An Application for a Certified Grid Computing Framework

Parallel Theorem Proving for Linear Logic

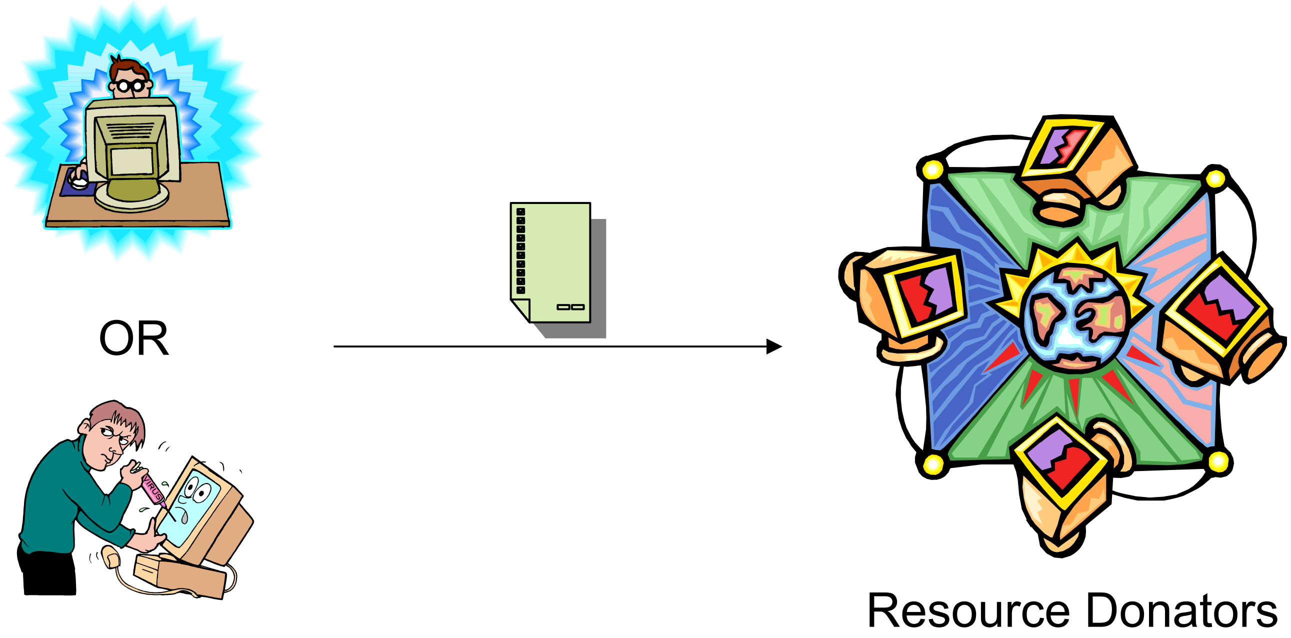
Bor-Yuh Evan Chang

Advisors: Prof. Robert Harper and Prof. Frank Pfenning

December 12, 2001

The Big Picture – the ConCert Project

- Suppose you had an ingeniously crafted massively parallelized algorithm to solve some problem. You would like use all the “wasted” computing resources of the Internet.
- Problem: How does a *resource donator* know you are a benevolent researcher and not a evil hacker?

