

Distributed Computing

Partitioning the Work

- Speedup
- Functional partitioning
- Data partitioning
- Supporting partitioned computations
 - Creation/destruction of network processes
 - Scheduling
 - Synchronization
 - Deadlock management

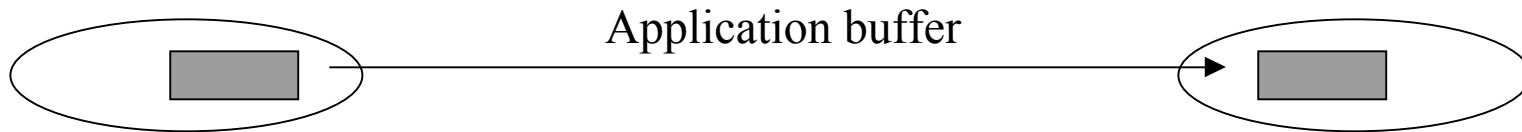
Scheduling

- Threads and processes
- Explicit scheduling
- Transparent scheduling
- Migration & load balancing

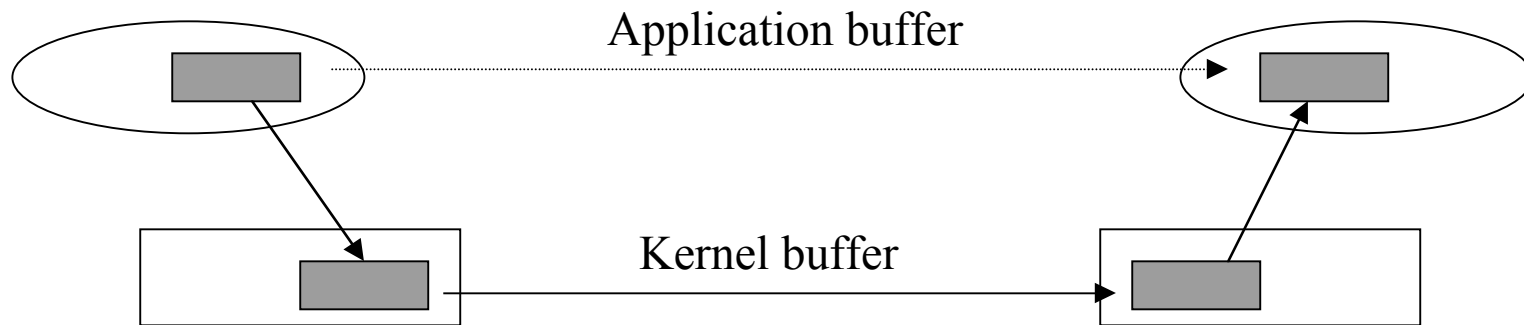
Coordinating the Parts

- Distributed synchronization
 - No shared memory \Rightarrow no semaphores
 - New approaches use logical clocks & event ordering
- Transactions
 - Became a mature technology in DBMS
 - Multiple operations with a commit or abort
- Concurrency control
 - Two-phase locking

