



# **Amoeba**

**Vijay Srinivas Nori**

**Feb 17, 2006**

# Key Points

- Overview
- Process Management
- Communication
- File System
- Protection & Security

# Overview

- System Components
  - Workstations (run basic jobs that interact intensively with user)
  - Processor Pool (provides bulk of the computing power, fault tolerance)
  - Specialized Servers (run dedicated tasks with unusual resource demands, e.g. file servers)
  - Gateway (shield local machines from WAN protocol issues)

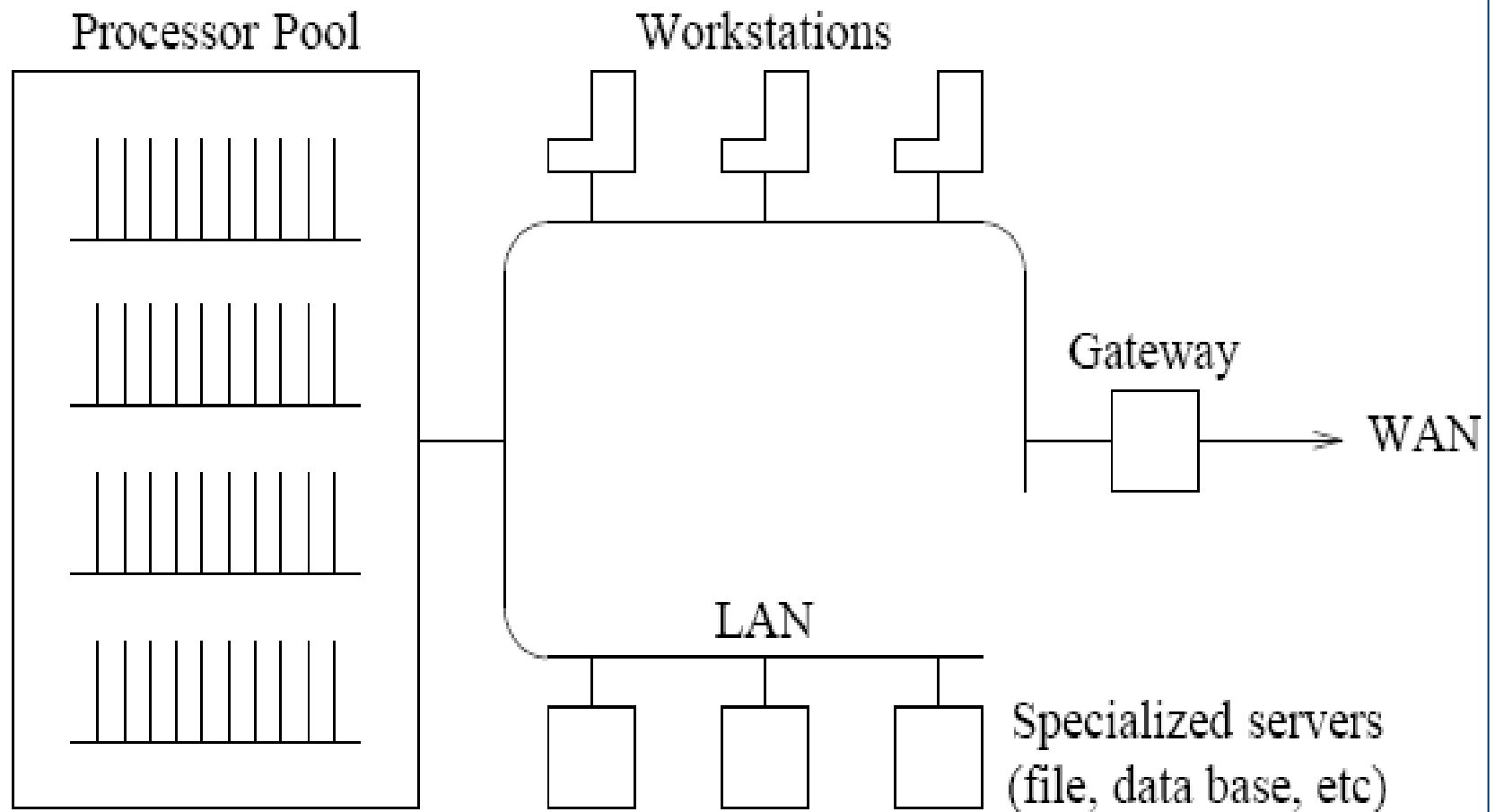


Fig. 1. The four components of the Amoeba architecture.

