PUC–Rio

Plano de Trabalho

Uma Linguagem de Programação Reativa para Redes de Sensores sem Fio

Autor:
Francisco Sant'Anna

Orientadores:
Roberto Ierusalimschy
Noemi Rodriguez

24 de Maio de 2012

Conteúdo

1 Resumo

Este documento descreve o plano de trabalho para o doutorado sanduíche na Chalmers University (Gotemburgo, Suécia) com duração de seis meses pelo programa de intercâmbio "Ciência sem Fronteiras". O proponente, Francisco Figueiredo Goytacaz Sant'Anna, é aluno de doutorado de ciência da computação da PUC–Rio (CAPES 7).

O plano de trabalho proposto é entitulado "Uma Linguagem de Programação Reativa para Redes de Sensores sem Fio" e descreve como a linguagem de programação sendo desenvolvida pelo proponente pode ser usada no contexto de Redes de Sensores sem Fio.

A pesquisa proposta está no escopo de Ciência da Computação e Sistemas de Informação, que são áreas prioritárias do programa de intercâmbio "Ciência sem Fronteiras".

2 Introdução

As Redes de Sensores sem Fio (RSSF) são compostas por um grande número de pequenos dispositivos (conhecidos como *motes*) capazes de sentir o ambiente e se comunicarem mutuamente. RSSF são usualmente aplicadas no monitoramento contínuo de fenômenos físicos em grandes áreas (possivelmente inacessíveis), tais como incêndios em florestas e deformações em grandes construções. Cada mote possui processamento limitado, comunicação via rádio de curto alcance e um ou mais sensores físicos. [1]

Software para RSSF é tipicamente desenvolvido na linguagem de programação C, e a adição de um sistema operacional de tempo real pode estender a linguagem com suporte a *multithreading* cooperativo ou preemptivo [6, 9]. No entanto, o uso de concorrência em C requer um exercício de muito baixo nível relacionado ao escalonamento, sincronização, e ciclo de vida das atividades (i.e. criação e destruição de *threads*).

Concorrência em C também não dispõe de garantias de segurança, uma vez que os programas são suscetíveis a condições de corrida, *deadlocks*, e execução por tempo indeterminado. No entanto, segurança é um aspecto importante em RSSF, já que os motes dispõem de recursos escassos, são implantados em lugares inacessíveis e devem funcionar por longos períodos sem a intervenção humana.

A linguagem de programação "Céu"[1] está sendo desenvolvida como parte do doutorado do proponente e incorpora características encontradas em linguagens reativas imperativas e de *dataflow* [8, 3], tendo como principal alvo

[1] `http://www.ceu-lang.org/`

os sistemas embarcados de recursos limitados, tais como os encontrados em RSSF.

Céu provê suporte para linhas de execução concorrentes que executam sincronamente, passo a passo, e podem compartilhar variáveis. Mesmo assim, a natureza síncrona e estática de Céu permite uma análise em tempo de compilação que garante que os programas executem de maneira determinística. Dessa maneira, Céu oferece um alternativa segura e de mais alto nível que os sistemas predominantes baseados em C. O compilador de Céu gera código comparável a programas escritos manualmente em C em termos de tamanho e portabilidade. No final do ano passado apresentamos um artigo resumido sobre Céu em um colóquio de doutorandos na conferência SenSys [12].

A nossa pesquisa na PUC–Rio com Céu é parte de um projeto maior chamado "Cidades Inteligentes", que envolve 20 universidades brasileiras, um investimento da ordem de 1 milhão de dólares, e que busca construir uma infra-estrutura de instrumentação, computação e comunicação para Redes de Sensores sem Fio.

3 Objetivos

O principal objetivo da pesquisa durante o período do doutorado sanduíche na universidade de Chalmers é avaliar a aplicabilidade de Céu no contexto de Redes de Sensores sem Fio. A avaliação envolve medidas quantitativas (e.g. uso de memória e bateria) e também qualitativas (e.g. facilidade de programação).

O grupo "Distributed Computing and Systems (DCS)" da universidade de Chalmers está oferecendo uma oportunidade de doutorado sanduíche na área de "Secure Wireless Sensor Networks". A vaga descreve tecnologias de linguagens de programação como um dos focos:

"The Security group has developed the link between two areas of computer science: programming languages and computer security. The group explores security models and enforcement mechanisms based on programming-language technology".

Acreditamos que RSSF sejam um cenário ideal para aplicar a linguagem Céu, uma vez que esta é direcionada a sistemas embarcados com poucos recursos, oferecendo concorrência de baixa granularidade, baixo consumo de memória e garantias de segurança.