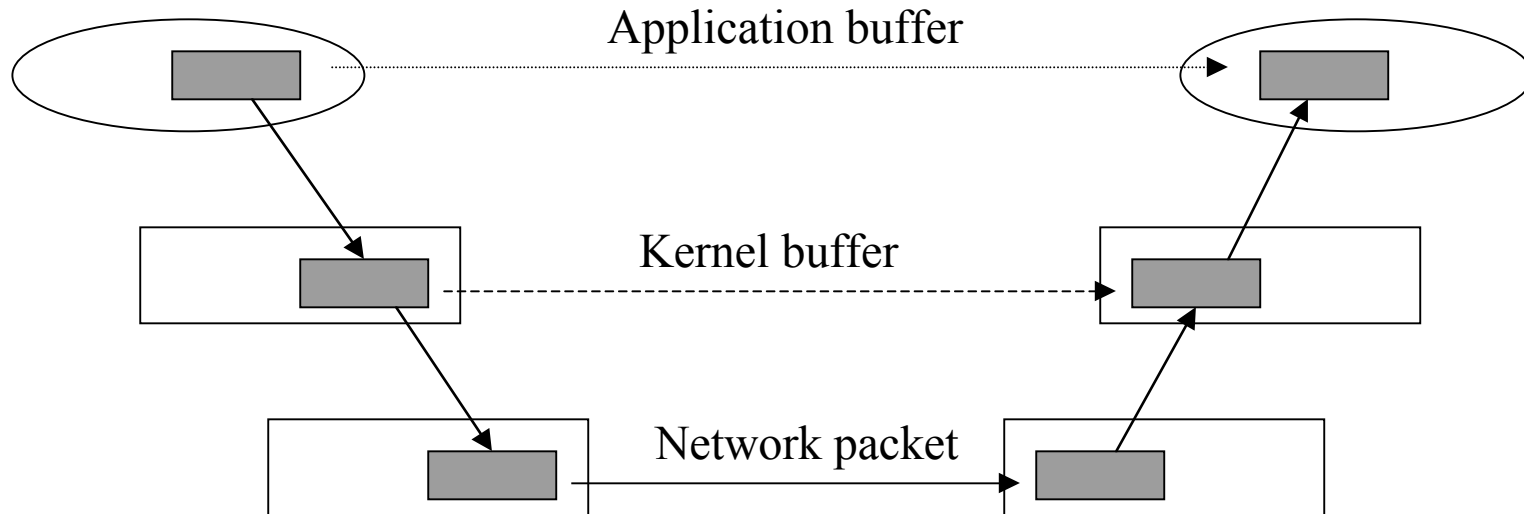
Message Passing



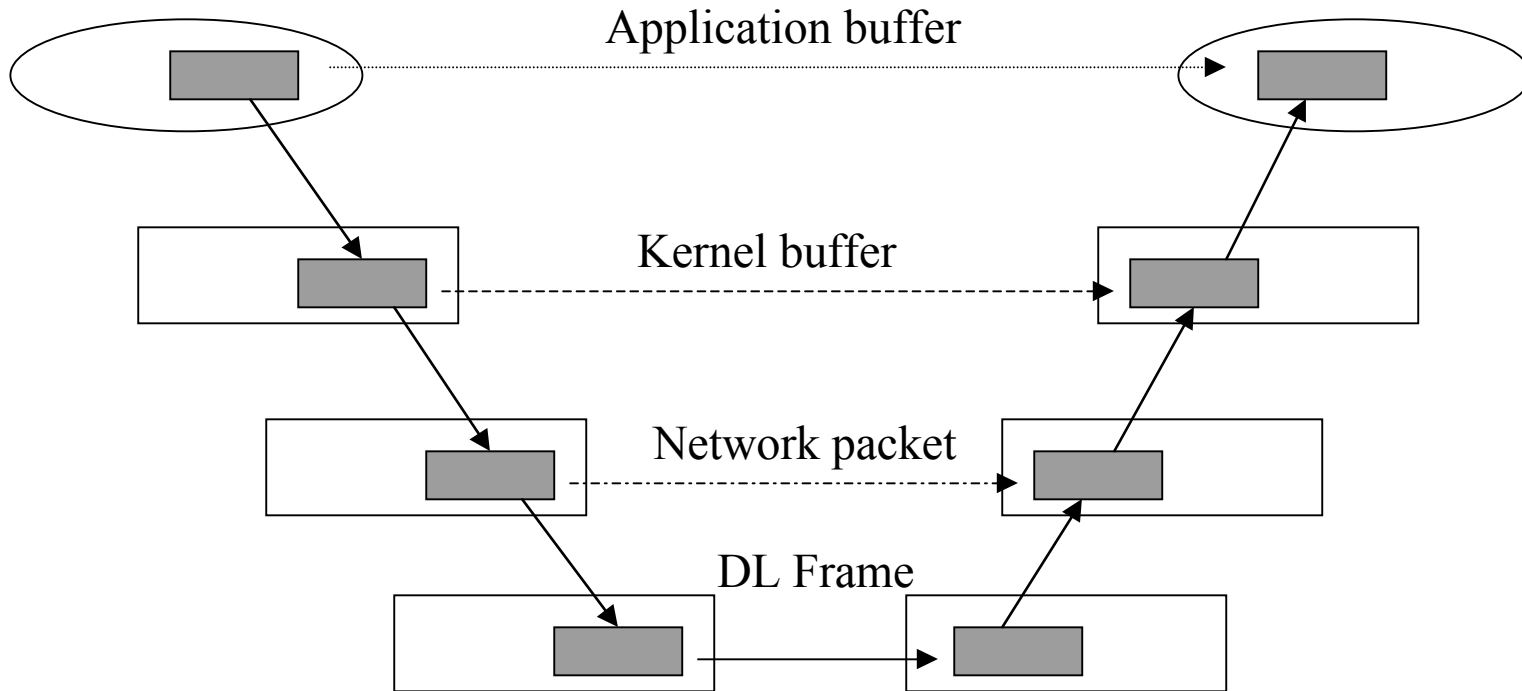
Message Passing



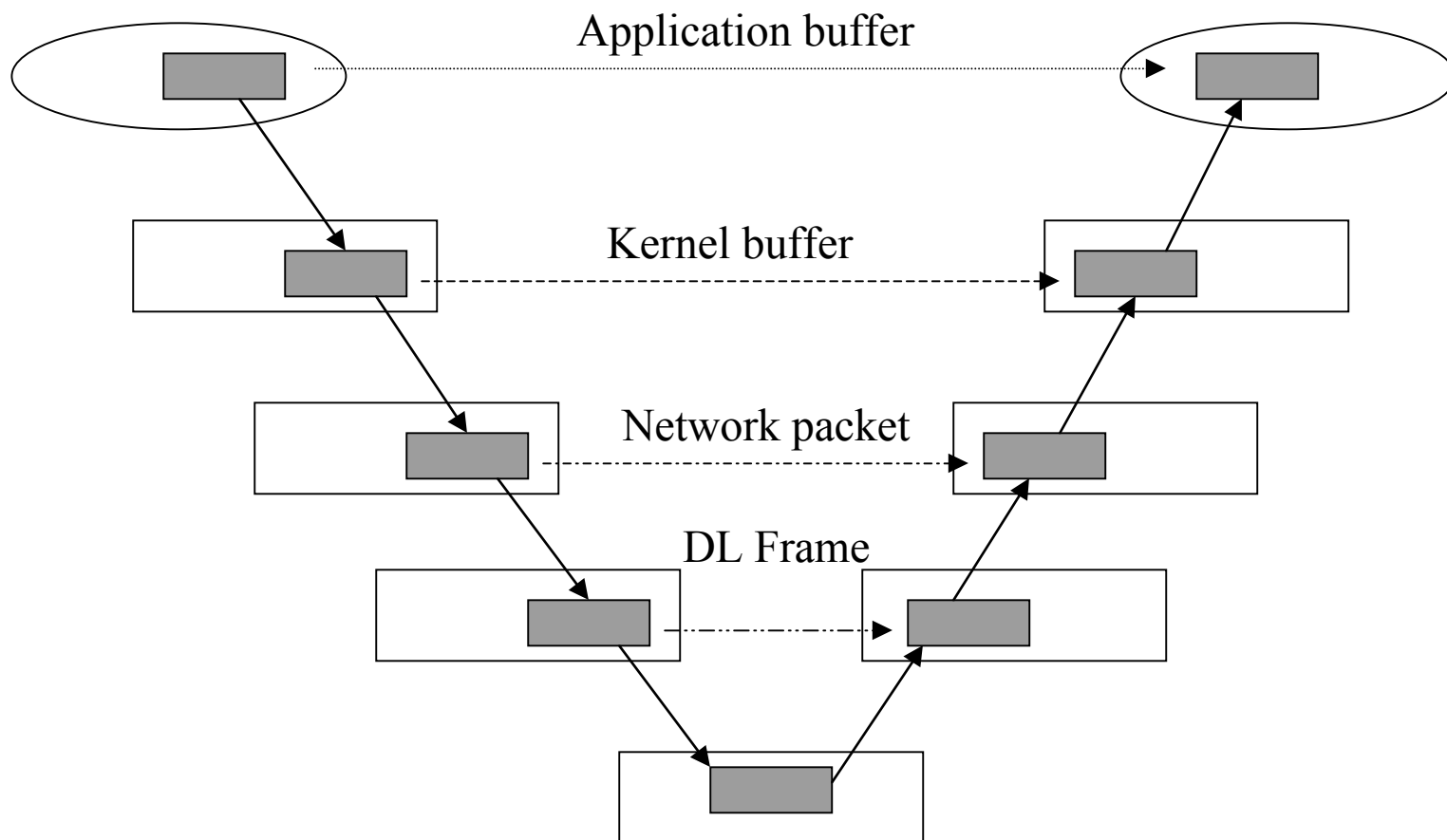
