

An Architectural Style for Solving Computationally Intensive Problems on Large Networks

Yuriy Brun and Nenad Medvidovic
University of Southern California

The Internet as a Medium

| Internet | Challenges |
|---|--|
| <ul style="list-style-type: none">• Billion machines• Many are idle• Insecure | <ul style="list-style-type: none">• Computationally intensive• Mission-critical |

Can the Internet's power be used to solve these problems "securely"?

Sample Target Scenario

- Pharmaceutical Trials
 - Process large data sets
 - Do not share the data sets

Adaptive? Self-Managing?

- **Biologically inspired** software engineering
- Applicable to a variety of computations

Outline

- Distributing Computation
- Tile Assembly Model
- Tile Architectural Style
- Discreetness and Fault-Tolerance

Outline

→ Distributing Computation

- Tile Assembly Model
- Tile Architectural Style
- Discreetness and Fault-Tolerance

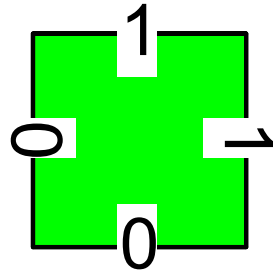
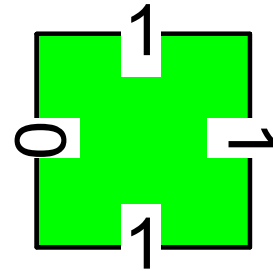
Distributed Computation

- SETI@home
- Google MapReduce
 - Distributes large datasets
 - Divides and conquers
- Computational Grids
- Nondiscreet

(Dean et al. 2004)

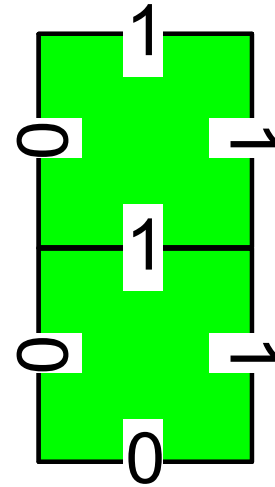
Tile Assembly Model

- A tile:
 - a square
 - labels on 4 sides
- Tiles attach
 - when labels match



Tile Assembly Model

- A tile:
 - a square
 - labels on 4 sides
- Tiles attach
 - when labels match

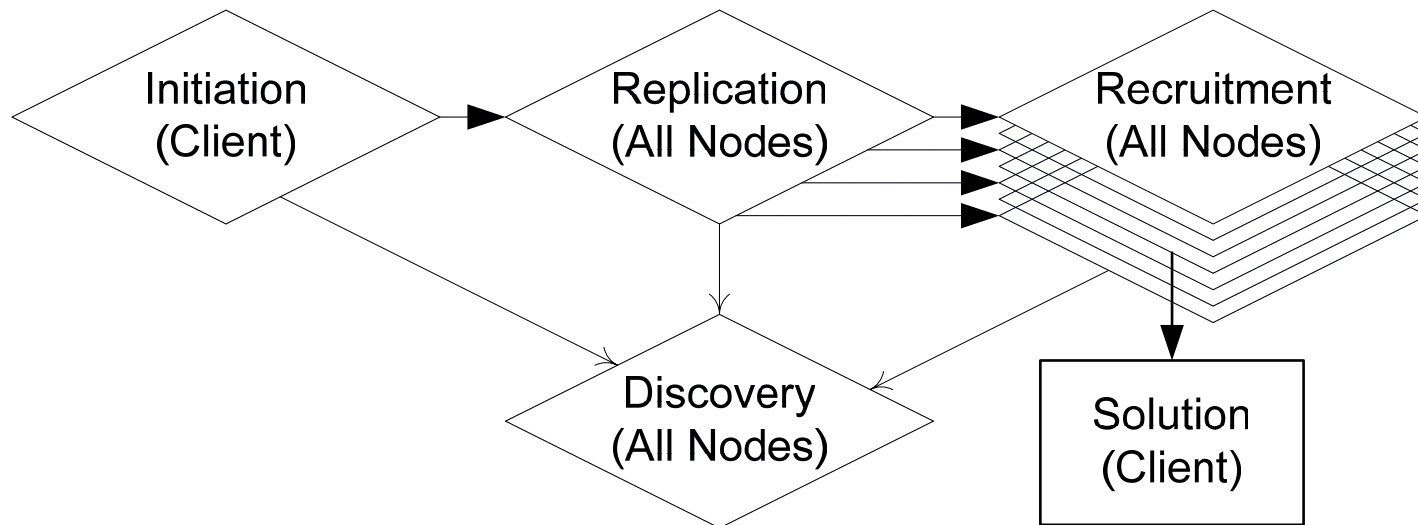


Outline

- Distributing Computation
- Tile Assembly Model
 - Tile Architectural Style
- Discreetness and Fault-Tolerance

