The Use of Distributed Computing for Dynamic PRA: The ADS Approach

Dongfeng Zhu*
Yung Hsien Chang
Ali Mosleh

Center for Risk and Reliability
University of Maryland

Currently affiliated with ITEM Software

Purpose

- Dynamic Probabilistic Risk Assessment
  - Discrete Dynamic Event Tree
    - Systematically explore all scenarios
  - Continuous Event Tree Simulation
    - Randomly selecting system states and the timing of events

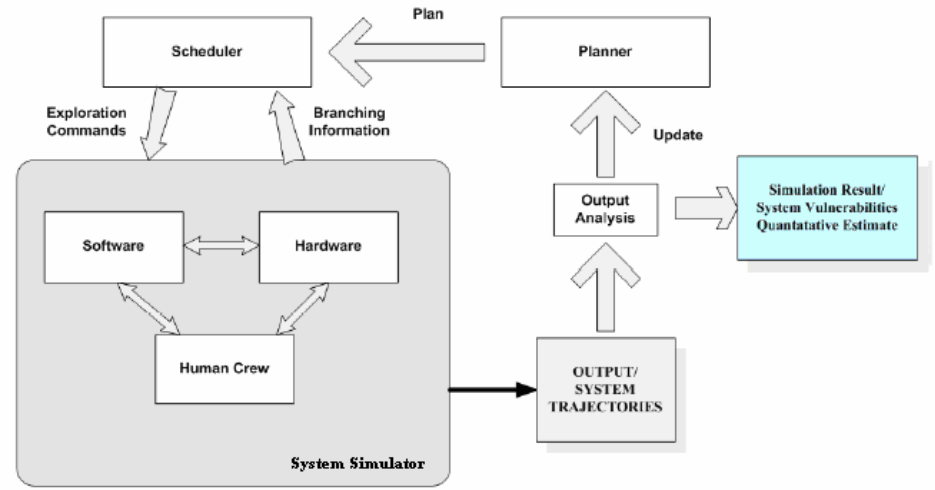
- State Explosion
  - The exponential growth in possible risk scenarios

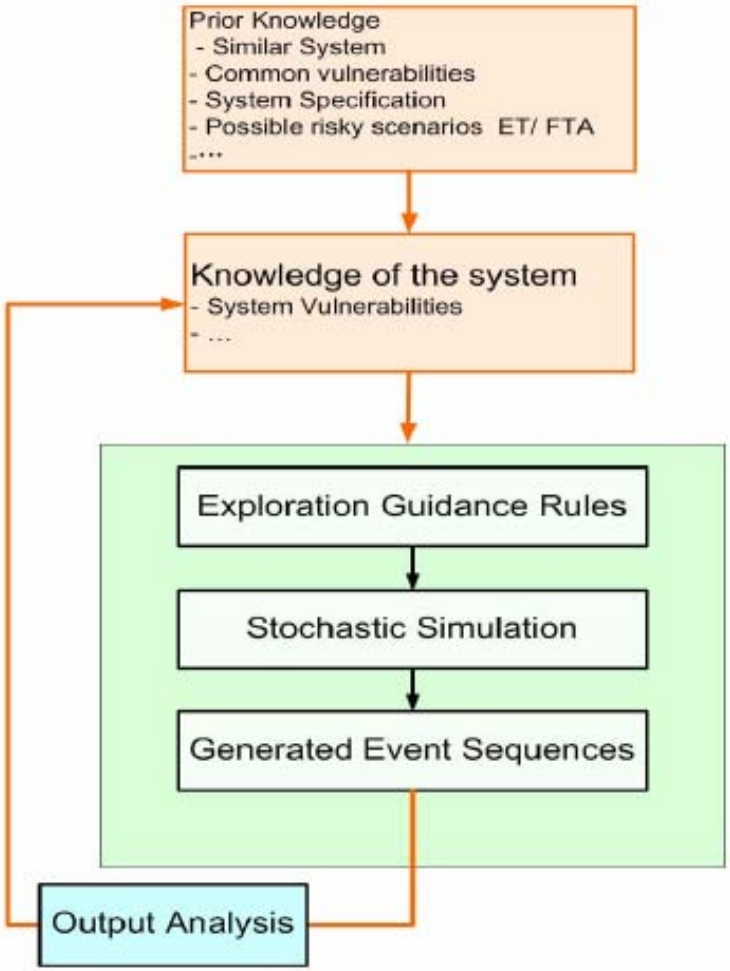
Current Approaches

- Reduce the number of risk scenarios
  - Combine system and operator states that lead to similar end states