Message Passing



Message Passing

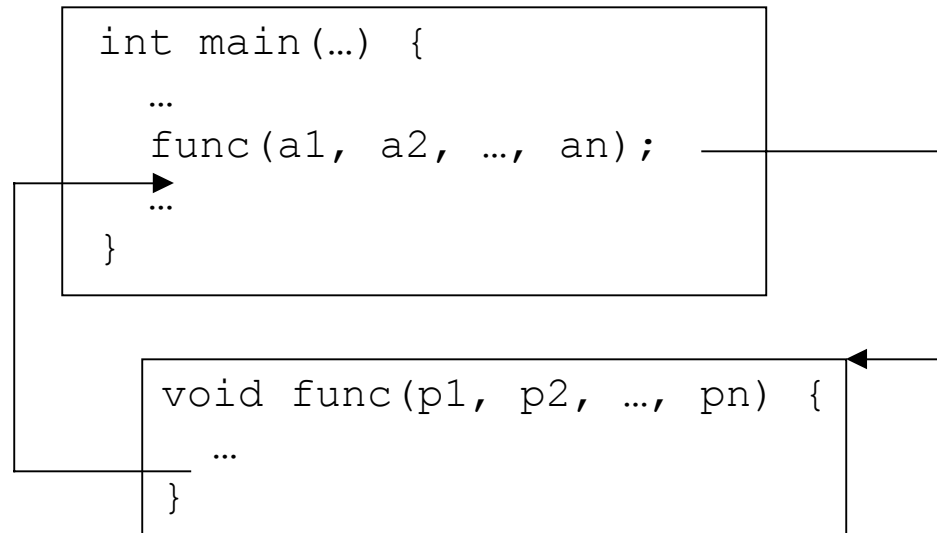


Message Passing

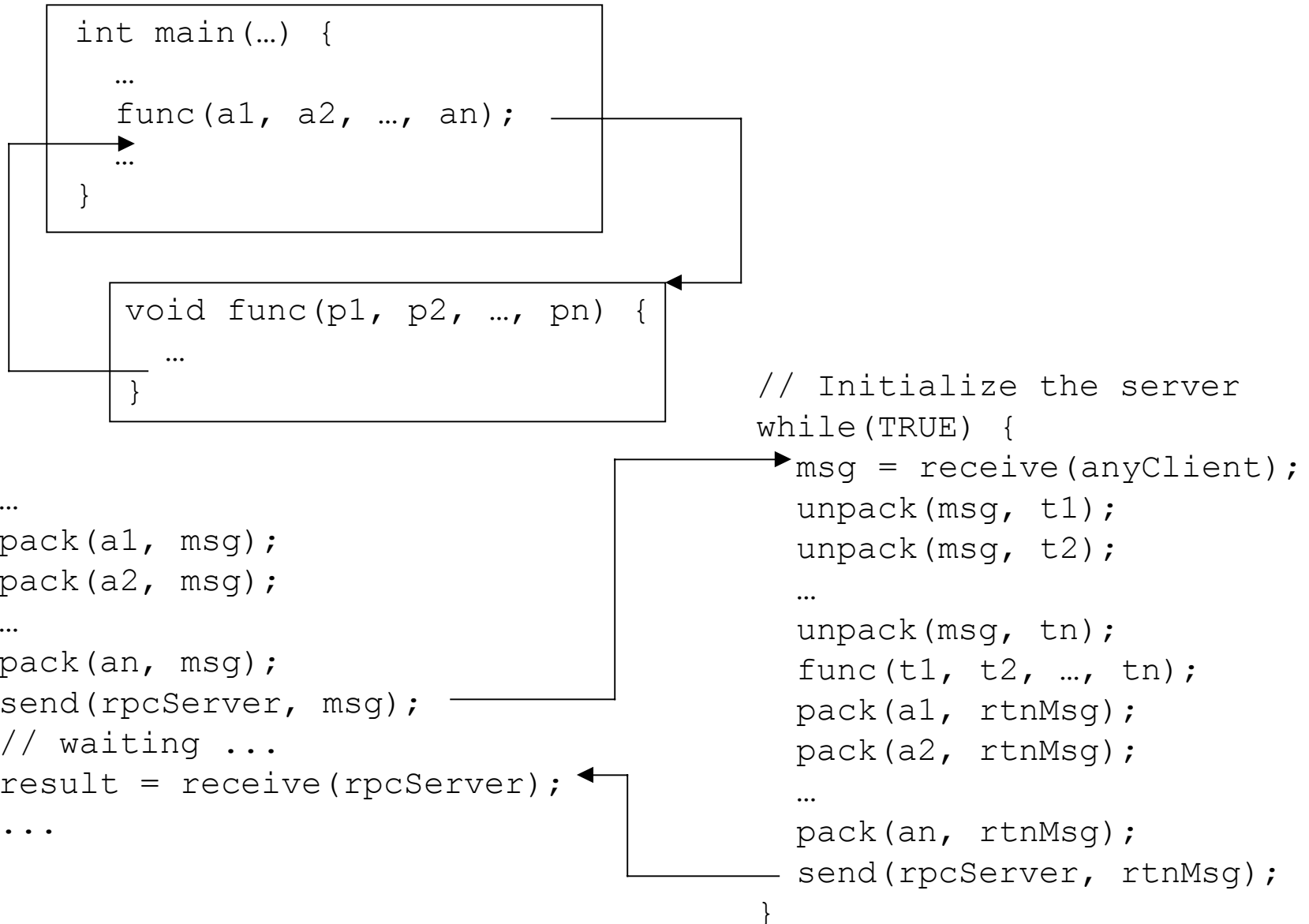


Remote Procedure Call

```
int main(...) {  
    ...  
    func(a1, a2, ..., an);  
    ...  
}  
  
void func(p1, p2, ..., pn) {  
    ...  
}
```



