Social Simulations with the Actor Model for HPC

Sumihiko Natsu

CSG, Dept. of Creative Informatics, The University of Tokyo

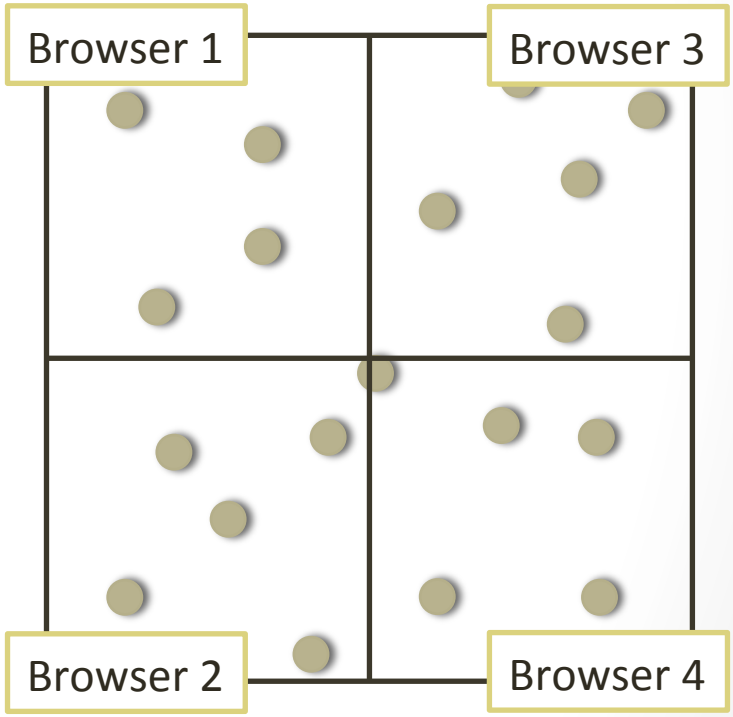
Last Rinko

Implement a “stone” interpreter with Lex & Yacc & LLVM

Distributed particle simulation with JavaScript

Ex. Fibonacci sequence

```
def fib(n:int):int {
  if (n > 1) {
    fib(n - 1) + fib(n - 2)
  } else {
    n
  }
}
fib(10)
```



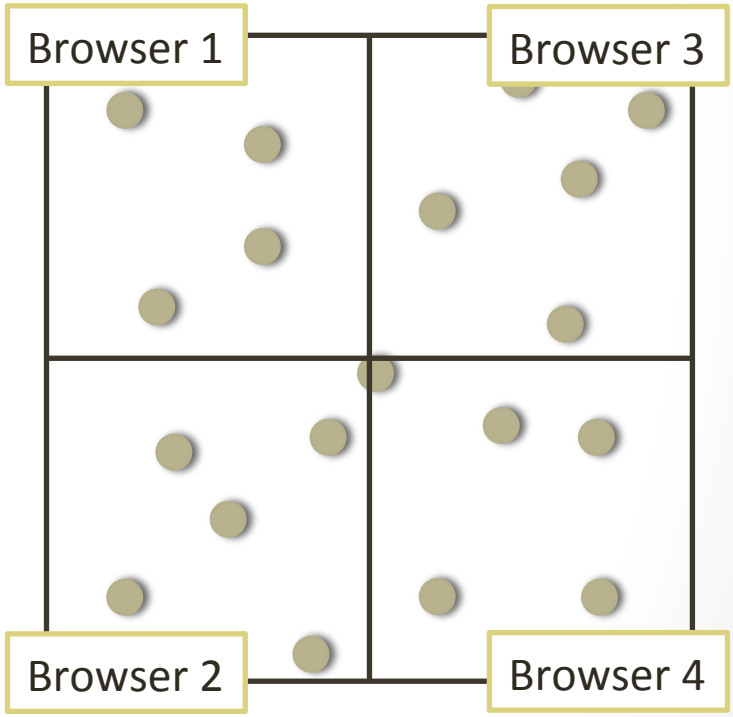
Last Rinko

Implement a “stone” interpreter with Lex & Yacc & LLVM

Ex. Fibonacci sequence

```
def fib(n:int):int {
  if (n > 1) {
    fib(n - 1) + fib(n - 2)
  } else {
    n
  }
}
fib(10)
```