- Bias the system and operator states toward interesting or risk significant events and end states
  - Reduces the computational effort expended on less important scenarios
  - Provides results for desired event sequences using less simulation effort

SimPRA


Plan
Scheduler
Planner
Exploration Commands
Branching Information
Update
Output Analysis
Simulation Result/ System Vulnerabilities Quantatative Estimate
Software
Hardware
Human Crew
OUTPUT/ SYSTEM TRAJECTORIES
System Simulator


Prior Knowledge
- Similar System
- Common vulnerabilities
- System Specification
- Possible risky scenarios ET/ FTA
-...
Knowledge of the system
- System Vulnerabilities
- ...
Exploration Guidance Rules
Stochastic Simulation
Generated Event Sequences
Output Analysis

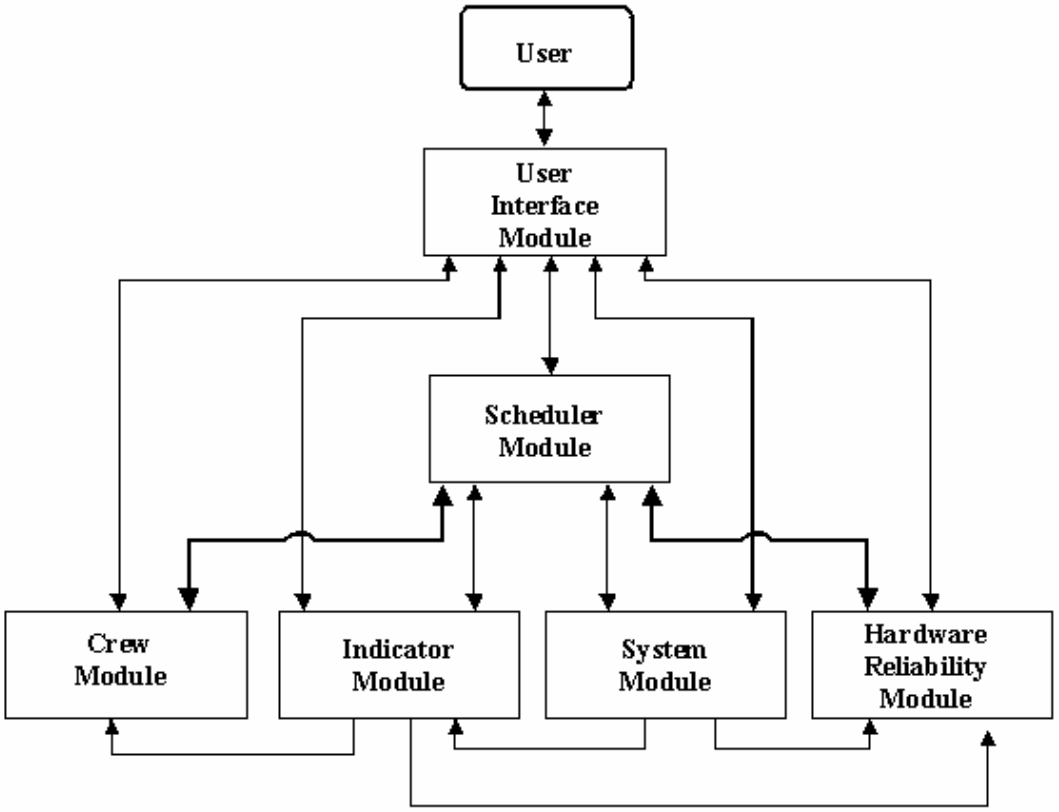
Distributed Computing

- Distributed Computing
  - Does not reduce the size of the state-space
  - Improves the efficiency of state-space exploration

