Reliability (fault tolerance)

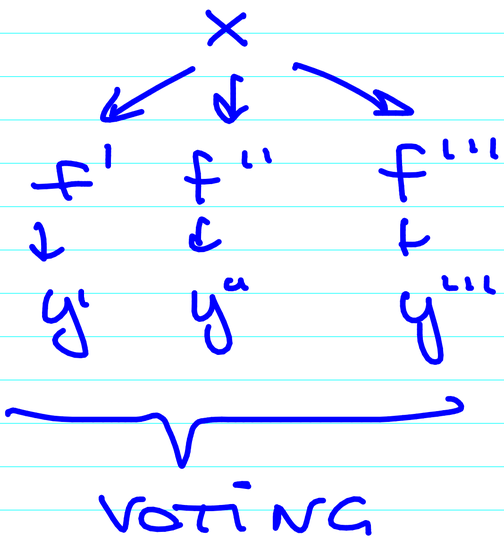
MTBF medium time between failure

ad hoc techniques

① redundancy

RAID

MISD



data flow

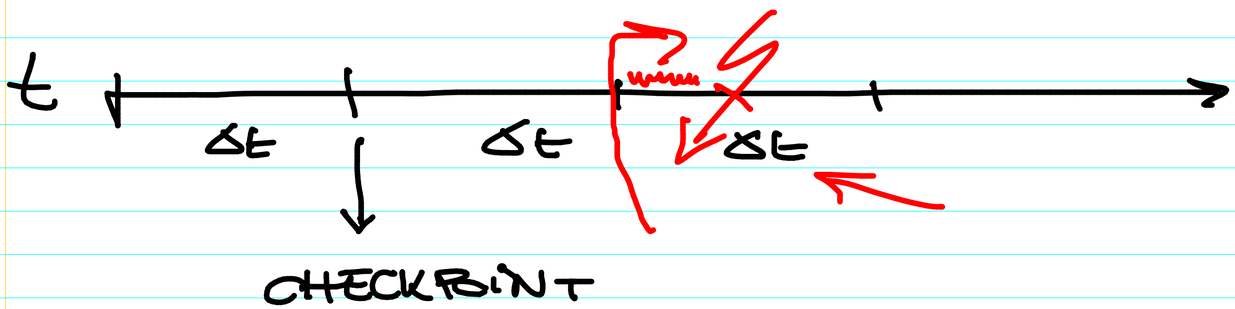
flussi di istruzioni

?	?
SISD	MISD
SIMD	MIMD

VECTOR processing

COW NOW

② CHECKPOINTING



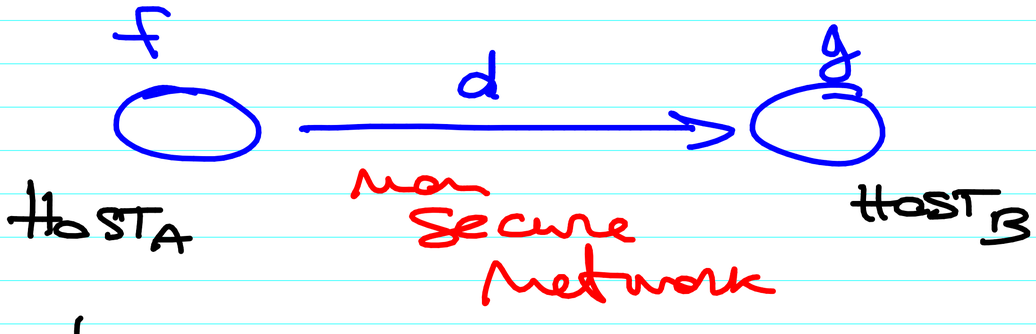
COMPUTATION + STATE

↓↓

DISK

Security
- confidentiality
- authentication

program code
data
access
diritti

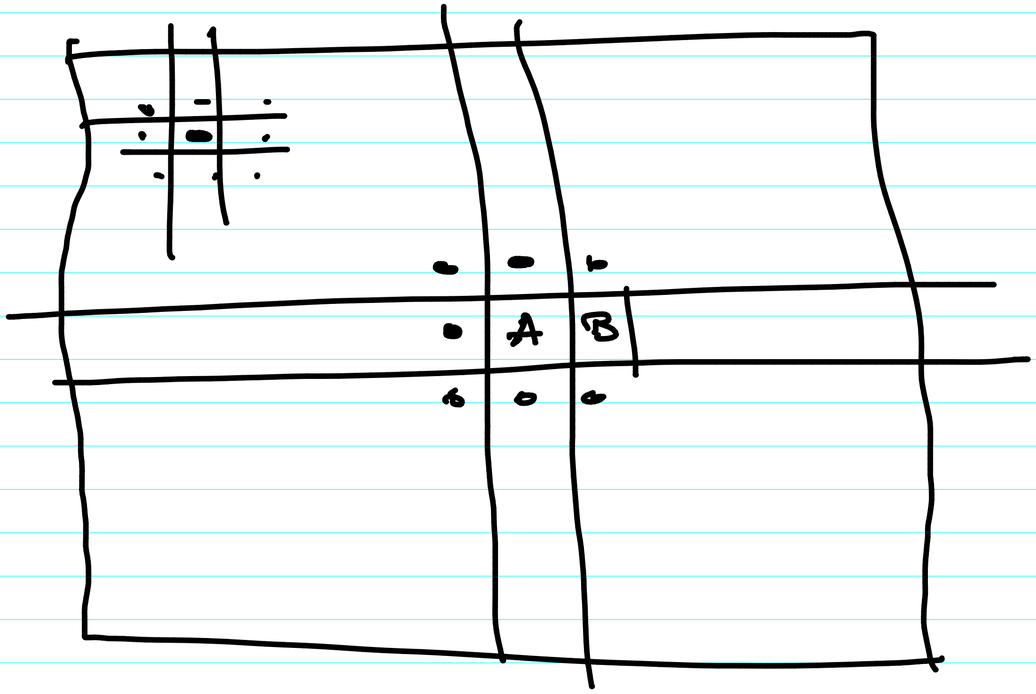


$T_{comm} = t_{setup} + \frac{d}{B}$

non sicuro

msec

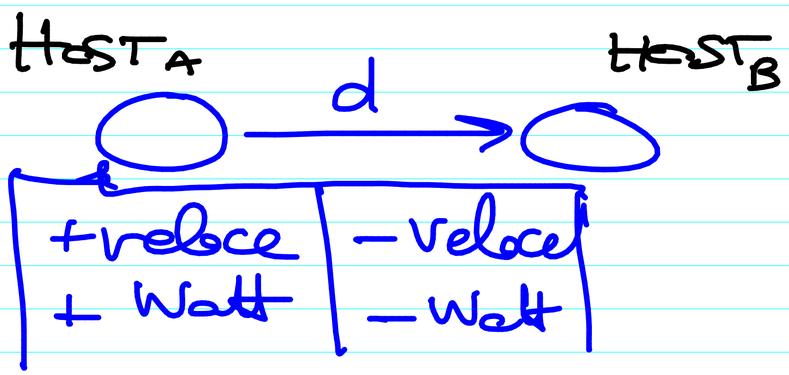
100 Mbit/sec



Power management

CPU GPU

data parallel



create conc. activity ‖ communication ‖ synchro

↳ POSIX / TCP/IP

pid' r = fork() (pid) exec

CREAZIONE ATTIVITÀ → PROCESSI / THREAD

pid / pid' ≠ pid

```
main( ... ) {
   ... x=123;
   int pid = fork();
   if (pid == 0) { // figlio
