Software Quality and Infrastructure Protection for Diffuse Computing

Principal Investigator: Andre Scedrov
Institution: University of Pennsylvania
URL: http://www.cis.upenn.edu/spyce

NEW START May’01
The SPYCE Team ....

- Cynthia Dwork* (Compaq)
- Joan Feigenbaum (Yale)
- Joseph Y. Halpern (Cornell)
- Patrick D. Lincoln* (SRI)
- John C. Mitchell (Stanford)
- Jonathan M. Smith (U Penn)
- Paul Syverson* (NRL)
What is Diffuse Computing?

The computer diffuses into the environment as ... computation, communication, and storage performed by a distributed, networked collective invisibly in the background

"freeing people from the tyranny of the desktop computer"
Diffuse vs Pervasive, Ubiquitous

- **Pervasive Computing**
  - Access to information from anywhere
  - Many humans, one information network

- **Ubiquitous computing**
  - Lots of little devices everywhere
  - One human, many little computers

- **Diffuse Computing**
  - Development of services: compute, store, ...
  - Accessing and combining services robustly
  - Teams of users, many machines at-the-ready
Where is Diffuse Computing?

- **Hosts**
- **Routers**
- **Diffuse Computing Elements**
Initial Examples of The Power of Diffuse Computing

- SETI at home
- Protein folding
- Pervasive Computing
Why Diffuse Computing?

● Large commercial computing markets
  - Yet personalized computing support

● Huge potential of p2p architectures
  - Leverage potential of the “whole”

● Needs of network-centric systems
  - High assurance: you can bet your life on it
  - Survivable: resists massive cyber attack
  - Scalable: can grow to support government
  - Smart: distributed control over things
  - Affordable: infrastructure can grow quickly
Research Challenges in Diffuse Computing

- Providing high quality solutions out of lower-quality computing and network resources working together

  **Make ordinary computers do extra-ordinary things together**

- New mechanisms for stability in diffuse systems

  **Create new business opportunities**

- Components combined on an as-needed basis

  **Think about computing in terms of economics, physics, & systems metaphors**

- Local autonomy in ultralarge-scale distributed systems

  **ad hoc is in, tightly coupled is out**

- Given up for self-synchronization
Multi-Disciplinary Approach

- Combines 4 complementary thrusts:
  - Incentive-compatibility in distributed computing
  - Authorization mechanisms
  - Secure data storage and retrieval
  - Communication protocols

- Multi-institution experimental platform + systematic, formal treatment of underlying models, algorithms & data structures
Multi-Disciplinary Approach

- Combines 4 complementary thrusts:
  - Incentive-compatibility in distributed computing
  - Authorization mechanisms
  - Secure data storage and retrieval
  - Communication protocols

- Multi-institution experimental platform + systematic, formal treatment of underlying models, algorithms & data structures
Market System of Autonomous Agents

- "Mechanism Design" – how to achieve global goals with local autonomy?
- Behavior of software as a system, described formally in spite of incomplete knowledge
- Initial development of this methodology
- Multi-institutional experimental platform for prototyping
Both game theory and computer science focus on multi-agent distributed systems

- In game theory, the emphasis is on strategic thinking
  - agent’s goals as quantified by their utilities (payoffs)

- In CS, the focus is on fault-tolerance, dealing with asynchrony, and problems of scaling up (computational complexity)

For many practical applications, we need to combine these concerns.
Example: Routing in Networks

Different companies control various parts of the internet

- no company is enthusiastic about routing another company’s traffic through its portion
- But … they must cooperate to transmit traffic
- Negotiation is carried out using BGP (Border Gateway Protocol)
  - this is done badly
  - doesn’t take into account strategic thinking
Modeling this in the standard game-theoretic way is unlikely to work well:

- We want to deal with strategic behavior on the part of routers and with failures but ...

- We typically don’t have an accurate probability distribution characterizing failures and when moves are made

- Even if we had the relevant probabilities the obvious game tree would have uncountable outdegree

  - How can we compute good solutions efficiently?
More Problems

- How do we specify the desired behavior
  - This is a hybrid system, with continuous changes
    + discrete moves
  - How could a spec take into account, say, denial-of-service attacks and privacy concerns?
- How do we prove correctness?
**Mechanism Design**

- **Mechanism Design**: design a system in which strategic agents behave in socially desirable ways
  - well studied in economics
- **Algorithmic mechanism design** [NR99]
  - takes complexity into account
- We need **fault-tolerant, computationally efficient** algorithmic mechanism design for hybrid distributed systems
Previous Work

Computationally efficient mechanisms have been given for many problems of interest:

• Shortest paths (Nisan-Ronen 1999; Hershberger-Suri 2001)
• Multiagent scheduling (Wellman et al. 1998; Nisan-Ronen 1999)
• Combinatorial auctions (Parkes 1999; Nisan-Ronen 2000)
• Digital-goods auctions (Goldberg et al. 2001)

All use a single, centralized mechanism; none take faults into account.
Decentralized Algorithmic Mechanisms

Distribute the mechanism computation among all nodes in the network.