Remote Procedure Call



Remote Procedure Call

```
int main(...) {  
    ...  
    func(a1, a2, ..., an);  
    ...  
}
```

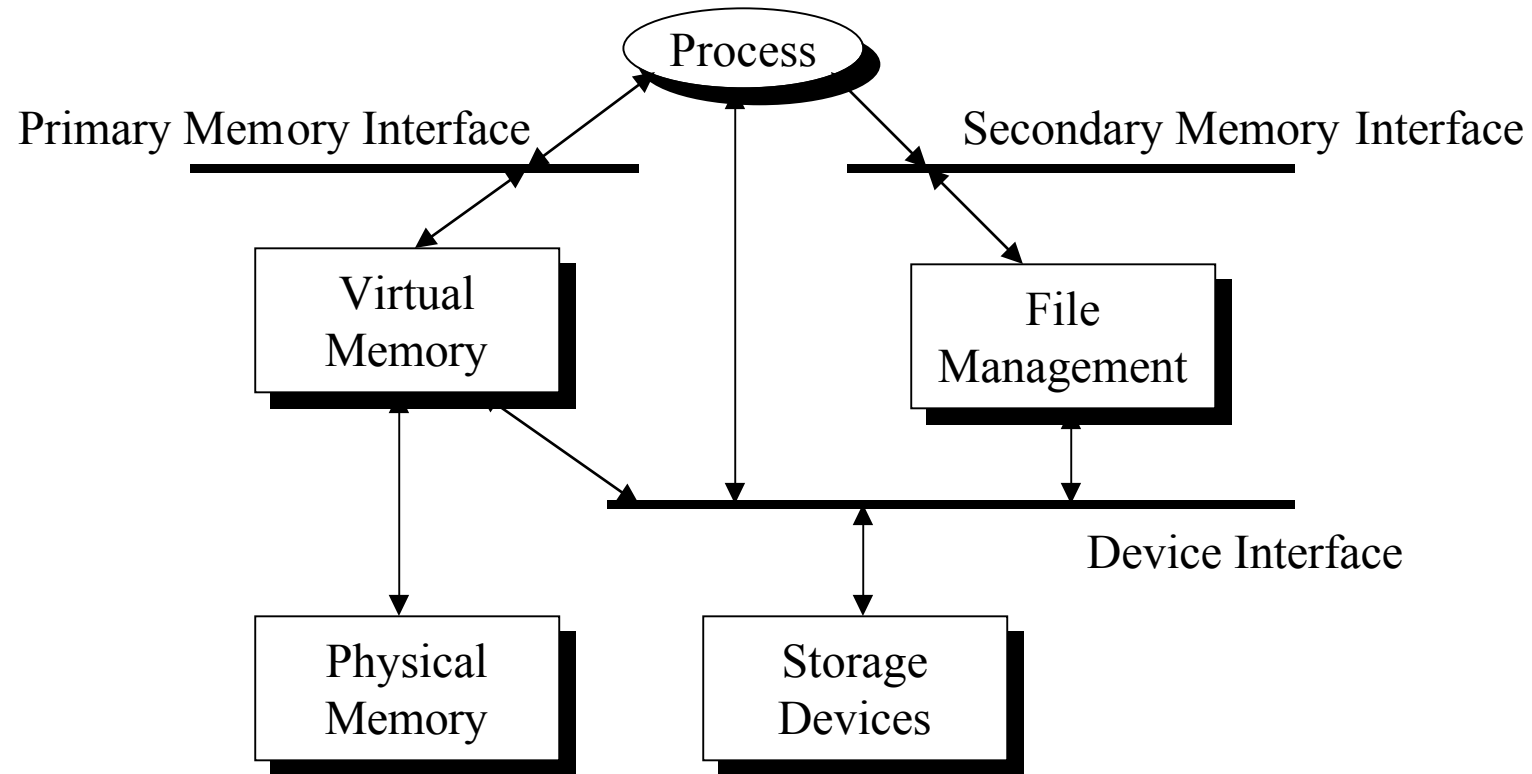
```
void func(p1, p2, ..., pn) {  
    ...  
}
```

```
...  
pack(a1, msg);  
pack(a2, msg);  
...  
pack(an, msg);  
send(rpcServer, msg);  
// waiting ...  
result = receive(rpcServer);  
...
```