## Overview (contd.)

- Kernel manages memory segments, multiple threads, handles IPC
- All other services provided by user level processes in accordance with microkernel style design
  - directory service
- Advantage: Flexible system without sacrificing performance

# Process Management

- Processes can be remotely created, destroyed, migrated
- Segmented Memory
  - Map in memory segment specifying capability
  - After unmap, segment remains in memory like a file, can be mapped in by another process
  - Used to do memory mapped I/O

# Process Management (contd.)

- Process Descriptor
  - Which machine can this process run on?
  - Capabilities of process and handler
  - What are the segments mapped in?
  - State of the threads (psw, sp, registers etc)
  - Elaborate: used for debugging, checkpointing
- Process State
  - Running
  - Stunned (debugging, migration)

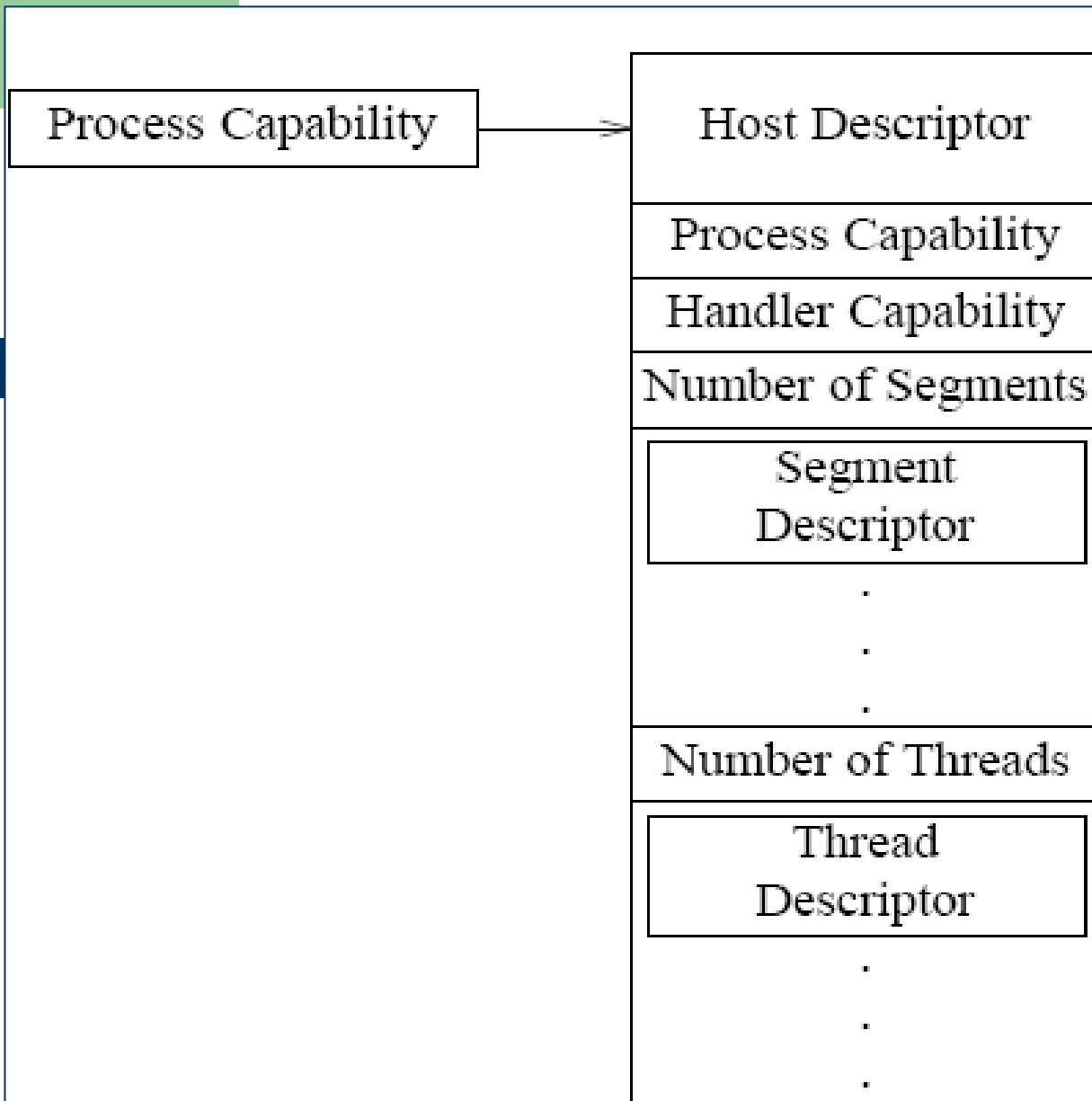


Fig. 6. Layout of a process descriptor.

# Process Management (contd.)

- Advantages:
  - Process Startup Time low
    - Process Descriptors cached
    - Kernels cache code of programs they start
  - [Note] Designed for environment where remote execution is the norm
  - Efficiency necessary for each: execution, migration, debugging, checkpointing

# Communication

- Basic model: Threads using RPCs to server services managing objects
- Basic System Calls
  - do\_operation (client sends msg to server, blocks till reply)
  - get\_request (server states that it will accept messages to a certain port)
  - send\_reply (server replies to client request)
- **AIL** defined to provide application programmers with a more high-level view

# Communication (contd.)

- RPC Transport
  - AIL compiler automatically generates code to process parameters and generate stubs of RPCs from operation descriptions
