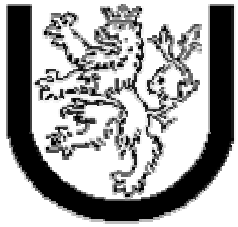
Load Redistribution in Heterogeneous Systems

Tomáš Koutný



University of West Bohemia
Faculty of Applied Sciences
Department of Computer Science and Engineering
Plzeň, Czech Republic

Load Redistribution in Heterogeneous Systems

- Motivation & Goals
- Active Networks
- Communication
- Performance
- Decision Strategy
- Sample Case
- Runtime Results
- Simulation
- Conclusion & Future Work

Motivation & Goals

- Disadvantages of traditional distributed and dynamic load-balancing methods:
  - They do not assume the heterogeneity of used hardware
  - They do not assume dynamically changing network
  - Processes are assigned onto nodes based on nodes free resources only
  - Many of them consider just the processor utilization as the only resource
  - They do not take full potential of modern environments such as active networks

Motivation & Goals

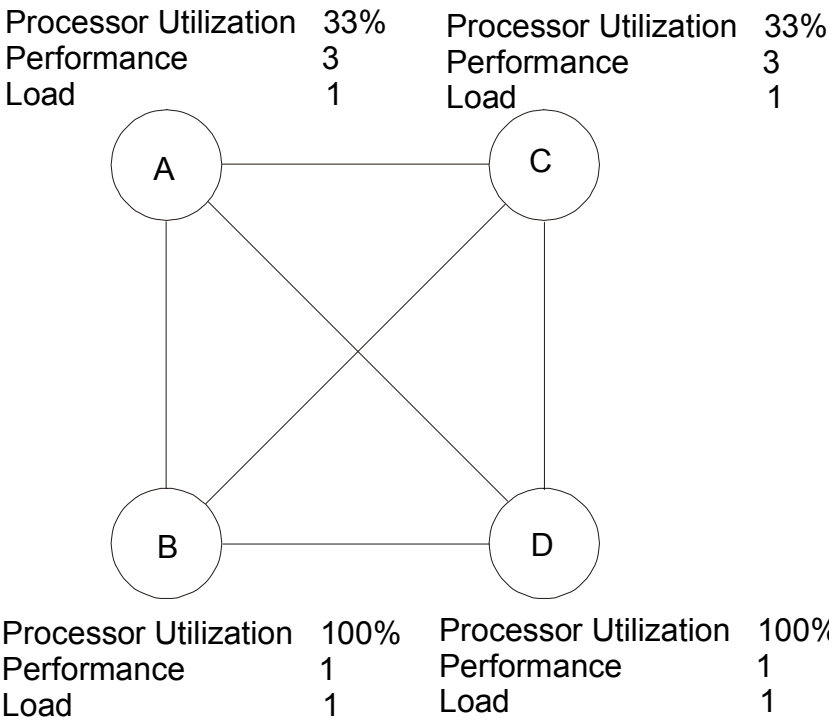


Figure 1. Traditional Load Balancing

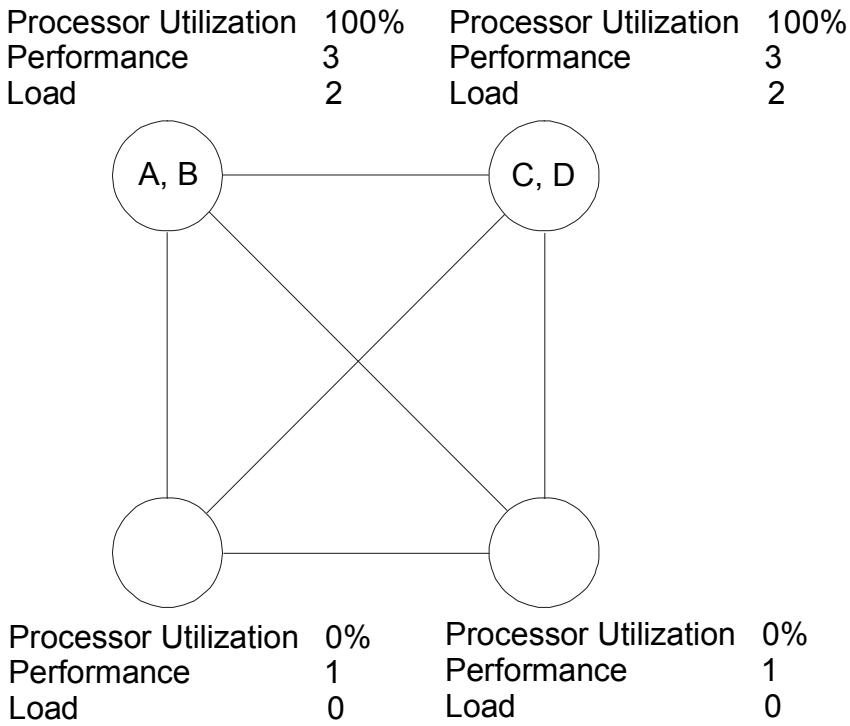
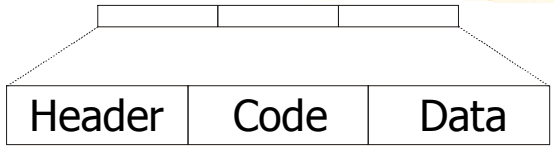


