Cycle Sharing Systems

Jagadeesh Dyaberi
Dependable Computing Systems Lab
Purdue University

Outline

- Introduction
- Design of Program
- Security
- Communication
- Architecture
- Implementation
- Conclusion

Introduction

- What is cycle-sharing?

Sharing CPU resources across a network so that all machines function as one large supercomputer.

Cycle sharing model across the Internet

Introduction

- Why Internet?
  - 93 million connected hosts
  - Rapid growth in computing power
  - Revolutionary expansion of mobile devices like cellular phones and PDA's.

Introduction

- Examples of Cycle Sharing Systems
  - SETI @ Home
  - Mersenne Prime Search
  - Distributed.net
  - XtremWeb

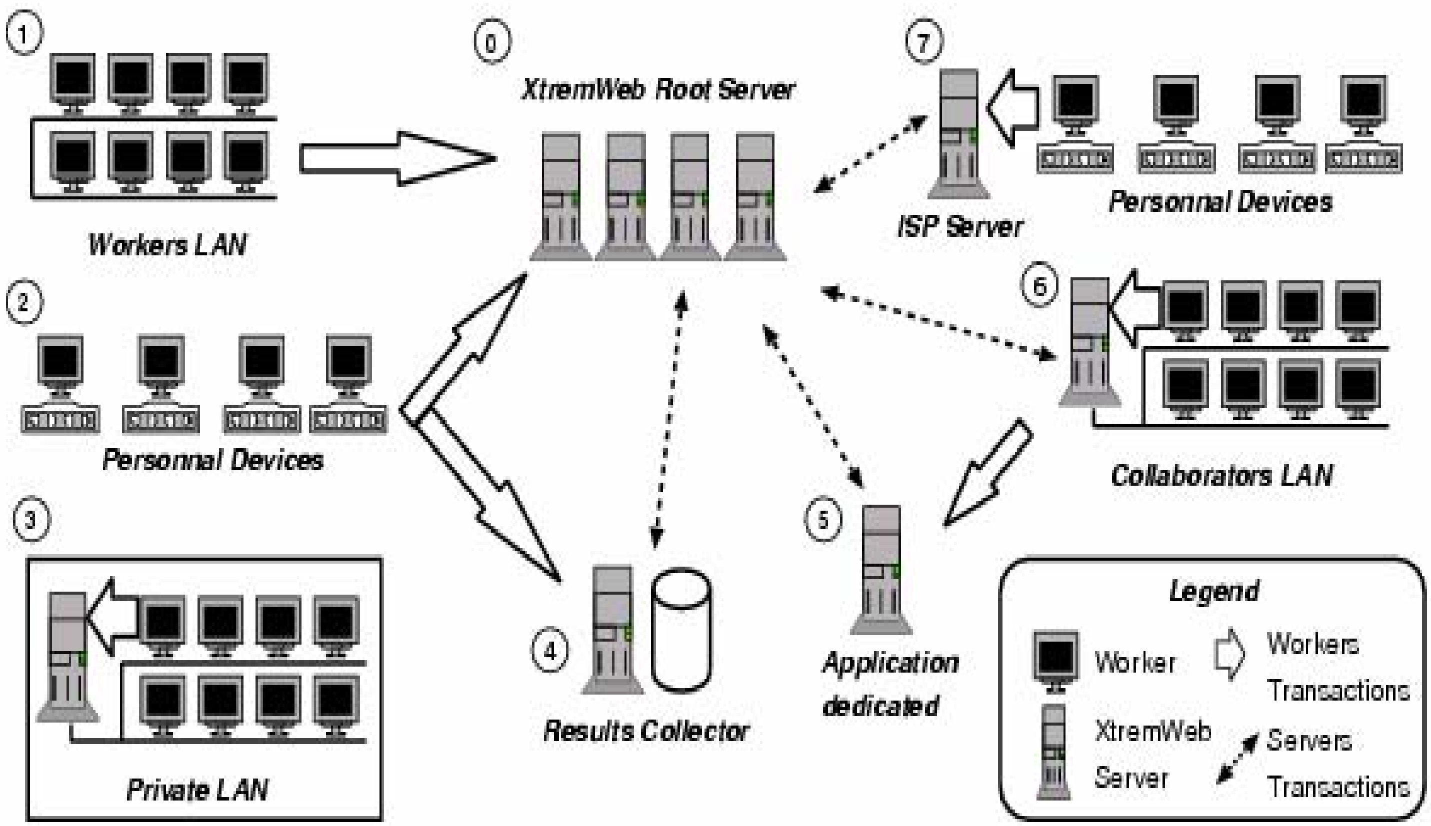
Introduction

- Global Computing Issues
  - Scalability
    - Scale to hundreds of thousands of nodes
  - Heterogeneity
    - Across hardware, OS and basic software
  - Availability
    - Owner sets limits on resource
  - Fault Tolerance
    - Must accept frequent faults while maintaining performance
  - Security
    - Protection from malicious or erroneous manipulations
  - Dynamicity
    - Varying configuration, communication latency