  - Then calls Amoeba's transport mechanism
  - Message
    - Header: capability, opcode
    - Buffer: Data or arguments to the opcode function
- Advantage: Efficient because data can be transmitted direct from and to arguments specified in the program

# Secure Communication

- Ports protect access to servers, capabilities protect access to individual objects
- Ports (except public ports) are secret
- Knowing the port number = right to communicate
- Q: How do you ensure that a malicious process doesn't execute `get-request` calls trying to impersonate a server?

# Secure Communication: Solution

- Either in Hardware or Software:
  - implements a one-way function, an f-box
  - Ensure all messages leaving the system go through this f-box
  - Strength of the one-way function crucial
- Advantages:
  - Flexible, system designers can design their own f-function, OSs in the future also just need to redefine f-box.

# File System

- Hierarchical Directory Structure

- Directories = objects with access capabilities
- Directory Service maps from ascii to capabilities
- Complex sharing: use  $(n+1)$  columns for access capabilities
- Reliability: we can create  $n$  copies/file and distribute to remote servers

- Bullet Service File Server

- Only supports read\_file, create\_file, delete\_file
- Immutable**, file once created cannot change

# File System (contd.)

- File size fixed, so files stored **contiguously**
  - Minimal information: ownership, capabilities, origin
  - Read is very fast, since it is contiguous
- Disadvantage: Fragmentation
- Advantage:
  - Performance
  - Reliable, centralized replication mechanism – directory service
  - Immutable files, **easily cached** (never stale!)

