Chapter 2

Design Issues

Part 2.3

Above Message Passing

Additional structure above message sending:

- remote procedure call
- client-server
- multicast

Remote Procedure Call (RPC)

- Invented by Bruce Jay Nelson
- idea:
  - hide message communication by a layer which simulates a procedure call!
  - Familiar, simple semantics
- The catch:
  - semantics are not Familiar or simple
  - see Chapter 5

Client-server:

- problem: server blocked time must be
  MINIMAL, so

- server can be built as *administrator*
  plus n *workers*
- workers upon instantiation do a blocking send to admin with block on reply

Administrator model

- hence they are waiting . . . So,
- admin accepts a work request and REPLIES to a

  a blocked worker with the job to be done
- worker unblocks & does the work

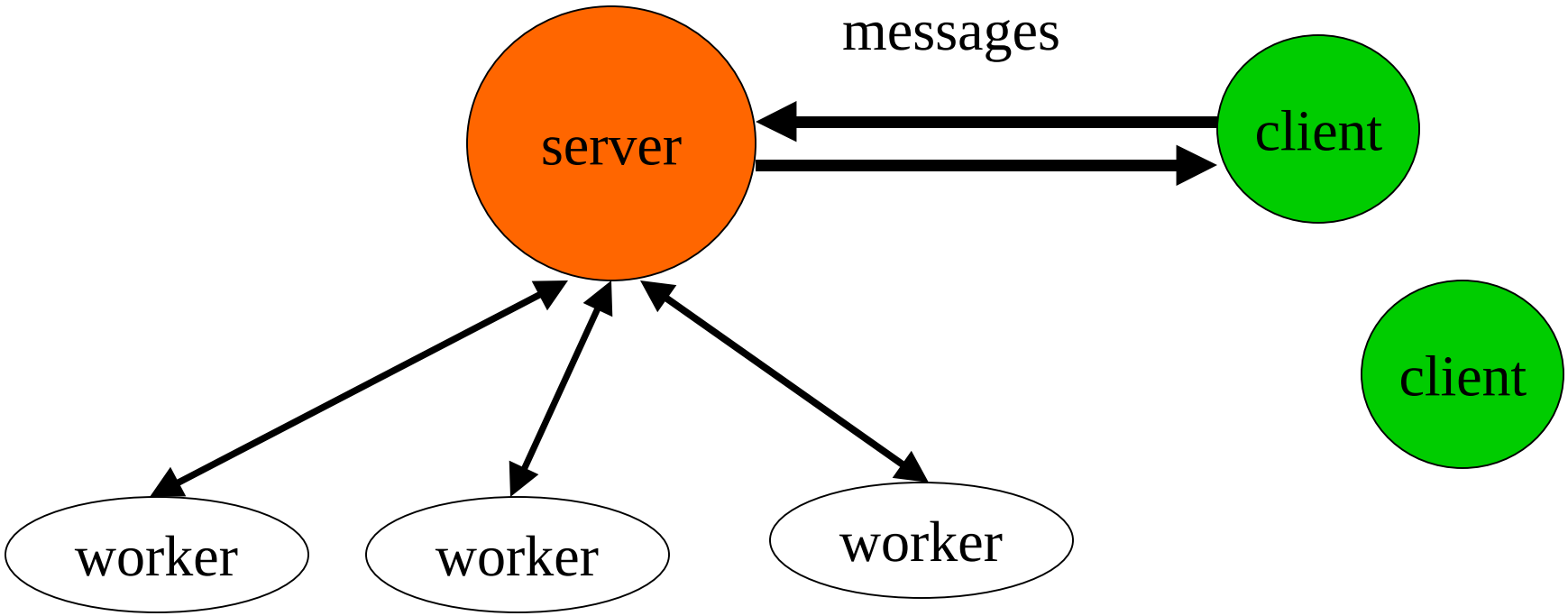
Administrator model

- admin returns at once to checking input queue or waiting on client requests
- worker when done repeats the send to server (with result) and blocks again on reply from administrator
- admin sends to client . . .

client-server binding

- cannot be done at writing time
- clients unanticipated at server coding time
- must be allowed to use the server.
- So . . .

Client-server (administrator)



client-server binding

- server upon creation registers w name service
  using well-known service name
  (e.g., print_service)

- clients get the name translated at runtime &
  thus can communicate with the server

Group multicast

- **WHY?**

Group multicast - why?