XtremWeb

- Two Ways of using XtremWeb
  1. Worker
     - Volunteer Machine
     - Registers with the administration server
     - Contributes when idle
  2. Collaborator
     - Setup their own global distributed application
     - Works for the main XtremWeb
       1. Idle
       2. Has no work to send to his community of PC's.

XtremWeb


1
0
XtremWeb Root Server
7
Workers LAN
ISP Server
Personnal Devices
2
6
Personnal Devices
Collaborators LAN
3
5
4
Results Collector
Application
dedicated
Private LAN
Legend
Worker
Workers
Transactions
XtremWeb
Server
Servers
Transactions

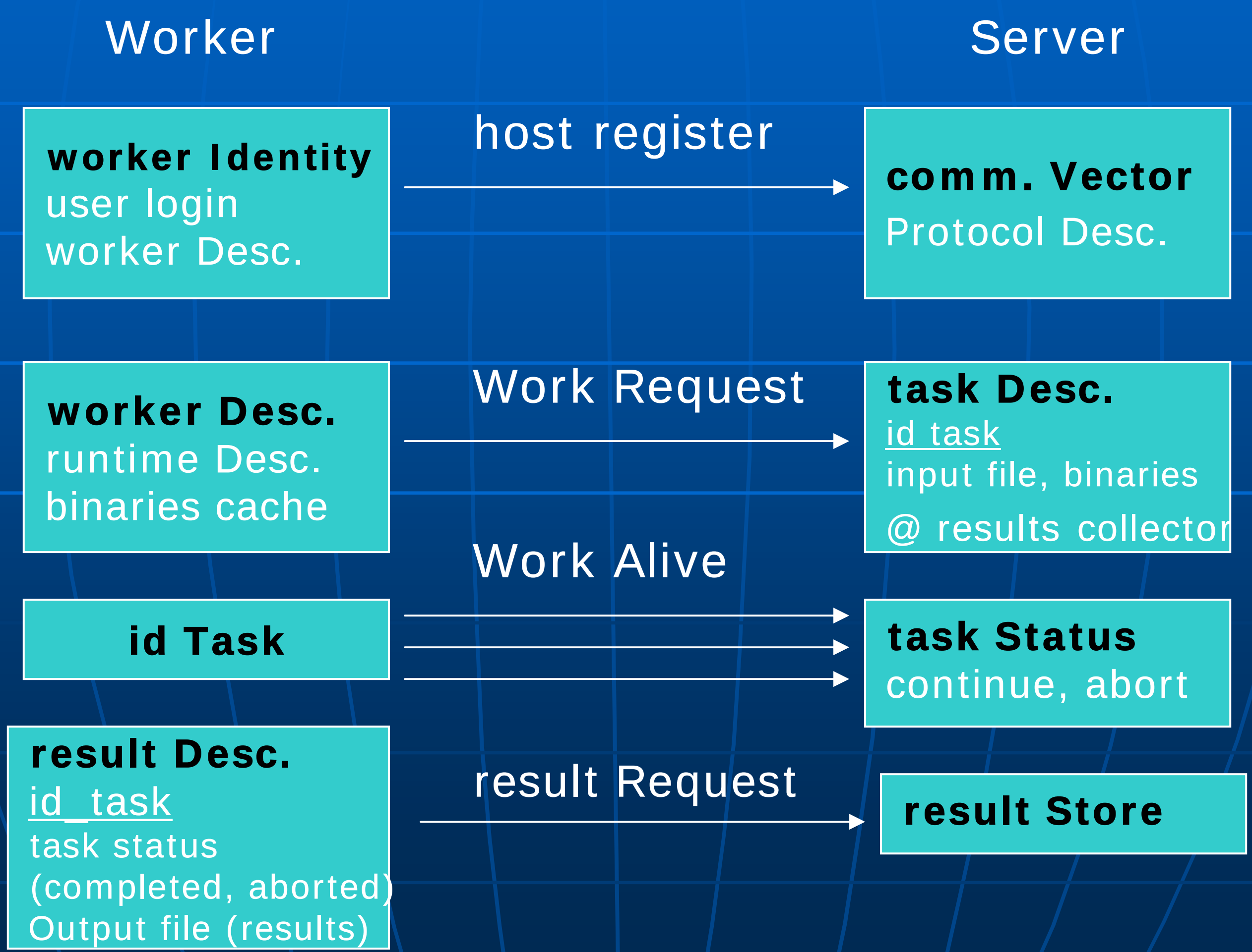
XtremWeb

- Security and Native Code Execution