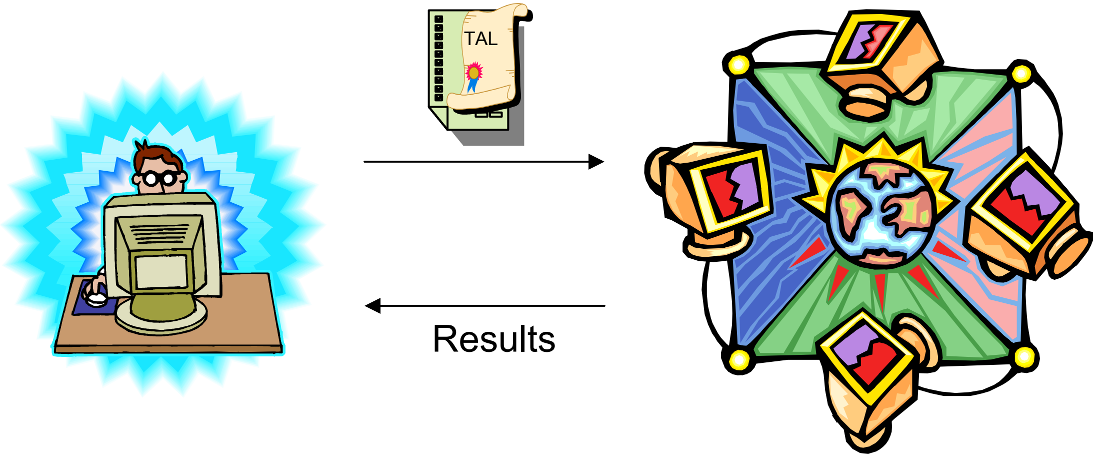


OR

Resource Donators

The Big Picture – the ConCert Project

- The ConCert project proposes to use *certified code* to resolve this issue of trust.



Results

Researcher

Resource Donators

Vision: Distributed-application developer's utilization of donated resources is completely transparent to the donator, but the donator is confident the specified safety, security, and privacy policies will not be violated.

My Contribution

Idea: The process of developing a substantial application using the ConCert infrastructure will help us better under the requirements on the infrastructure and how to program in such an environment.

- Goals
  - Make apparent the current shortcomings
  - Drive the architecture to a more robust and stable state
  - Better understand the requirements from a programmer's perspective

- What Application?
  - A *bottom-up* parallel theorem prover for intuitionistic linear logic
    - Advantages
      - the *focusing* strategy helps with producing independent subproblems
      - able to check validity of results easily
      - few existing linear logic provers
    - Concerns
      - how to balance the cost of communication
      - how to limit frivolous parallelism

Parallelism in Theorem Proving

- AND-parallelism

$$\dfrac{\begin{matrix}\vdots\\ \Gamma;\Delta \Longrightarrow A\end{matrix} \qquad\qquad \begin{matrix}\vdots\\ \Gamma;\Delta \Longrightarrow B\end{matrix}}{\Gamma;\Delta \Longrightarrow A \mathbin{\&} B}\ \&\mathrm{R}$$

↑ Direction of Search

- OR-parallelism ← exploitable

$$\dfrac{\begin{matrix}\vdots\\ \Gamma;\Delta \Longrightarrow A\end{matrix}}{\Gamma;\Delta \Longrightarrow A \oplus B}\ \oplus\mathrm{R}_1 \qquad\qquad \dfrac{\begin{matrix}\vdots\\ \Gamma;\Delta \Longrightarrow B\end{matrix}}{\Gamma;\Delta \Longrightarrow A \oplus B}\ \oplus\mathrm{R}_2$$

↑ Direction of Search

Algorithm Overview

- Focusing [Andreoli ’92][Pfenning ’01]
  - Refinement of the plain sequent calculus to reduce the non-determinism in proof search
  - Advantageous for parallelization by concentrating several non-deterministic choices into one place
  - Procedure:
    - first apply invertible rules eagerly
    - select a “focus” proposition and apply non-invertible rules until reach an atom or an invertible connective
    - upon reaching an atom, proof attempt either fails or succeeds