Figure 2. Optimal Assignment of Processes

Active Networks
The Implementation Choice



- Capsules supersede packets
  - Each capsule is associated with a code that is run every time capsule visits a node
  - The code may perform a number of activities such as
    - Specialized routing
    - Gathering performance snapshots
- Distributed application consists of possibly many active applications, which inject capsules
- Each active application and capsule runs as a process in a standardized Execution Environment

Communication

- Processes are being created, move and terminate dynamically
- Intensity of communication between a particular process and other processes varies dynamically
- Adapting to currently used part of a communication model
- Communication cluster identifies processes communicating together
- Communication clusters are used to reduce communication delays

Performance

- Performance snapshot
  - Processor benchmark – a number of well selected instructions executed within a given interval
  - Number of processors
  - Average usage of all processors
  - Least average usage of a single processor
  - Average number of processes/AN EEs
  - Available physical memory

Decision Strategy

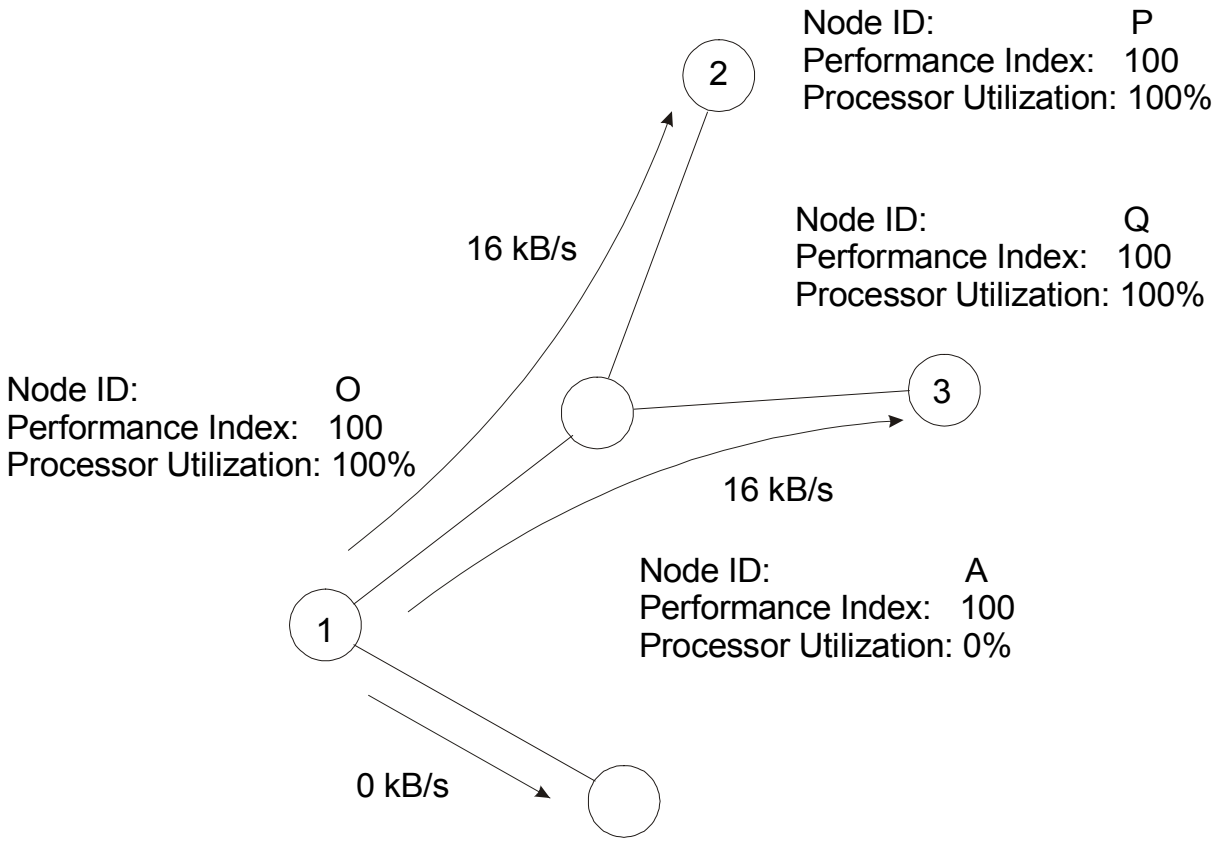
- Compare and select
  - Node performance

$$local\ node's\ performance = \frac{used\ \ processor\ time \times benchmark}{transmission\ time\ between\ nodes}$$

$$remote\ node's\ performance = \frac{(1 - lowest\ usage\ of\ \ one\ \ processor) \times benchmark}{transmission\ time\ between\ nodes}$$