  - Native code execution -> High Performance
  - Ad-hoc verification process
    1. Only trusted institutions can propose code
    2. Code is tested on dedicated workers
    3. Code is encrypted before downloading to workers
    4. Code download uses a private-public key to secure transaction
    5. Worker sends back a checksum of code to server which verifies it

XtremWeb

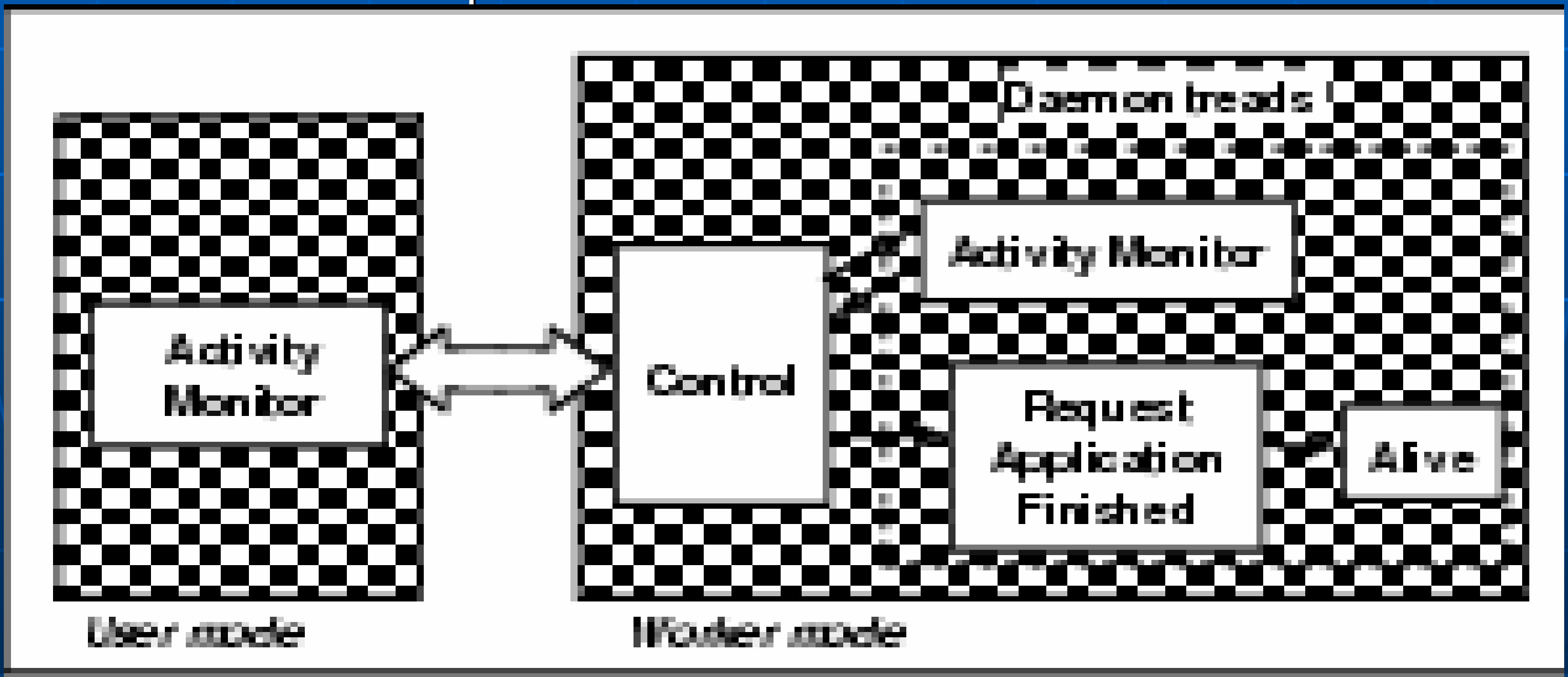
- Communication between Worker and Server
  - Three protocol layers
    1. First level (connection) – Enable connection between entities behind firewall or a proxy
    2. Second level (transport) – Responsible for reliable and secure transport
    3. Third level (protocol) – RPC API