Um grande número de serviços de software é necessário para construir aplicações de RSSF. A técnica de compor serviços em cima de outros serviços e, ao mesmo tempo, separá-los em unidades tratáveis individualmente

é uma abordagem poderosa que a linguagem Céu fomenta através de suas composições sequenciais e em paralelo.

Como um exemplo ilustrativo, as aplicações de RSSF tipicamente executam versões customizadas de algoritmos distribuídos clássicos, tais como descobrimento de topologia, eleição de líder e *broadcast* de mensagens [2]. Esses algoritmos provêem suporte para a construção de aplicações mais complexas e devem executar continuamente em paralelo com a aplicação na qual estão contidos, uma vez que RSSF são suscetíveis a falhas intermitentes.

O grupo de RSSF da PUC–Rio está desenvolvendo uma biblioteca de algoritmos distribuídos em Céu. Dessa maneira, os algoritmos podem ser facilmente compostos em paralelo com as aplicações finais. A linguagem detecta qualquer inconsistência nas composições, tais como acessos concorrentes às variáveis compartilhadas e também códigos inalcançáveis. A mesma idéia pode ser estendida para os algoritmos usados e/ou desenvolvidos pelo grupo DCS de Chalmers (e.g. [11]).

Também estamos investigando formas de descrever uma abordagem para sintetizar representações de dados para programas concorrentes. É parte do processo evoluir Céu com o grupo DCS conforme forem sendo identificados novos requisitos não endereçados pela linguagem.

O grupo DCS também desenvolve pesquisa em diversos tópicos de linguagens de programação que também podem ser explorados durante o doutorado sanduíche [13, 4, 7]. O grupo possui uma biblioteca de construções concorrentes (NOBLE[2]) e uma plataforma de testes para aplicações paralelas e concorrentes (GULLIVER[3]).

A lista a seguir enumera outros tópicos de pesquisa alinhados com o modelo de programação de Céu que foi retirado da homepage do DCS[4]:

- Investigar novas técnicas para atingir paralelismo e tolerância a falhas em sistemas paralelos e distribuídos.
- Avaliar a performance de técnicas não bloqueantes para sincronização de sistemas concorrentes.
- Aumentar a performance através de sistemas de escalonamento e sincronização cooperativo.

[2] `http://www.cse.chalmers.se/research/group/noble/`
[3] `http://www.chalmers.se/hosted/gulliver-en/`
[4] `http://www.cse.chalmers.se/research/group/dcs/`

4 Metodologia

Durante o período do doutorado sanduíche, nossa intenção é desenvolver aplicações de RSSF em Céu com alto grau de complexidade para avaliar a aplicabilidade da linguagem como uma alternativa às tecnologias atualmente empregadas. Iremos utilizar métricas quantitativas e qualitativas para os aspectos relevantes em RSSF, comparando implementações de aplicações em Céu e em outras linguagens (e.g. C).

Um trabalho existente específico para RSSF [5] mede a performance de diferentes linguagens de programação com respeito ao consumo de memória, energia e responsividade de aplicações. Esses aspectos são de extrema importância, uma vez que, nesse contexto, os recursos de memória, bateria e processador são bastante limitados.

Também consideramos segurança como um aspecto essencial, dado que os motes devem executar por longos períodos sem a intervenção humana. Finalmente, um alto grau de expressividade também é desejado em qualquer linguagem de programação.

Segue a lista de aspectos que iremos avaliar durante a nossa pesquisa:

- Consumo de memória: quanto as aplicações consomem em termos de memória volátil (RAM) e não volátil (ROM).
- Consumo de bateria: quanto de energia as aplicações consomem.
- Responsividade: com que velocidade as aplicações respondem a requisições de alta prioridade (e.g. mensagens de rádio).
- Segurança: que garantias de segurança a linguagem oferece automaticamente (e.g. acesso seguro a varáveis compartilhadas).
- Expressividade: com que facilidade as aplicações podem ser desenvolvidas e mantidas.

Os primeiros três aspectos podem ser facilmente avaliados com medidas quantitativas [5], enquanto que os dois últimos requerem uma análise qualitativa mais cuidadosa.

Inicialmente, iremos portar para Céu um conjunto de aplicações desenvolvidas em outras linguagens a fim de comparar os aspectos quantitativos. Esses resultados irão servir como feedback imediato para otimizar o código final gerado pela linguagem.