  - Communication clusters
- Credit system
- Fighting the mass migration using blackboards
- Load is being well distributed incrementally

Sample Case

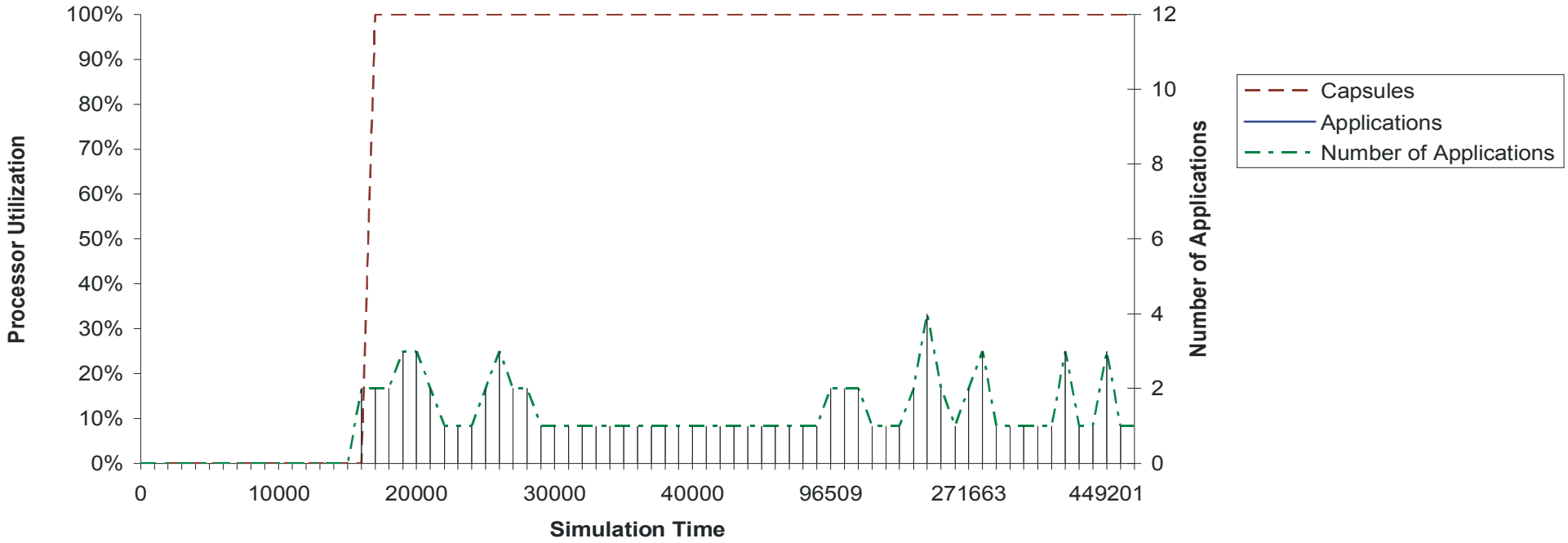

Node ID: P
Performance Index: 100
Processor Utilization: 100%
2
16 kB/s
Node ID: Q
Performance Index: 100
Processor Utilization: 100%
Node ID: O
Performance Index: 100
Processor Utilization: 100%
3
16 kB/s
Node ID: A
Performance Index: 100
Processor Utilization: 0%
1
0 kB/s
Node ID: B
Performance Index: 100
Processor Utilization: 0%

Runtime Results

- Testing application
  - Parallel prefix sum computation
  - Array of values is distributed across computing processes
  - Usage of synchronization primitives
    - Barrier and non-barrier synchronization is used
  - Configurable degree of processor-time consumption
- No benefit for a particular application
- Measured values were used in a simulation

Simulation

- Method has to adapt to an unknown application
  - => Simulation randomly changes virtual topology
  - => Communication clusters change



Conclusion & Future Work

- Load-redistribution
  - Dynamic and distributed
  - Modest complexity
  - No important requirements
  - Co-operative and adaptive
  - Supports heterogeneous systems
- Smart Active Node
  - Controlled code interpretation => Virtual machine
  - Utilize the knowledge from the simulation