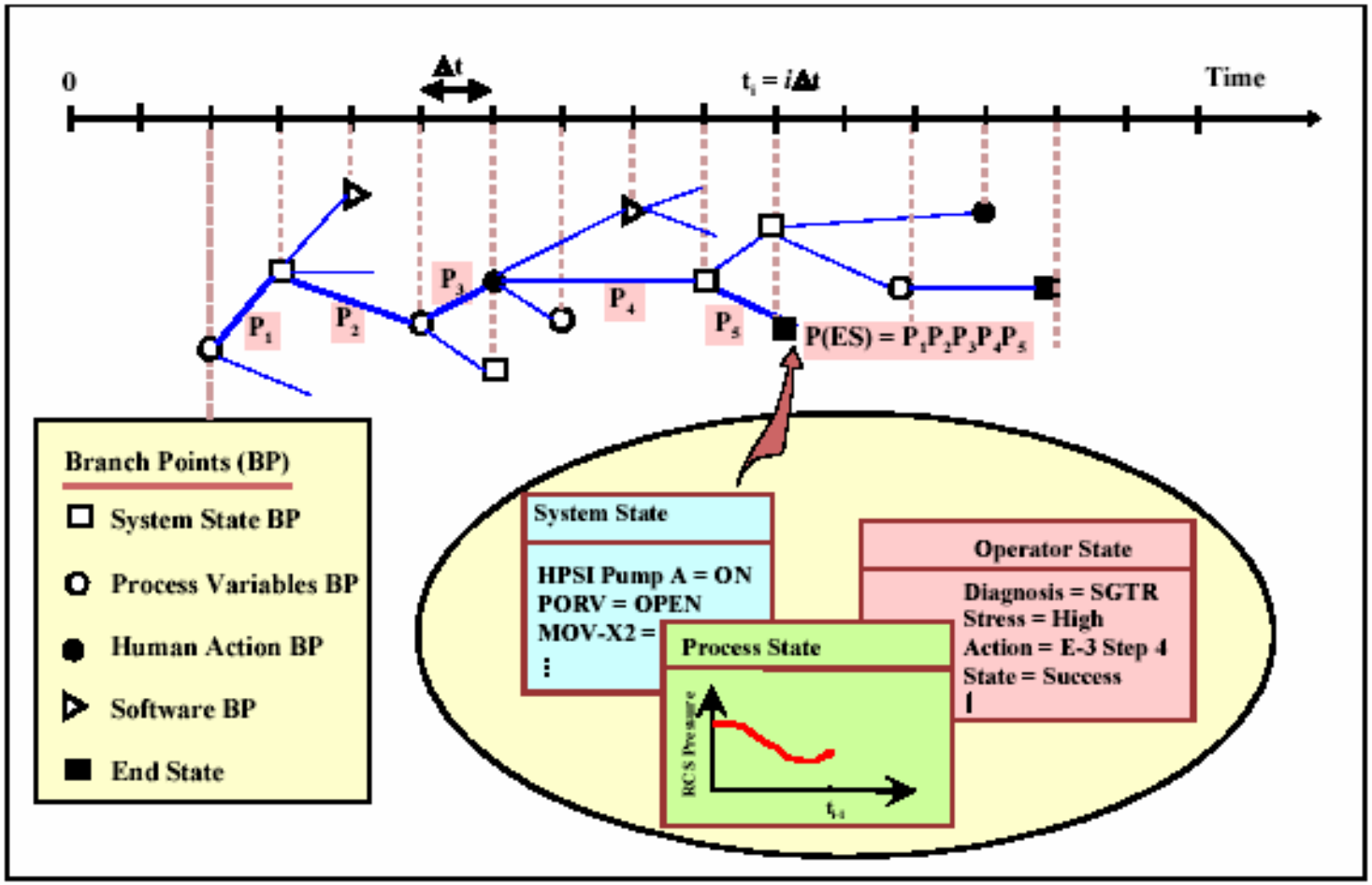
- Key Challenges
  - Divide a simulation into small segments that can run simultaneously
  - DDET
    - Reduce the size of the task information sent between the client and server
  - CET
    - Synchronization of biasing information

Accident Dynamic Simulator (ADS-IDAC)

- A simulation program designed to perform a dynamic PRA of nuclear power plants



ADS

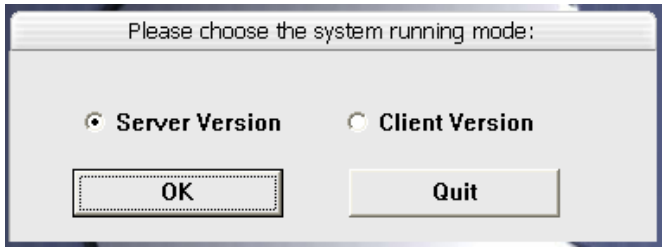

0
Δt
$t_i = i\Delta t$
Time
P_1
P_2
P_3
P_4
P_5
$P(ES) = P_1P_2P_3P_4P_5$
Branch Points (BP)
System State BP
Process Variables BP
Human Action BP
Software BP
End State
System State
HPSI Pump A = ON
PORV = OPEN
MOV-X2 =
Operator State
Diagnosis = SGTR
Stress = High
Action = E-3 Step 4
State = Success
Process State
RCS Pressure
t_{si}