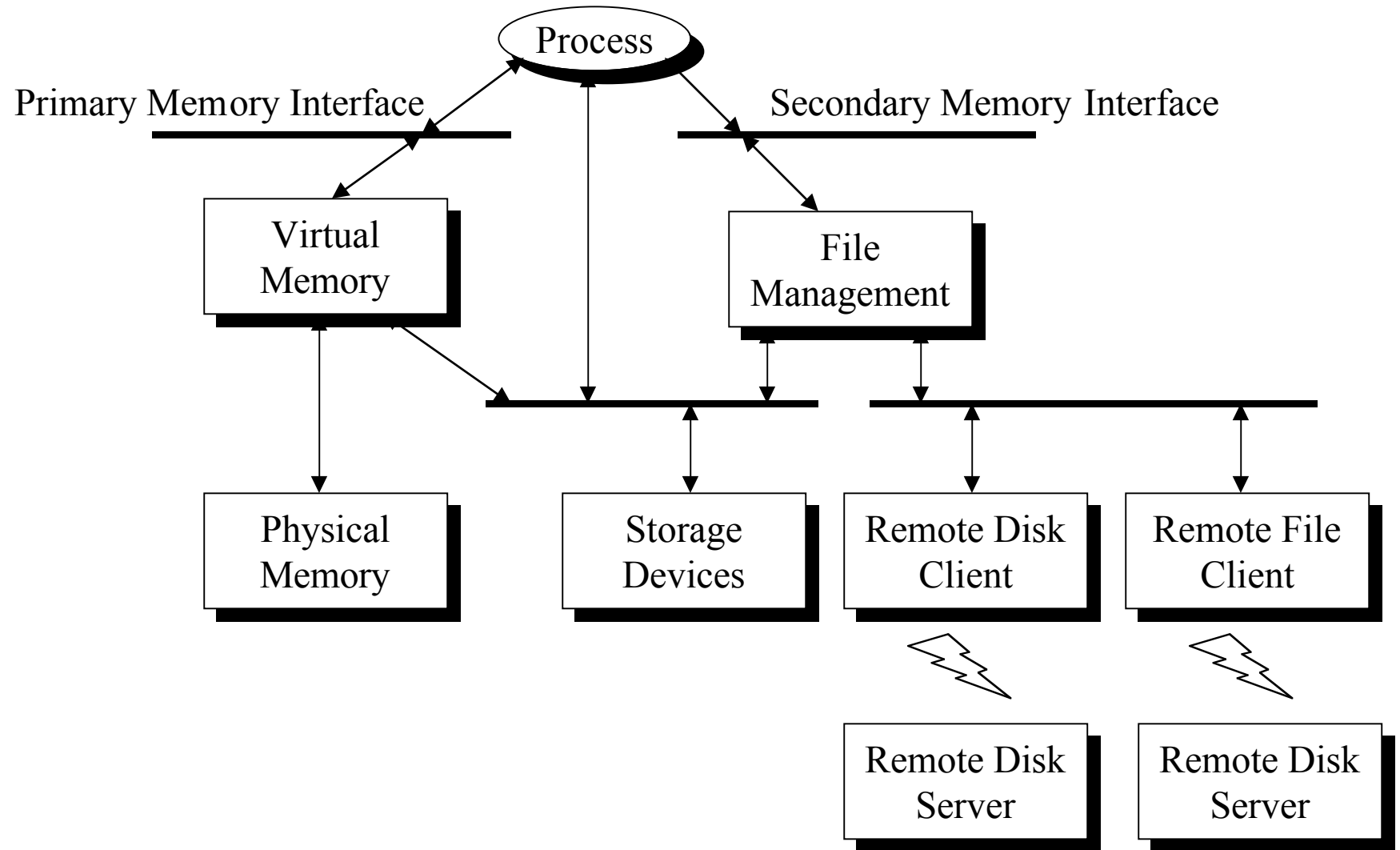
```
// Initialize the server  
while(TRUE) {  
    msg = receive(anyClient);  
    unpack(msg, t1);  
    unpack(msg, t2);  
    ...  
    unpack(msg, tn);  
    func(t1, t2, ..., tn);  
    pack(a1, rtnMsg);  
    pack(a2, rtnMsg);  
    ...  
    pack(an, rtnMsg);  
    send(rpcServer, rtnMsg);  
}
```

