Verification of Concurrent Data Structures

From 2011-02-01 to 2016-01-31, closed project

Project details

<table>
<thead>
<tr>
<th>Total cost:</th>
<th>Topic(s):</th>
</tr>
</thead>
<tbody>
<tr>
<td>EUR 1 306 500</td>
<td>ERC-SG-PE6 - ERC Starting Grant - Computer science and informatics</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>EU contribution:</th>
<th>Call for proposal:</th>
</tr>
</thead>
<tbody>
<tr>
<td>EUR 1 306 500</td>
<td>ERC-2010-StG_20091028  See other projects for this call</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Coordinated in:</th>
<th>Funding scheme:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Netherlands</td>
<td>ERC-SG - ERC Starting Grant</td>
</tr>
</tbody>
</table>

Objective

Increasing performance demands, application complexity and explicit multi-core parallelism makes concurrency omnipresent in software applications. However, due to the complex interferences between threads in an application, concurrent software is also notoriously hard to get correct. Instead of spending large amounts of money to fix incorrect software, formal techniques are needed to reason about the behaviour of concurrent programs.

In earlier work, we developed a variant of permission-based separation logic that is particularly suited to reason about multithreaded Java programs with dynamic thread creation and termination, and reentrant locks. The VerCors project will extend expressiveness of the logic, to specify and verify concurrent data structures. The verification logic will be parameterised over the locking policy, so that a high-level specification of the behaviour of a data structure can be reused for different implementations. Thus the implementation of a concurrent data structure can be changed, without affecting correctness of the applications using it.

The logic will also be parameterised with concurrency and synchronisation primitives, so that a logic for a different programming language can be defined as an instance of the general logic. It will also be adapted to reason about programs with benign data races, i.e., data races where the same value is written simultaneously by different threads. Also techniques to generate part of the specifications automatically will be developed. Finally, the logic will be adapted to a distributed setting, where data consistency between the different sites has to be maintained.

All results will be integrated in a tool set that generates and proves proof obligations automatically. It will be validated on realistic case studies.

Related information

Report Summaries

Final Report Summary - VERCORS (Verification of Concurrent Data Structures)
**Principal Investigator**

Marieke Huisman  
Tel.: +31534894662  
Fax: +31534893247  
E-mail

**Host Institution**

UNIVERSITEIT TWENTE  
DRIENERLOOLAAN 5  
7522 NB ENSCHEDE  
Netherlands  
EU contribution: EUR 1 306 500

**Activity type:** Higher or Secondary Education Establishments  
**Administrative contact:** B.J. Pals  
Tel.: +31534893702  
Contact the organisation

**Beneficiaries**

UNIVERSITEIT TWENTE  
DRIENERLOOLAAN 5  
7522 NB ENSCHEDE  
Netherlands  
EU contribution: EUR 1 306 500

**Activity type:** Higher or Secondary Education Establishments  
**Administrative contact:** B.J. Pals  
Tel.: +31534893702  
Contact the organisation

**To know more**

http://erc.europa.eu/

**Subjects**

Information Processing and Information Systems - Physical sciences and engineering

**Last updated on** 2017-05-26  
**Retrieved on** 2019-07-27

**Permalink:** https://cordis.europa.eu/project/rcn/97362_en.html  
© European Union, 2019