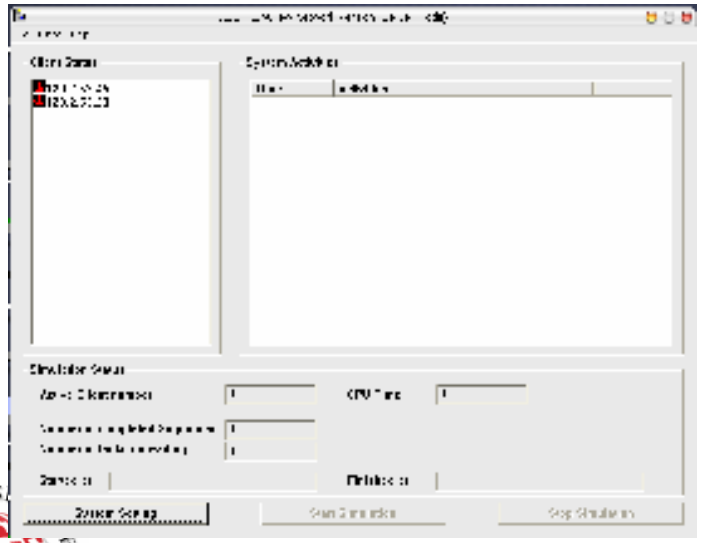
ADS Net Architecture

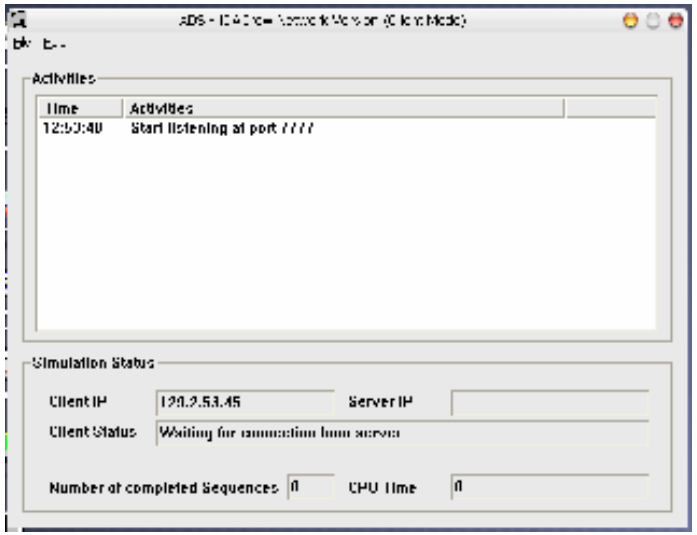
- Central Server
  - Scheduler
  - Communicator
  - Client Monitor
  - Simulation Task Manager
  - Simulation Result Repository
  - Post-simulation Processor

- Clients
  - Communicator
  - Simulation Task Manager
  - Simulator

ADS Net


Please choose the system running mode:
Server Version
Client Version
OK
Quit




Activities
Time
Activities
Start listening at port 7777
Simulation Status
Client IP
Server IP
Client Status
Number of completed Sequences
CPU Time

