimodal Interaction*, Alicante, Spain, November 2011.

[5] M. M. Michael. Safe memory reclamation for dynamic lock-free objects using atomic reads and writes. In *Proceedings of the twenty-first annual symposium on Principles of distributed computing - PODC '02*, page 21, New York, New York, USA, 2002. ACM Press. ISBN 1581134851.

[6] C. Scholliers, L. Hoste, B. Signer, and W. D. Meuter. Midas: A Declarative Multi-Touch Interaction Framework. In *Proceedings of TEI 2011, 5th International Conference on Tangible, Embedded, and Embodied Interaction*, Funchal, Portugal, Jan 2011.