Em seguida, iremos focar em aplicações novas de maior complexidade, de modo a avaliar os aspectos qualitativos. Ainda assim, poderemos usar

adicionalmente medidas quantitativas, tais como o número total de linhas de código escritas como indicativo para expressividade.

Na linha qualitativa, a análise estática de Céu em programas de maior complexidade poderá trazer um grande ganho de segurança para os programas gerados, uma vez que ela irá detectar automaticamente acessos concorrentes em partes que o programador não espera que haja conflitos. Além disso, o uso recorrente de códigos pré-existentes através das composições de Céu poderá se mostrar um facilitador para o desenvolvimento de aplicações complexas.

5 Cronograma

O plano de trabalho define um cronograma de seis meses a ser seguido durante o período do doutorado sanduíche:

	M1	M2	M3	M4	M5	M6
Apresentações de Céu	X		X		X	
Desenvolvimento de aplicações existentes	X	X				
Avanços e melhorias em Céu		x	X	x	X	
Desenvolvimento de novas aplicações			X	X	X	
Elaboração de um Artigo					X	X
Outras atividades	x	x	x	x	x	x

(células com **X** indicam muita atividade / células com **x** indicam pouca atividade)

Meses 1–2: Durante o primeiro mês, iremos apresentar a linguagem Céu ao grupo de pesquisa, focando em sua aplicabilidade a RSSF e em como ela difere de sistemas existentes. Em seguida, com o suporte do grupo, iremos escolher aplicações de RSSF existentes para serem portadas para Céu, a fim de avaliar os aspectos quantitativos nas diferentes implementações.

Meses 3–4: Após o trabalho em aplicações existentes, iremos coletar os resultados para serem usados em melhorias na linguagem. Também iremos apresentar os resultados ao grupo e discutir o desenvolvimento de novas aplicações com maior complexidade para os meses seguintes.

Meses 5–6: Ao redor do quinto mês, esperamos ter desenvolvido aplicações mais complexas que possibilitem uma análise mais qualitativa da linguagem Céu, conforme discutido na Seção 4.

Tanto a análise quantitativa, quanto a análise qualitativa serão usadas na elaboração de um artigo a ser submetido a uma conferência de RSSF.

Meses 1–6: Também existe o interesse em explorar outras áreas de pesquisa do grupo DCS relacionadas a linguagens de programação e sistemas distribuídos, conforme discutido na Seção 3. Esse trabalho pode ser desempenhado ao longo de todo o período do sanduíche, com menor intensidade.

6 Resultados Esperados

Ao final do período do doutorado sanduíche, esperamos que Céu tenha se tornado uma alternativa real para o desenvolvimento de aplicações complexas para Redes de Sensores sem Fio.

Como a nossa metodologia envolve inicialmente uma análise quantitativa facilmente mensurável, conseguiremos um feedback quase que imediato para evoluir a linguagem em pouco tempo. Ao fim do processo e com o desenvolvimento de aplicações mais complexas, teremos um maior embasamento para defender a aplicabilidade de Céu (ou não aplicabilidade) no contexto de RSSF.

Os experimentos iniciais já mostram um resultado competitivo de Céu em termos de uso de memória e responsividade de aplicações [12]. Com um uso mais difundido da linguagem durante o doutorado sanduíche, disponibilizaremos de mais recursos para estendermos os experimentos.

Também esperamos que o grupo de RSSF da PUC–Rio tire proveito do intercâmbio com o grupo do Chalmers e vice-e-versa. Os grupos compartilham os mesmos interesses acadêmicos e uma relação contínua pode ser iniciada com essa primeira experiência. Como exemplo de colaboração, a biblioteca de algoritmos distribuídos para RSSF sendo desenvolvida pelo grupo da PUC–Rio pode ser mais facilmente adotada pelo grupo de Chalmers através do intercâmbio a ser promovido.

Por fim, pretendemos escrever um artigo completo para uma conferência de RSSF em conjunto com o grupo de Chalmers descrevendo os resultados de Céu durante o período do sanduíche. Dentre as possíveis conferências, destacamos duas:

- ACM/IEEE Conference on Information Processing in Sensor Networks (IPSN). Submission deadline expected to October 2012.
- ACM Conference on Embedded Networked Sensor Systems (SenSys). Submission deadline expected to April 2013.