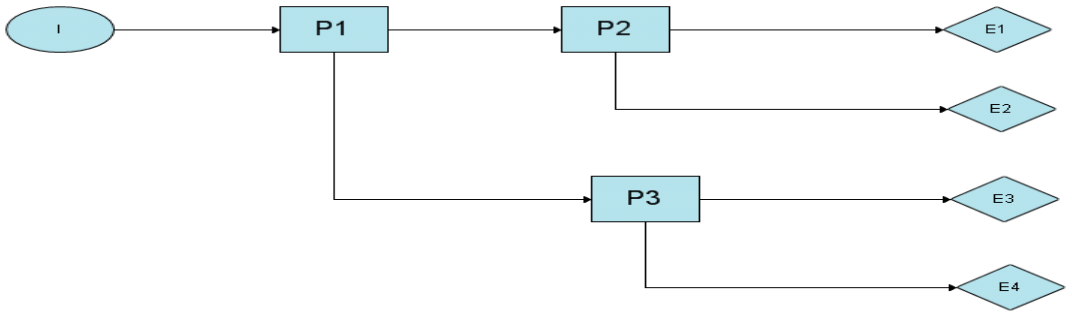
Server

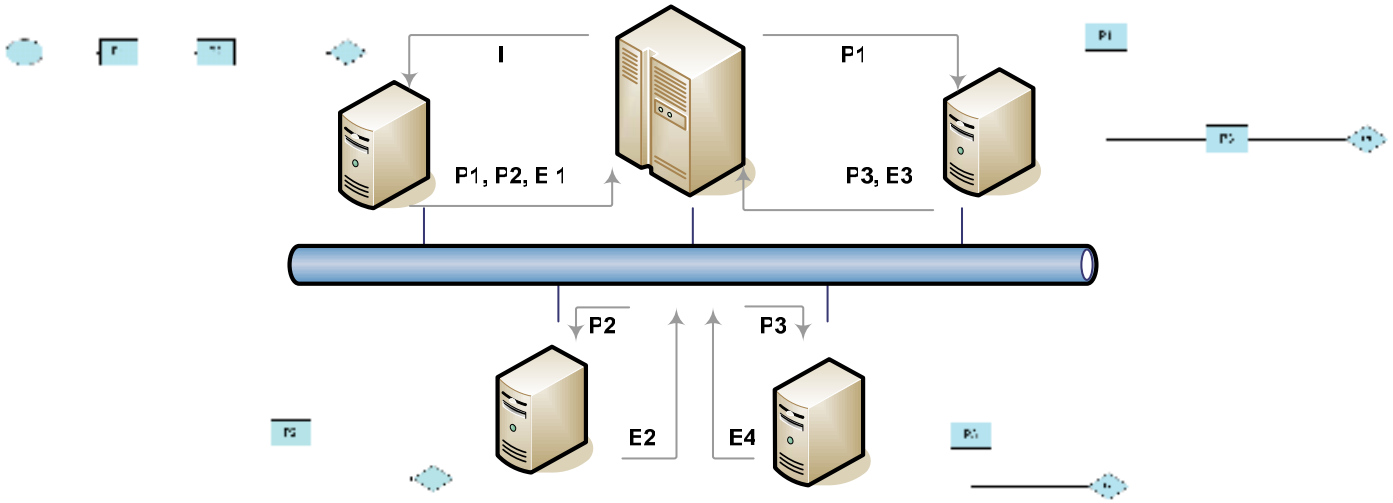
```
Initialize client Monitor, update client Status
Load Input Data
Send the input data to first client
while (total number of tasks running on client side and pending in the task repository is greater than 0) {
    Wait for the client/ client status monitor to send a message {
        If (new tasks message received) {
            Add the new tasks to the task repository
            Update running task information for the client
            Check if (any clients available) {
                Send new tasks to next available clients
            }
        }
        else if (end state message received) {
            Save results in the result repository
            if (tasks pending in the task repository) {
                Send the next available task to the client
            }
        }
        else if (new client available message) {
            if (client is back online) {
                Notify client to clean all previous simulation
            }
            check  if (tasks available in the repository) {
                Send next available task to the client
            }
        }
    }
}
Start post simulation processing
End simulation
```