READ about
implementing RPC
Section 17.3.2

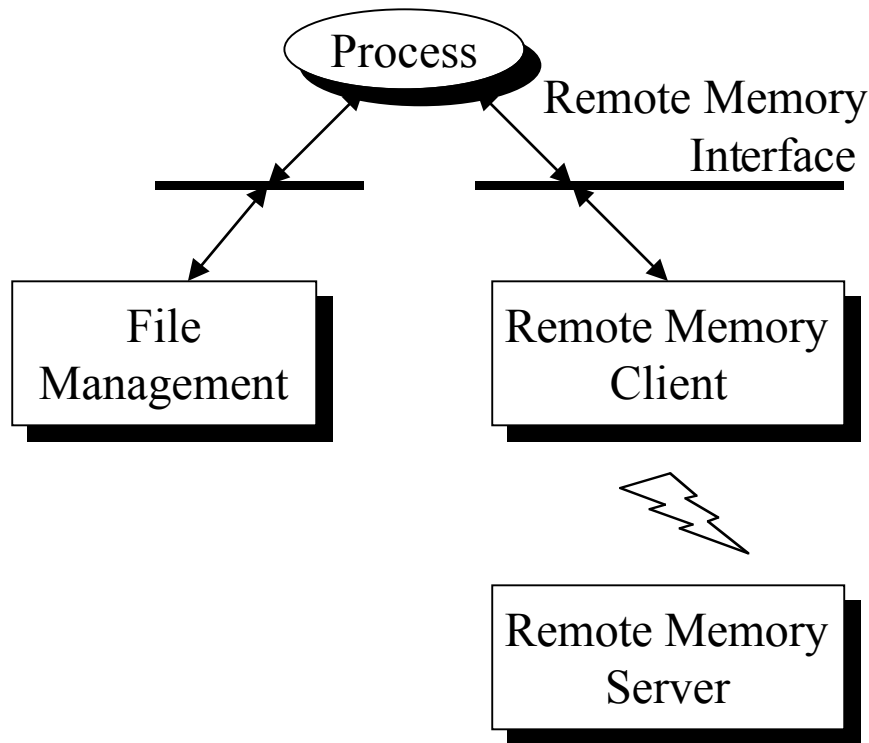
Traditional Memory Interfaces



Remote File Services

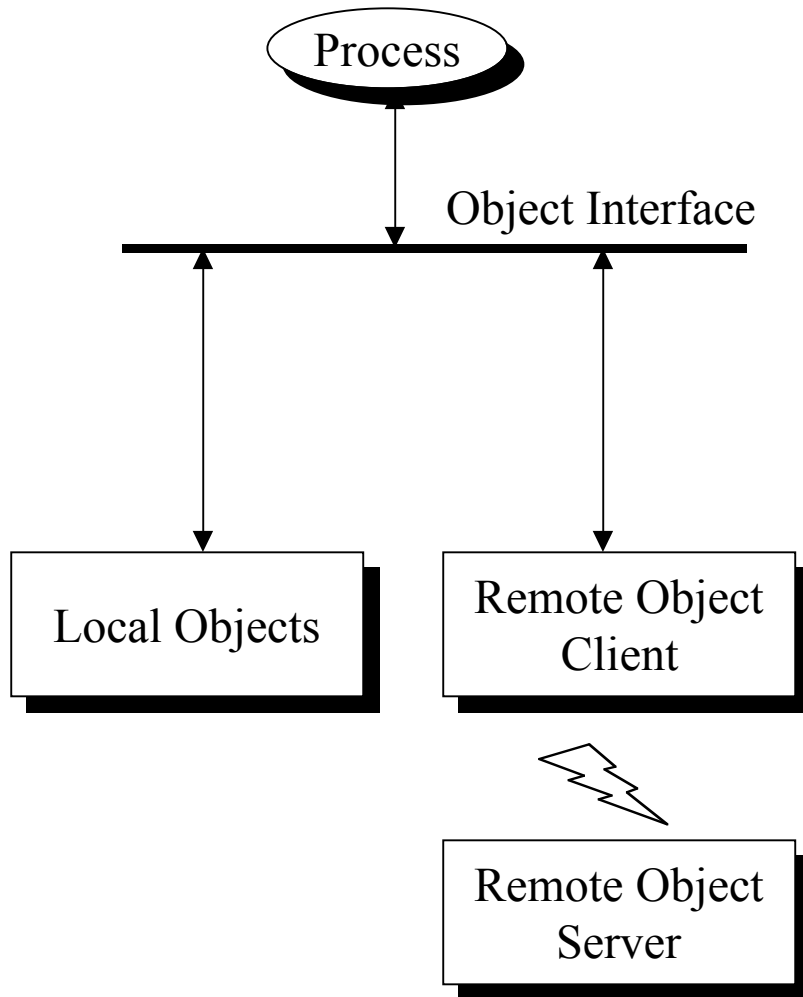


Remote Memory Services

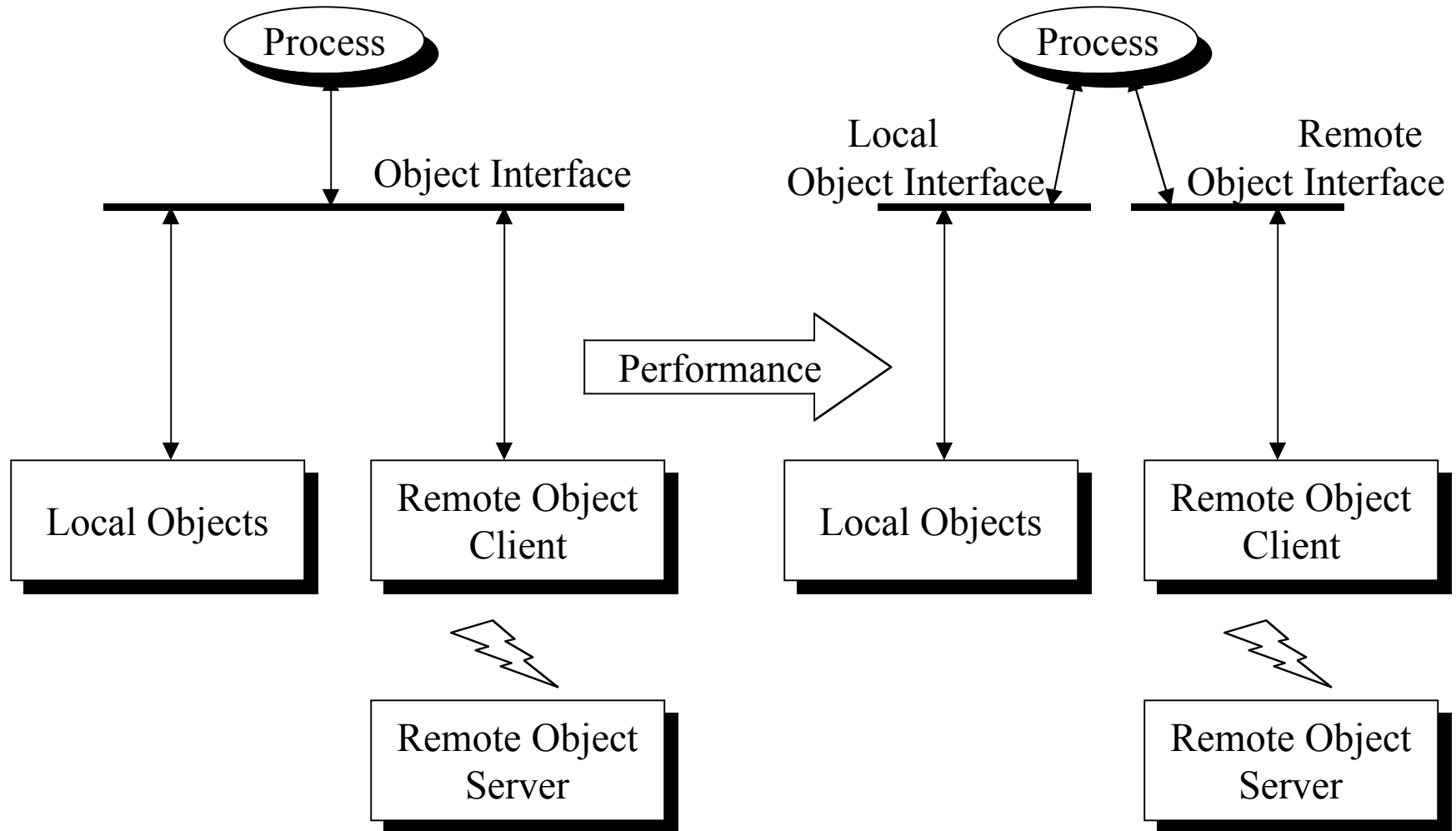


- Static memory \Rightarrow New language
- Dynamic memory \Rightarrow New OS interface
- Low level interface
 - Binding across address spaces
 - Shared memory malloc
- High level interface
 - Tuples
 - Objects

Distributed Objects



Distributed Objects



Distributed Virtual Memory