Distributed particle simulation with JavaScript



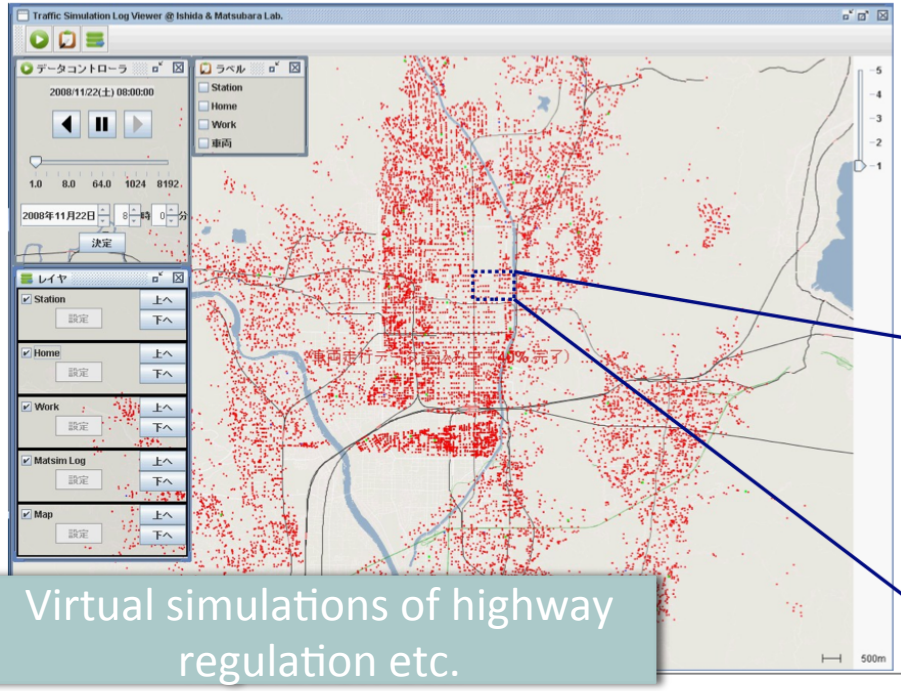
Social Simulations on HPC

Background

- Only few social simulations run on supercomputer
  - Users don't know what they can do by using HPC
    - They are not computer experts
  - Implementing social simulations on HPC is difficult
    - MPI, the de facto standard library on HPC, is too low level
    - Typical HPC applications (e.g. matrix multiplication)
      - Regular and synchronous
    - Social simulations
      - Irregular and asynchronous

Social simulation (agent based)

- Large scale traffic simulation (Kyoto)
  - Roads: 50000 links, 100000nodes
  - 100000 Agents
  - 200 days

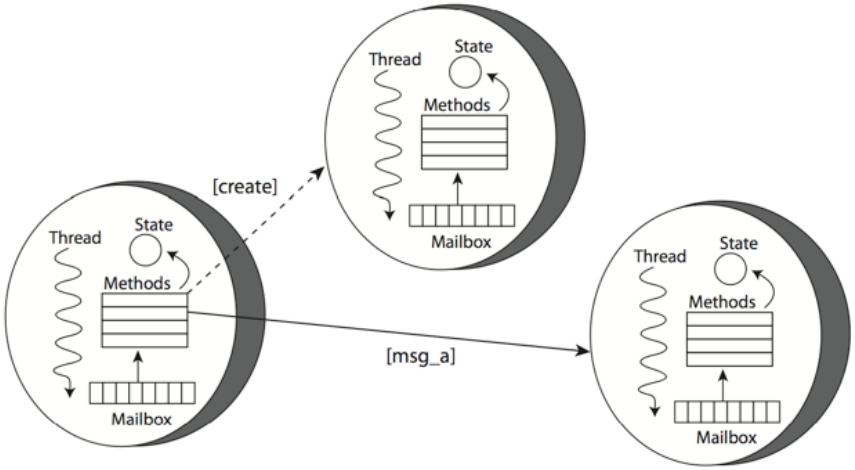


Virtual simulations of highway regulation etc.

Agent

- Destination selection
- Speed regulation
- Collision avoidance
- Optimum path finding

Actor Model [1973]



- A concurrent computation model
  - for distributed systems
  - message passing model
  - consists of many concurrent and autonomous entities called actors

- Each actor
  - has its own thread of control and state
  - sends and receives message with each other
  - updates its own local state by processing received messages

Actor Model

- Gained attention recently
  - The advent of multi-core CPUs and Cloud computing
  - Scala support actor on its standard library
- Other concurrent/parallel computing solutions
  - Threads + Socket
  - MPI
  - MapReduce
- Many social simulations
  - consist of a lot of objects modelled from real-world
  - interact with each other

Play with Actor

- Solve shortest path problem with actor model
  - Bellman-Ford algorithm
  - Very simple self-made actor framework
    - Macbook Pro
  - Akka
    - fx10
- Read some papers related to Actor
  - Actor Frameworks for the JVM Platform: A Comparative Analysis [Rajesh K. Karmani, et al., PPPJ 2009]
  - Actor Oriented Programming in Chapel [Amin Shali]
  - CUDAを用いたメッセージ送信型並行計算Actorモデルの実装 [高柳　亘, et al.]