- locate an object by multicast (ARP)
- fault tolerance: multicast of an idempotent operation request to a set of servers
- update of replicated databases

Distributed OS structure:

Kernel plus servers model

DOS structure

- not a single lump of code

(Unix syscall ifce & OS) but

- an extensible set of servers
  - each with its own interface (arggh!) and
  - all using *kernel services* such as

DOS structure: kernel services

- memory allocation & protection
- process management
  (create, destroy, schedule)
- ipc
- device handlers (maybe not)

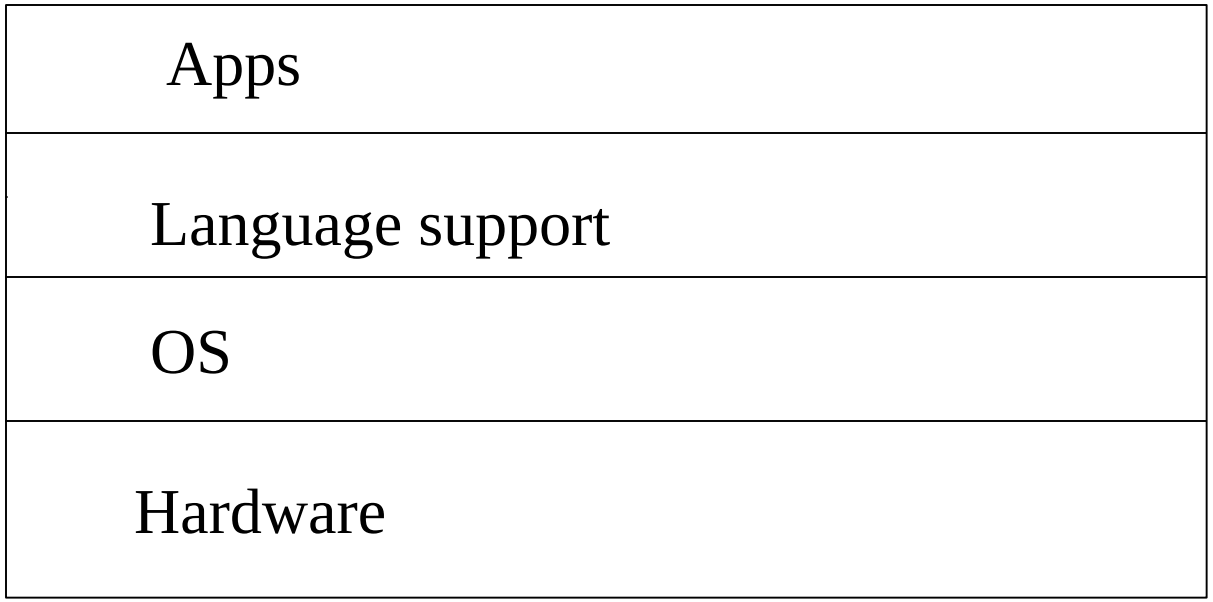
- all provided by a protected
  micro (smaller than unix's) kernel which
  abstracts the hardware