“Low network complexity” [FPS00]:
- Small total number of messages
- No link is a “hot spot”
- Small maximum message size
- Fast local processing

Open Problems Include

- Distributed multiagent-scheduling mechanisms
- (Distributed) mechanisms for DB-access and information retrieval
- Similar “user-layer” market-design problems
- Proofs of correctness
- Agent privacy
Multi-Disciplinary Approach

- Combines 4 complementary thrusts:
  - Incentive-compatibility in distributed computing
  - Authorization mechanisms
  - Secure data storage and retrieval
  - Communication protocols

- Multi-institution experimental platform + systematic, formal treatment of underlying models, algorithms & data structures
Outline

- Active networks and diffuse computing
- Experimental platform
  - ALIEN prototype
  - Extensions for market-based computation
- First experiments:
  - diffuse model in network control
- Plans for enhancing the infrastructure
Experimental Platform: Where?

- Hosts
- Routers
- Diffuse Computing Elements
Active Network Model

- Packets can change the behavior of the switches “on-the-fly”
  - In-band *active packets*
  - Out-of-band *active extensions*
Experimental Platform

• Based on ALIEN AN prototype
  – CAML language and runtime
  – Dynamic module loading (over the network)
  – Restricted general computation model (sandboxing)
  – Strong crypto support
The Design Space

- Usability vs. Flexibility vs. Security vs. Performance
- A General-Purpose Language gets the first two for free; other two are hard!
Protection vs. Quality

All Programs

Node Safe Programs

Network Safe Programs
Protection and Quality

Node Safe Programs

Market "Safe" Programs

Network Safe Programs

All Programs
Market-based computation on ALIEN

- Trading of “resource access rights”
  - Between producers, consumers, brokers
- Trust management
  - Express+verify resource access rights
  - Glue to administrative policy
- Embedded market mechanisms
  - For managing “raw” resources
Experiment: network control

Motivated by flaws in Internet model:
- Global cooperation assumed
  - For how much longer?
- Network-side function static
  - Users can’t touch routing
  - Infrastructure gets bloated
- Users are captives at the end-points
  - Latency, uncertainty

Clear need for a diffuse approach
The “Bourse of Packets”

- Non-cooperative environment
- Main ideas:
  - Diffuse services in the network
  - Embed strategy in active packets
- Expected impact:
  - Local+intelligent reaction to congestion
  - Increase utility, reclaim local autonomy
Enhancing the infrastructure

- Currently a local (per-node) market
- To scale up we need:
  - Distributed brokers / service location / information distribution / state management (starting from BGP...)
- ALIEN designed for routers:
  - How about diffuse elements, hosts?
Multi-Disciplinary Approach

- Combines 4 complementary thrusts:
  - Incentive-compatibility in distributed computing
  - Authorization mechanisms
  - Secure data storage and retrieval
  - Communication protocols

- Multi-institution experimental platform + systematic, formal treatment of underlying models, algorithms & data structures
When will Diffuse Computing be here?

- Currently an emerging paradigm
- Significant current commercial interest
- Increasing operational need
- Dramatic potential for DoD benefit
Diffuse Computing Support for Network-Centric Warfare

Attributes of Network-Centric Operations

- Degree of Interoperability
- Survivability
- Lethality
- Responsiveness
- Operational Tempo
- Ability to Self-Synchronize
- Increased Awareness

Major Impact

Attributes of Diffuse Network-Centric Operations
Focus: Middle Layer of Self-Synchronization

Sensor / Awareness

- Sensor Networks:
  - Joint Composite Tracking Network (CEC)
    - < 100 Users
  - Joint Data Network (Link 16/11)
    - < 500 Users
  - Joint Planning Network (GCCS)
    - ~1000 Users

Shooter / Transaction

- Sensor Fusion
- Weapons Control
- Force Control
- Self-Synchronizing Forces
- Force Coordination

Information Timeliness

- Minutes
- Seconds
- Sub-seconds

Information Accuracy

Variable Quality of Service

CEC: Cooperative Engagement Capability
GCCS: Global Command and Control System
Possible Impact of Successful Research on Diffuse Computing

- Improved Self Synchronization
- New forms of collaboration
- Compressed NCW OODA-loops
- Networked information-based acceleration of understanding the environment of a mission capability package
Expected Impact

- New range of “global” software-design techniques for today’s and tomorrow’s systems
- New software technology realizing full potential of network-centric computing
END