Node Operations

- Initiation (by the client)
- Replication
- Recruitment
- Node Discovery

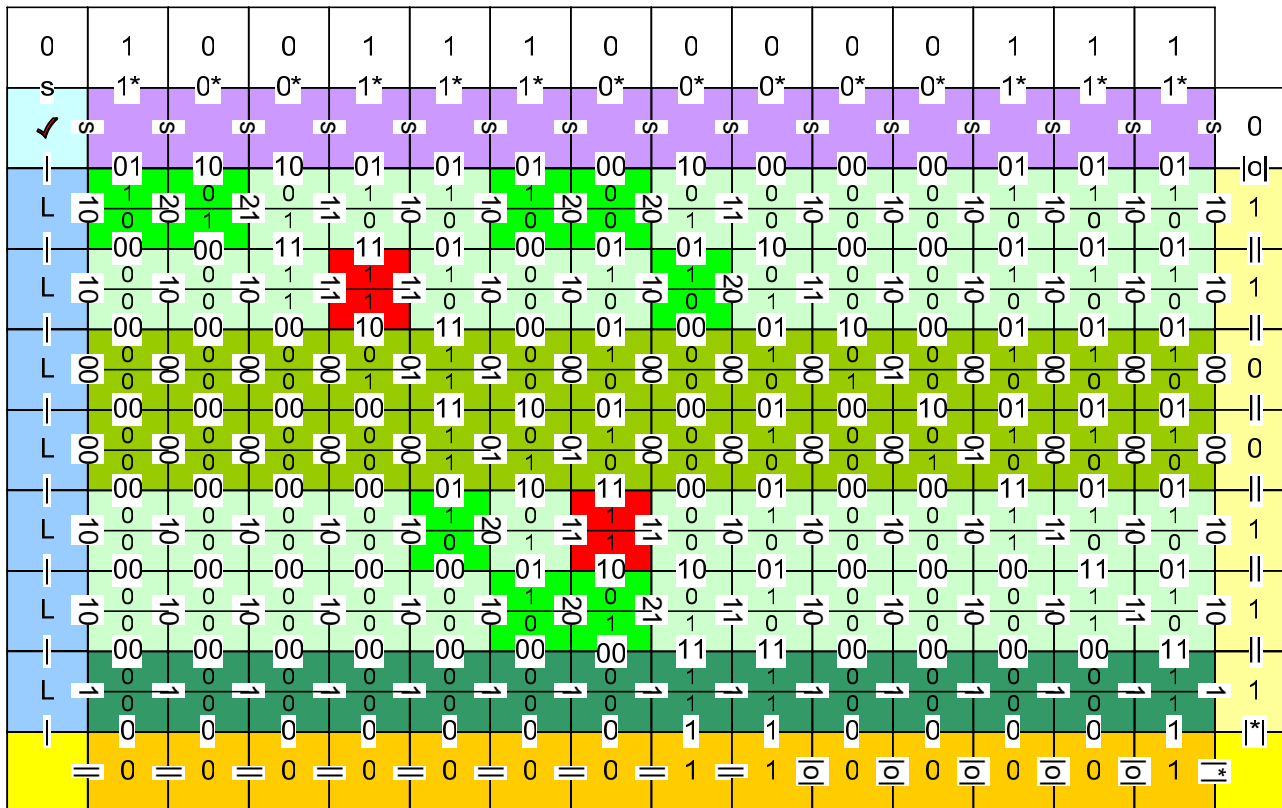


Analysis of the Tile Style

- Discreetness
 - Algorithm discreetness
 - Data discreetness
- Fault- and adversary-tolerance
- Scalability

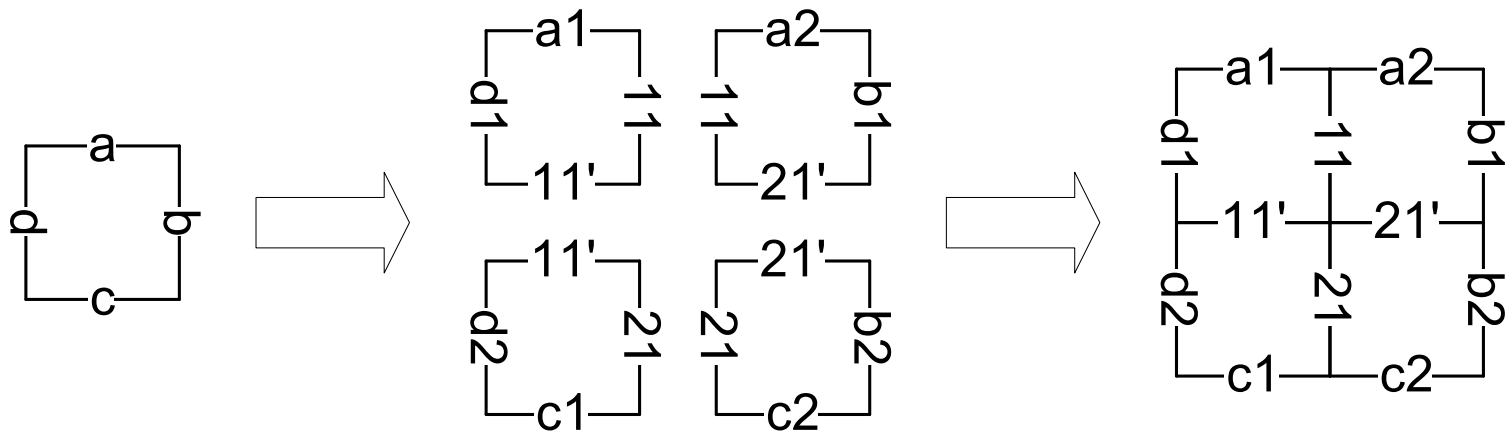
Discreetness (Data)

- Every tile component knows a single bit of data but not its location



Fault- and Adversary-Tolerance

- Tile systems can be designed to be tolerant to misbehaving tiles
- For example,



Contributions

- **Architectural style** for distributing computation on large networks in a manner that is
 - discreet
 - fault- and adversary-tolerant
 - scalable
- **Theoretical** analysis of the style
- **Self-assembling** computation
 - arithmetic
 - factoring
 - solving NP-complete problems

An Architectural Style for Solving Computationally Intensive Problems on Large Networks

Yuriy Brun and Nenad Medvidovic
University of Southern California