Multiserver DOS structure

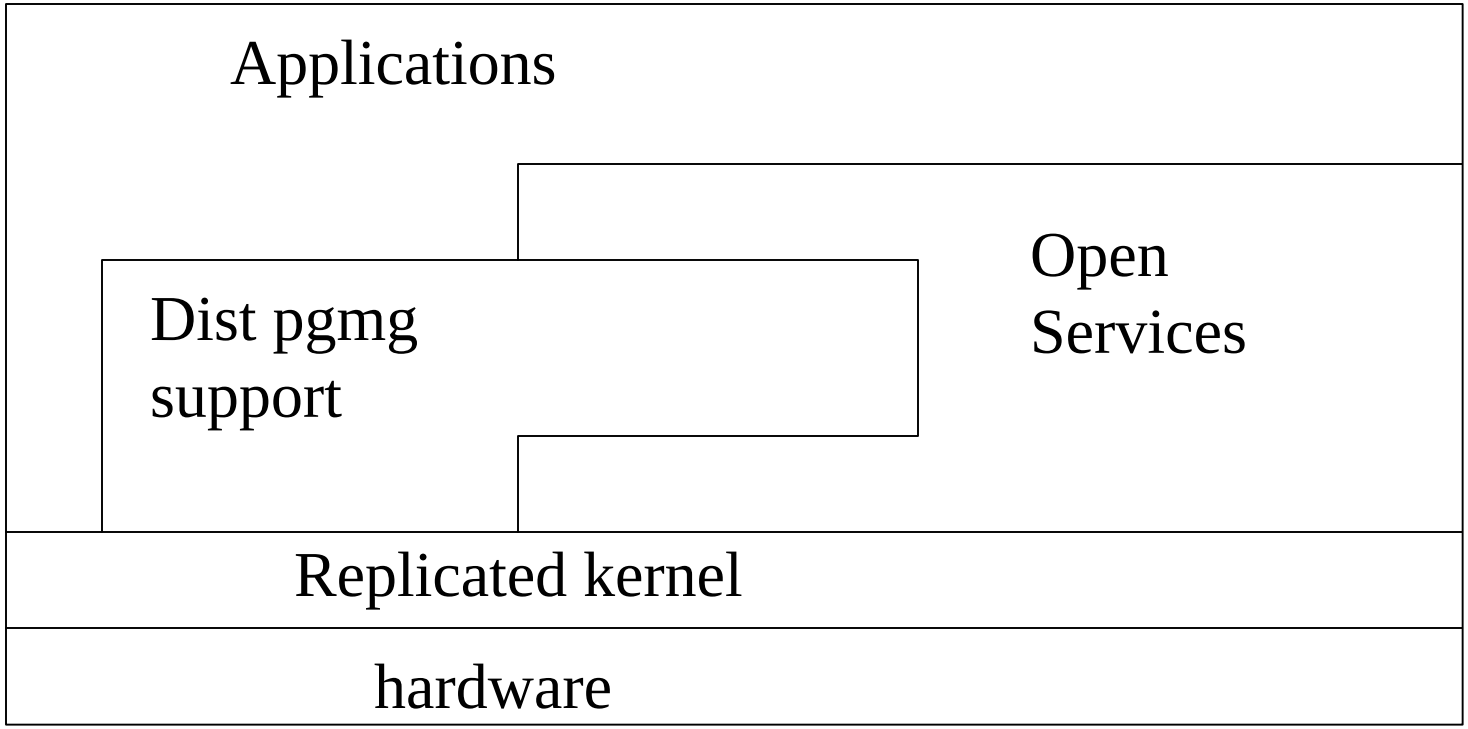
- plus servers providing:
  - peripheral device service (printers, archival store)
  - filesystem
  - timing services
  - security
  - name translation
  - email . . .
- Everything an OS did, and more .

Summary

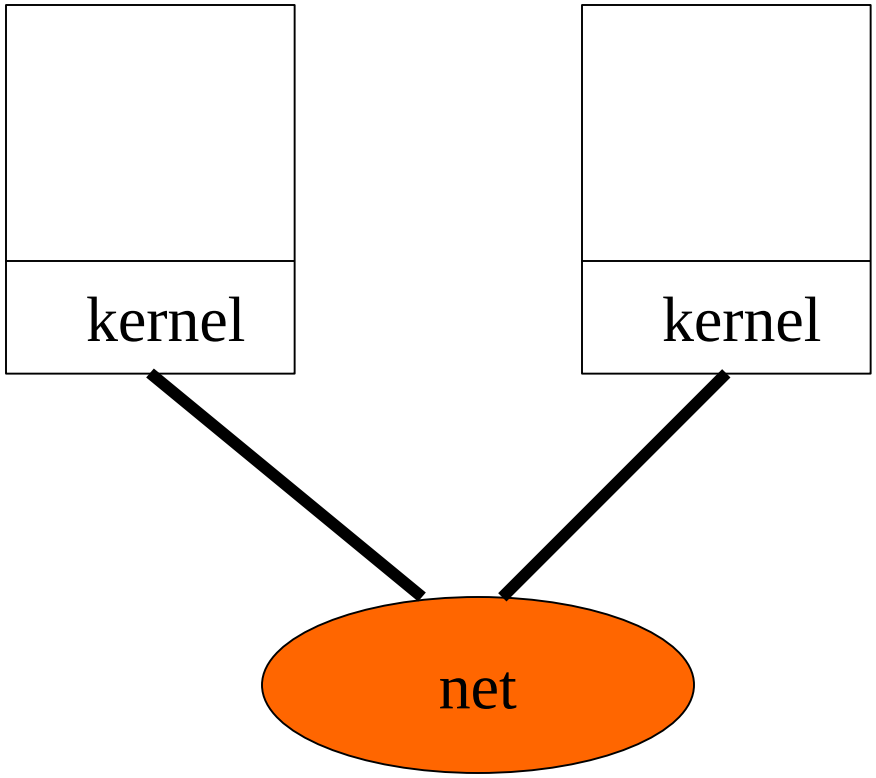
- not



- but


Applications
Open
Services
Dist pgmg
support
Replicated kernel
hardware

- Realized as



Allocating Workload