Play with MPI

- To study differences between TCP/IP and MPI
  - Many actor libraries and researches aimed at TCP/IP
  - MPI is faster than TCP/IP in most supercomputer
- We found a serious performance degradation in a specific situation when using MPI
  - This might be happen if create a new actor library with MPI
  - Investigate the cause
    - OS, MPI, OpenMPI, OpenMPI Java binding?
- Conclusion
  - Misunderstanding about MPI's blocking and non-blocking send
    - Blocking-send is not “block until message received” but is “block until message safely copied to the buffer”

Research direction

- Create a actor library which can be used for social simulations on HPC
  - Problems
    - Scheduling
    - Locality
    - Fault tolerant 

The importance of fault tolerance for HPC

- As the number of HPC nodes grows, the possibility of failure also grows
  - Today's HPC have 100,000 nodes or more and this will increase if HPC become exascale
  - Ex.) Tianhe-1A
    - Fastest supercomputer in the TOP500 list (2010/11 ~ 2011/7)
    - MTBF: 6 hours

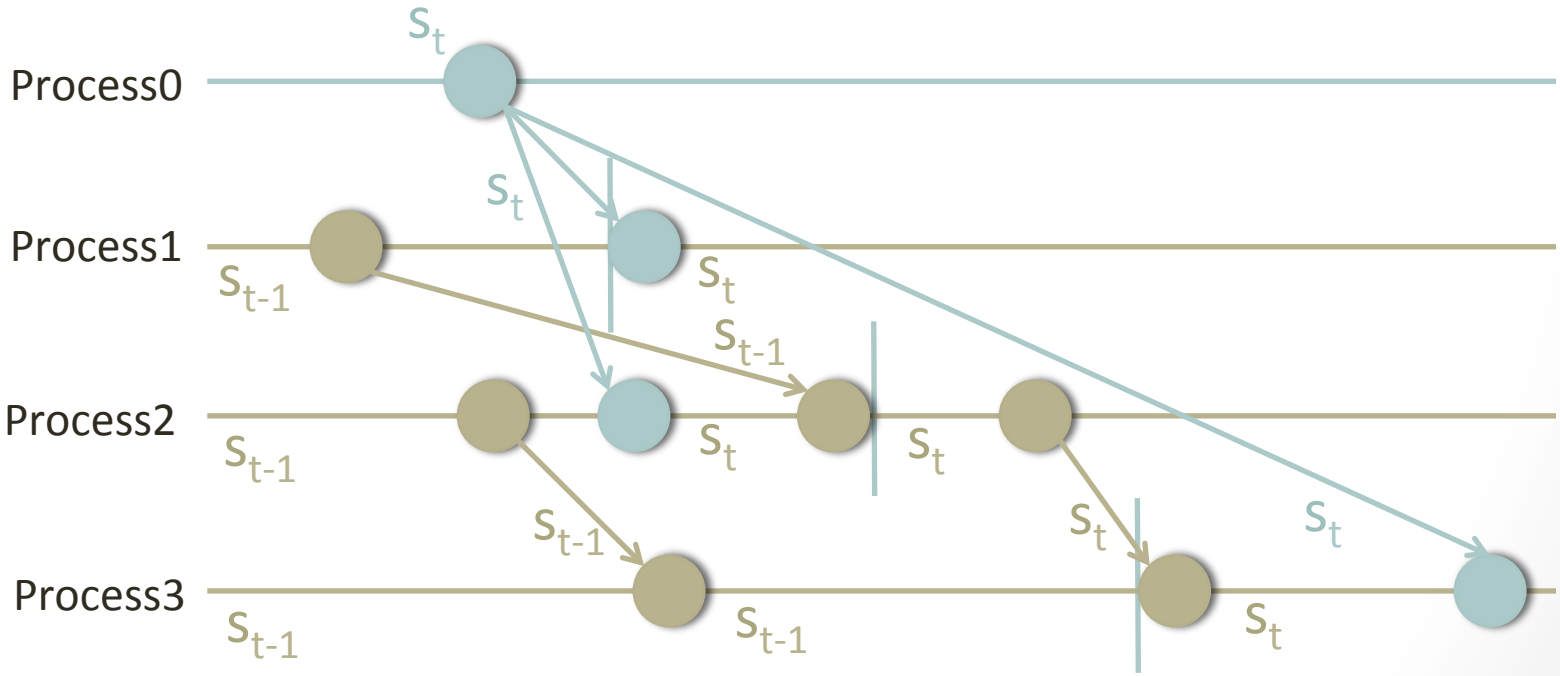
The study of fault tolerant system for message passing model, especially MPI, became more popular recently

Fault tolerant system for message passing model

- Coordinate checkpointing
- Message logging
- Hybrid

Coordinate checkpointing

- Take distributed system's snapshots with Chandy-Lamport algorithm
- Advantage
  - Easy to implement
- Disadvantage
  - Saving and retrieving a checkpoint can be very slow
  - When a failure occurs, all processes roll back to the last checkpoint



Message logging

- Messages are logged during failure-free execution and replayed in the same order after a failure
- Advantage
  - Only failed process has to roll back
- Disadvantage
  - Logs can grow very fast leading to a huge memory use

Fault tolerant actor system

- Social simulation's agents (= actors) are unbalanced
- ➢ Use idle HPC nodes to run backup actors
  - Receive corresponding message logs and replay them in advance

- Shorten MTTR furthermore than Hybrid
  - Availability = MTBF / (MTBF + MTTR)
- Replaying message logs in advance will reduce memory
  - HPC's memory size per CPU is small
- No need to save processes' entire stack trace
  - Only actor's states and mailbox

Future Work

- Implement a fault tolerant actor system
- Hold hearings from social simulation scientists to investigate ...
  - Is it really easy to write social simulation programs with Actor Model?
  - What they want if run the simulation on HPC?