7 Pesquisadores Envolvidos

PUC–Rio (proponentes):

Francisco Sant'Anna possui graduação em Engenharia de Computação (2003) e mestrado em Informática (2009) pela PUC–Rio. Atualmente é aluno bolsista CNPq do terceiro ano de doutorado do Departamento de Informática da PUC–Rio. O título de sua tese é "Céu: Embedded, Safe, and Reactive Programming", que deve ser concluída em setembro de 2013. Seus orientadores são o Prof. Roberto Ierusalimschy e a Prof. Noemi Rodriguez, que atuam, respectivamente, na área de linguagens de programação e sistemas distribuídos.

Homepage: `http://www.lua.inf.puc-rio.br/~francisco/`

Lattes: `http://lattes.cnpq.br/0077491494754494`

Roberto Ierusalimschy é professor associado do Departamento de Informática da PUC-Rio e bolsista de produtividade do CNPq (nível 1C). Roberto é também o principal projetista da linguagem de programação Lua e autor do livro Programming in Lua [10]. Suas atividades de pesquisa têm como foco linguagens de programação, principalmente linguagens de script e de domínio específico.

Homepage: `http://www.inf.puc-rio.br/~roberto/`

Lattes: `http://lattes.cnpq.br/0427692772445368`

Noemi Rodriguez é professora associada do Departamento de Informática da PUC–Rio e bolsista de produtividade do CNPq (nível 2). Seus interesses se concentram em programação distribuída e concorrente, com ênfase no papel que linguagens de programação têm a desempenhar nessa área. Noemi também coordena o grupo de Redes de Sensores sem Fio da PUC–Rio.

Homepage: `http://www.inf.puc-rio.br/~noemi/`

Lattes: `http://lattes.cnpq.br/4933326132948063`

Chalmers University (destino):

Philippas Tsigas é professor do "Department of Computing Science and Engineering" na universidade de Chalmers, onde coordena o laboratório "Distributed Computing and Systems Research Group". Seus interesses de pesquisa se concentram na área de sistemas paralelos e distribuídos.

Homepage: `http://www.cse.chalmers.se/~tsigas/index.html`

Referências

[1] Akyildiz, I. F., Su, W., Sankarasubramaniam, Y., and Cayirci, E. Wireless sensor networks: a survey. *Computer Networks 38*, 4 (2002), 393–422.

[2] Andrews, G. R. Paradigms for process interaction in distributed programs. *ACM Comput. Surv. 23*, 1 (Mar. 1991), 49–90.

[3] Berry, G., and Gonthier, G. The ESTEREL synchronous programming language: design, semantics, implementation. *Science of Computer Programming 19*, 2 (1992), 87–152.

[4] Cederman, D., Tsigas, P., and Chaudhry, M. T. Towards a software transactional memory for graphics processors. In *EGPGV* (2010), pp. 121–129.

[5] Duffy, C., Roedig, U., Herbert, J., and Sreenan, C. J. A comprehensive experimental comparison of event driven and multi-threaded sensor node operating systems. *JNW 3*, 3 (2008), 57–70.

[6] Dunkels, A., Gronvall, B., and Voigt, T. Contiki - a lightweight and flexible operating system for tiny networked sensors. In *Proceedings of the 29th Annual IEEE International Conference on Local Computer Networks* (Washington, DC, USA, 2004), LCN '04, IEEE Computer Society, pp. 455–462.

[7] Ha, P. H., Tsigas, P., and Anshus, O. J. Nb-feb: A universal scalable easy-to-use synchronization primitive for manycore architectures. In *OPODIS* (2009), pp. 189–203.

[8] Halbwachs, N., Caspi, P., Raymond, P., and Pilaud, D. The synchronous data-flow programming language LUSTRE. *Proceedings of the IEEE 79* (September 1991), 1305–1320.

[9] Hill, J., Szewczyk, R., Woo, A., Hollar, S., Culler, D., and Pister, K. System architecture directions for networked sensors. *SIGPLAN Notices 35* (November 2000), 93–104.

[10] Ierusalimschy, R. *Programming in Lua, Second Edition*. Lua.Org, 2006.

[11] Larsson, A., and Tsigas, P. A self-stabilizing (k, r)-clustering algorithm with multiple paths for wireless ad-hoc networks. In *ICDCS* (2011), pp. 353–362.

[12] Sant'Anna, F. Céu: A reactive language for wireless sensor networks. http://www.cse.ust.hk/~lingu/SenSys11DC/, 2011.

[13] Sundell, H., Gidenstam, A., Papatriantafilou, M., and Tsigas, P. A lock-free algorithm for concurrent bags. In *SPAA* (2011), pp. 335–344.