# Bullet server memory

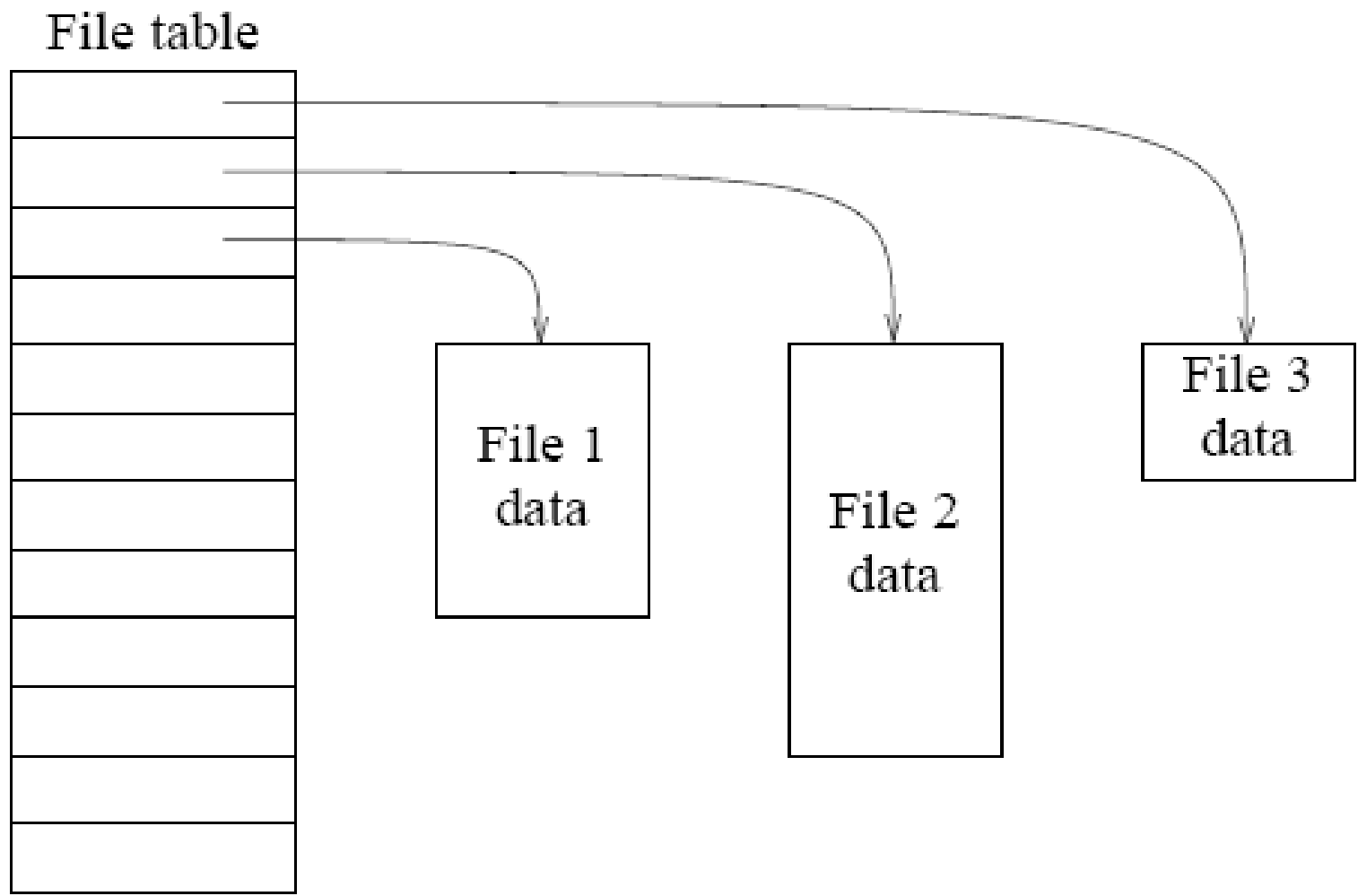


Fig. 5. Bullet Server file representation.

Figure 3: Ref: [1]

# Take Home Points

- Process Management, capabilities, segmented memory, low startup time
- Novel File System, immutable contiguous files
- Lightweight RPC based secure communication
- Cryptographic protocols, immutable filesystem, object oriented design principles enhance security

# References

- [1] Amoeba, A Distributed OS for the 1990s, S.J. Mullender, A.Tanenbaum et.al.  
<http://citeseer.ist.psu.edu/mullender90amoeba.html>
- [2] <http://www.cs.vu.nl/pub/amoeba/>