▪ Communication between Worker and Server



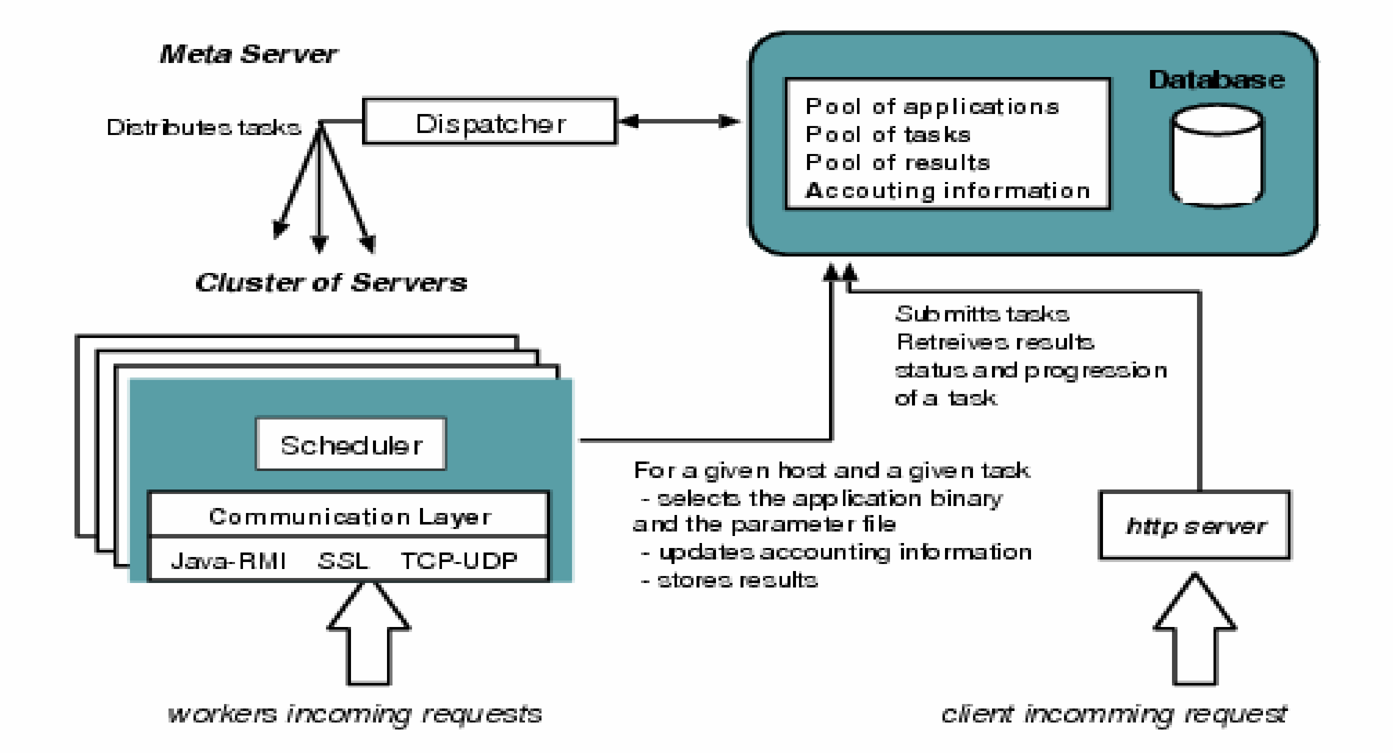
Worker Architecture

- Implementation
  - Java
    - Easily portable
    - Not related to performance but to security and portability
  - Calls to specific OS functionalities - C



Server Architecture

- Pool of Applications
- Pool of Jobs
- Accounting Modules



Server Architecture

- Scheduling
  - Dispatcher
    1. Selects tasks from the task pool and forwards to the scheduler.
    2. Priority is set by minimum ratios of tasks running per application.
  - Scheduler
    1. Tasks are scheduled in a FIFO scheme.
    2. Determines task that may be run by a worker.

