Risky Business: Developing an economic infrastructure for third party computation

Ian Wakeman

With Julian Rathke, Tim Owen and Others

University of Sussex

Outline

- Why should third party computation happen?
- Economic Models
- Programming Languages
- Platforms
- Reputation Systems

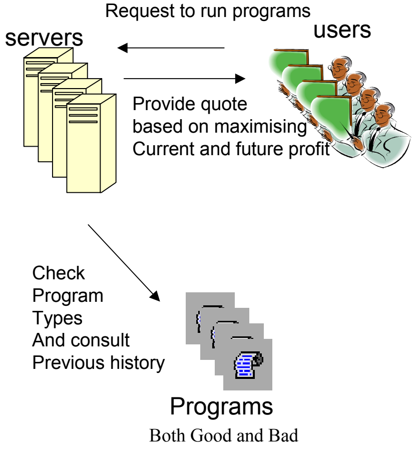
Third Party Computation

- Where is the novelty?
  - Most programs are written by third parties, but the platforms have been the property of the users
  - In the future, significantly more platforms will **not** be the property of the users
  - Payments needed to recover costs and maintain QoS
- Examples – Web hosting, pervasive computing, programmable network services

Economic Models

- Utility functions
  - Server performance degrades with popularity, user wants reliability
- Differential Pricing
  - Effective at capturing a range of markets.
- Risky Software
  - Bad software causes crashes, disrupts other customers
- Highly Scalable
  - Not just for Microsoft…

Providing the Incentives for Decent Software


Request to run programs

servers

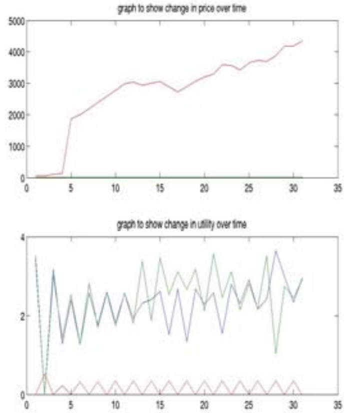
users

Provide quote
based on maximising
Current and future profit

Check
Program
Types
And consult
Previous history

Programs

Both Good and Bad



Mechanism Design

- What protocols do we use to provide strategy-proof tender and contract mechanisms?
  - Can we do better than registration and providing a credit card – PayPal and friends.
  - Digital Cash?
- How to protect against Denial of Service attacks?

Platforms

- Requirements
  - Language agnostic
  - Robust to failure and cross-interference
- XenoServers from Cambridge
- Execution Environments from ANs
- Trusted Computing Platforms

Programming Languages

- Support for Policies
  - Who can do what to whom when
  - Type systems supporting and checking contracts
- Resource Management
  - Providing a priori and measured data for resource usage
- Security Types
  - Assurance of correctness of protocols

Reputation Systems

- After each transaction, decide upon outcome.
- If all agree that it was positive, sign a certificate saying so.
- If it was negative, unilaterally sign a certificate saying so.
- Insert transaction certificate into transaction web

The Transaction Web

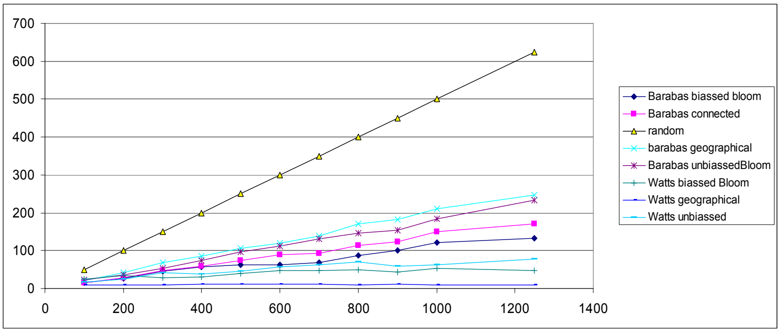
- The graph of transaction certficates between entities and software
- We want a strategy proof method of extracting an approximate indication of the "reputation" of some entity or software
  - Could look at links surrounding entity – but entity could insert bogus links.
  - Could look at minimum cut flow between the entities – but too costly
  - Can we discover random paths from "trusted" nodes to target?

Small Worlds

- Unverified Claim: The transaction graph will exhibit the properties of a small world graph
  - We will generally work with entities geographically close to us with a few remote links (Watts, Kleinberg)
  - The graph links will obey a power law, like the web, with a few very highly links entities such as Amazon and Microsoft (Balabas).
- The path finding can be based on heuristics such as “closest node” and “highest degree”.
  - Augment the graph with signposts generated from the source entities providing directions (Bloom Filters)

A Practical Scalable Reputation System

- Building an implementation based on public keys and distributed hash tables



Further Research Challenges

- Distributed market design
- HCI issues
- Business case