Algorithm Overview

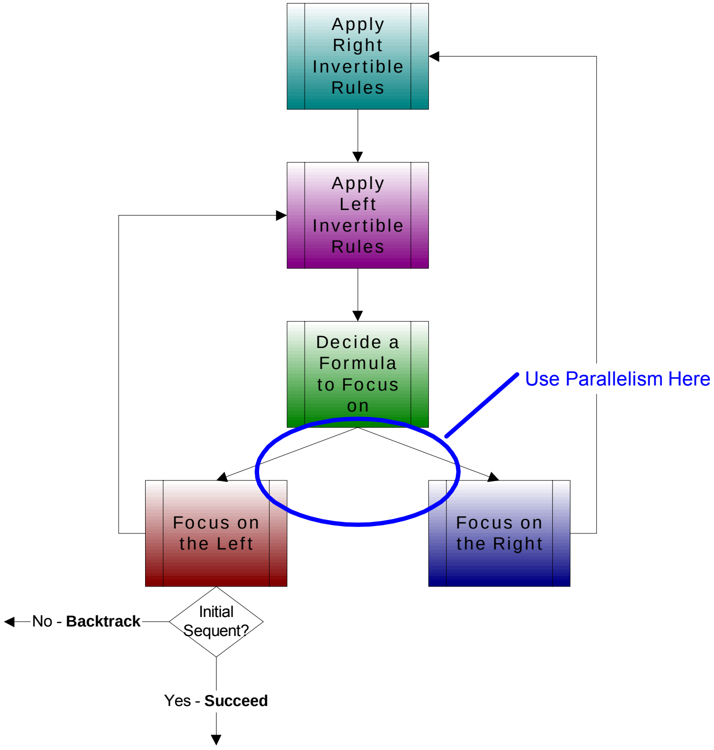
- *Resource-distribution via Boolean constraints* [Harland and Pym ’01]
  - Method to postpone the distribution of resources for multiplicative connectives

$$\dfrac{\begin{array}{c}\vdots\\ \Gamma;\Delta_1 \Longrightarrow A\end{array} \qquad \begin{array}{c}\vdots\\ \Gamma;\Delta_2 \Longrightarrow B\end{array}}{\Gamma;(\Delta_1,\Delta_2) \Longrightarrow A \otimes B}\;\otimes\mathrm{R}$$

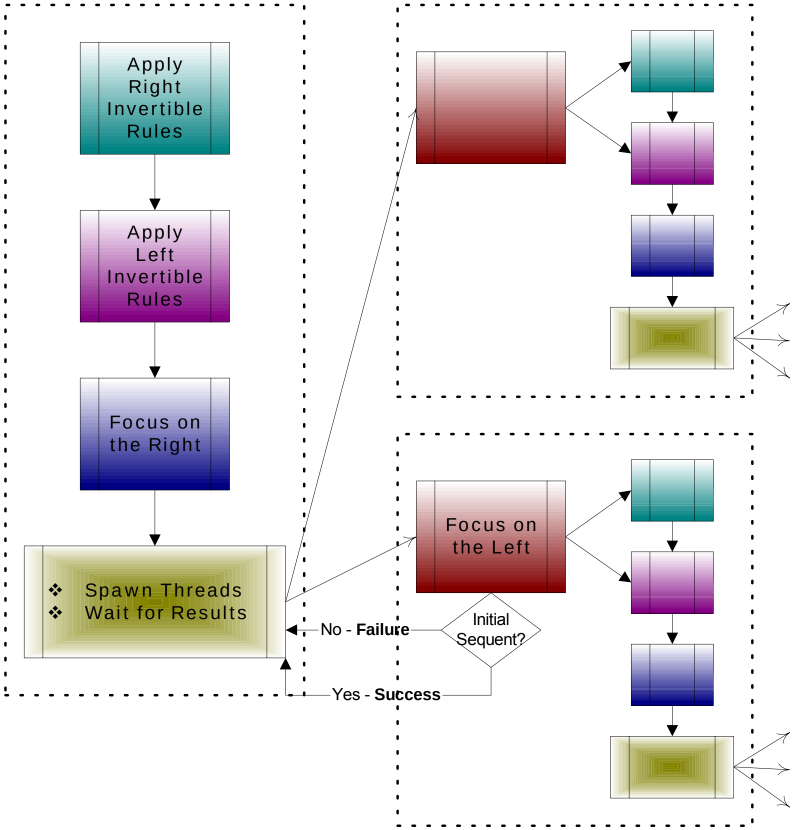
Direction of Search

  - Represent constraints using OBDDs (Ordered Binary Decision Diagrams)

Focusing



Sequential Implementation



Concurrent Implementation

Status Update

- Current Status
  - ☑ Built a working non-concurrent prover in SML
  - ☑ Modified prover to introduce concurrency using CML
- Next Steps
  - □ Theorem Proving Optimizations
    - eliminate spurious focusing based on logical compilation
    - integrate more efficient OBDD implementation
  - □ Extend theorem prover to return proofs
  - □ Integrate with the ConCert infrastructure