Implementation

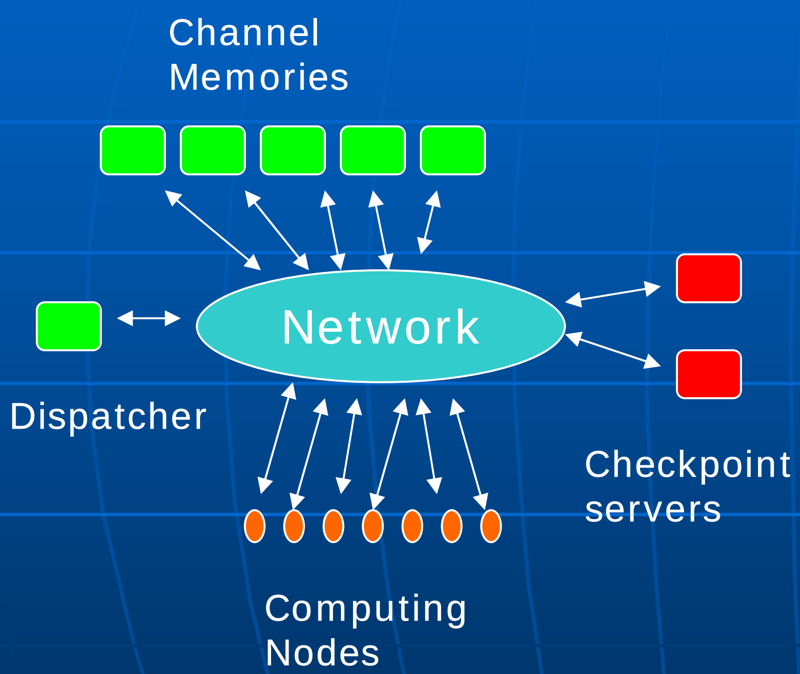
- Java, PHP and Perl
- MySQL database
- Volunteers and Administrators interact though a Web interface

Fault Tolerance

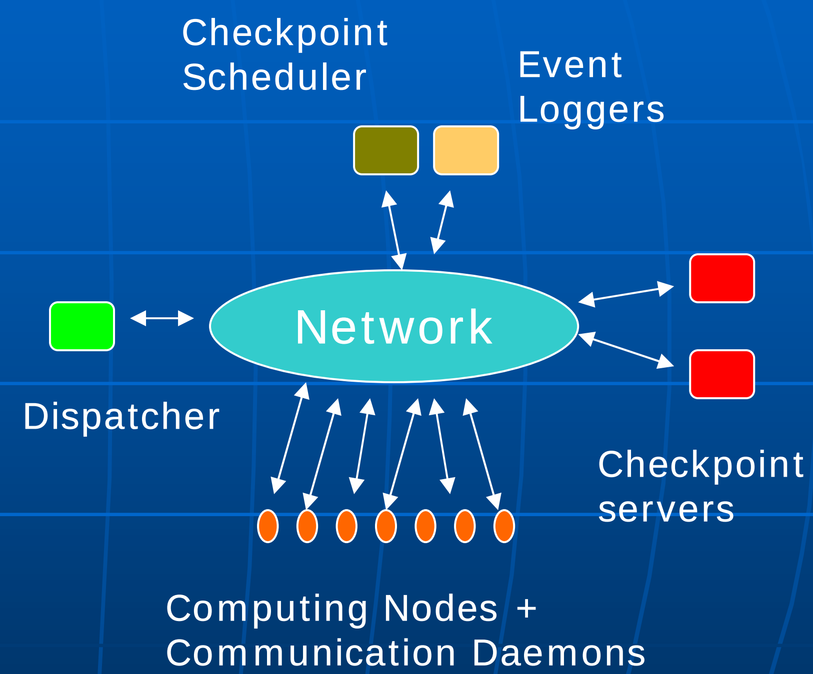
- Uses Fault-Tolerant MPI implementation.
  - Programmer may save results periodically in case of an entire restart
  - MPI hides faults from programmer by using a fully automatic fault detection and recovery.
    - Pessimistic logging principle to tolerate *N* concurrent faults. (*N* = total number of MPI processes)

Fault Tolerance

MPICH-V1



MPICH-V2



Implementation

- Projects implemented
  - AIRES
  - Protein Folding by Mutations

Conclusion

- Global Computing systems are concerned with ease of deployment.
- Fault tolerance is critical since there are external threats
- Threats come from application, data and the computing nodes.

References

[1] Fedak, G., et al. *XtremWeb : A Generic Global Computing System in Computing Cluster and the Grid*. 2001. Brisbane: IEEE

[2] Fedak, G., et al. *Computing on Large Scale Distributed Systems: XtremWeb Architecture, Programming Models, Security, Tests and Convergence with Grid*. Future Generation Computer Systems, 21(3):417-437, March 2005.