Client

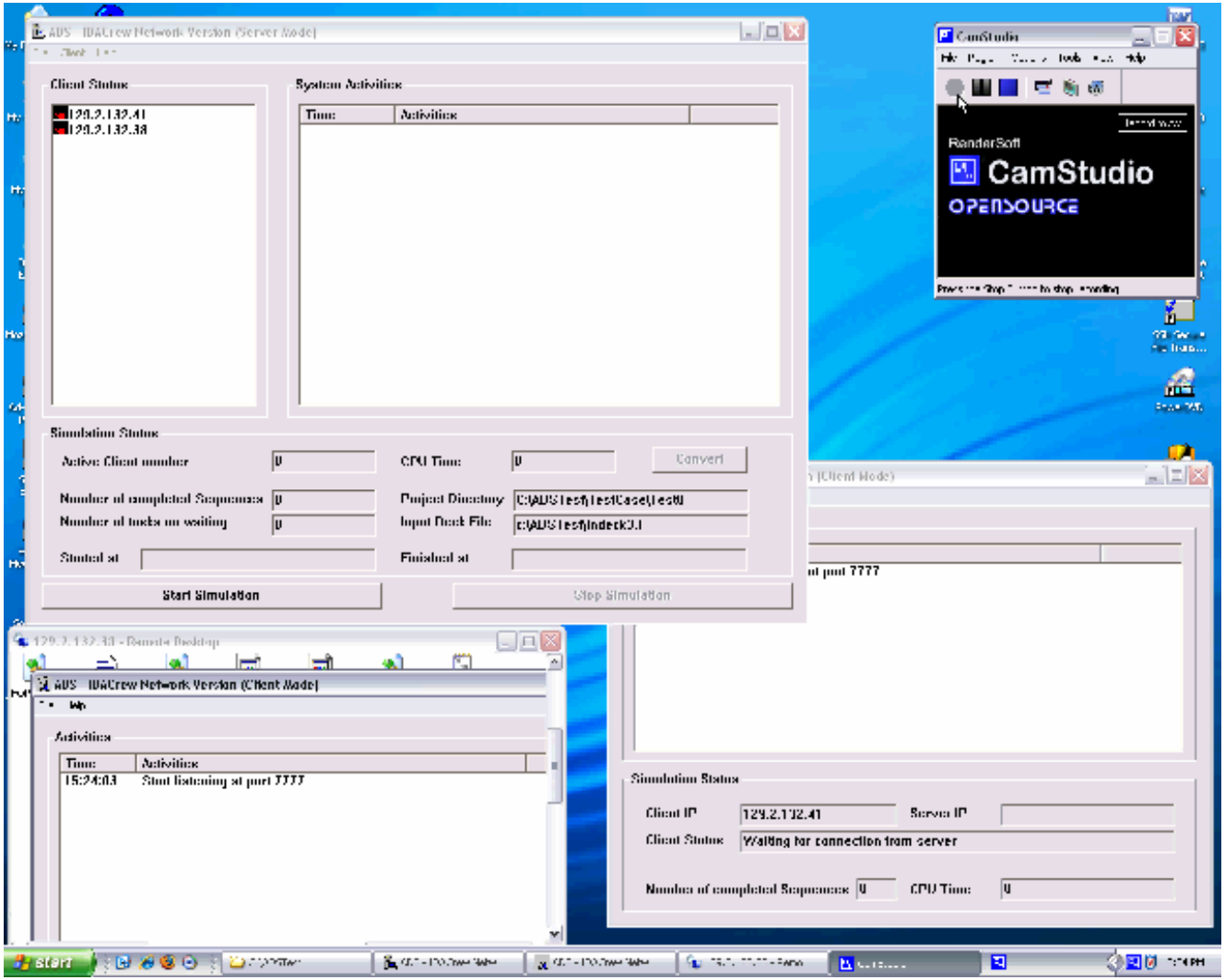
```
Initialize client communicator
Open a port and wait for server to establish connection
if (connection established) {
    while (connection is active and exit command != false) {
    Wait for the new task from server{
        If (new tasks received) {
            Load the task information using simulation task manager
            Start simulation
            while (end state not reached) {
                continue simulation
                if (branch point reached) {
                    Save system status
                    Create new task for each additional branch
                    Send system status and branch information back to server
                }
            }
            Send the system status and end state back to server
            clean memory;
        }
        else if (reset message received) {
            stop the current simulation;
            clean memory;
        }
    }
}
```

Computational Task Distribution





Video


Client Status
System Activities
Time
Activities
Simulation Status
Start Simulation
Stop Simulation
RenderSoft
CamStudio
OPENSOURCE
Activities
Time
Activities
15:24:03 Start listening at port 7777
Simulation Status
Client IP
129.2.132.41
Client Status
Waiting for connection from server
Number of completed Sequences
CPU Time
start

Example

- United States Pressurized Water Reactor (PWR) nuclear power plant model
  - Initiating Event: SGTR event with a rupture size of two centimeters in diameter
  - 32 sequences and 305 pivotal events

- Environment
  - Intel Pentium Duo processor running at 2.8 GHZ
  - 1GB Memory
  - A network setting with connection of 1, 2, 3, 4, and 5 client computers

Example Results