      g(x) }
   else { // padre
      f(x) };
```

nome del prog

```
execlp("/home/marcod/bin/a.out",
       "/home/baxando/bin/a.out",
       "123", "_a", NULL);

int p[2];
pipe(int *)
```

ssh indirizzo IP cmd

```
> ssh 131.114.2.17 prog1 &
> ssh 131.114.2.18 prog2 &
```

#P #P

THREAD

pthread_create (&tid, &attr, (void *) f(void *), param)

```
int x;
main( ) { ...
  ...
  pthread_create( ..f.. );
  x= 123;
  ...
```

```
f( ) {
  ...
  x=456;
  ...
}
```

v int
pthread_create(pthread_t *restrict thread,
const pthread_attr_t *restrict attr, void *(*start_routine)(void *),
void *restrict arg);

Socket (TCP)

{ ip = "x.y.z.t"
port = 9000

fork

padre } { figlio

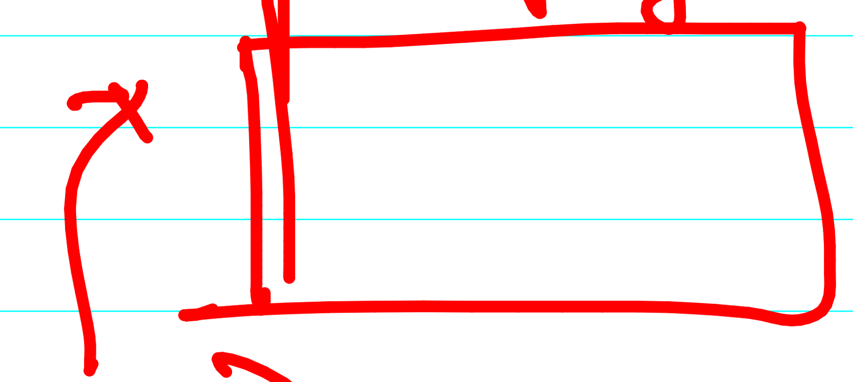
server socket
9000
accept ← connect

read ← write
write → read

pipe (fd[2])

unidirezionale

mem. phy



SYS ✓

- shmem * = attach (mem)
  detach
- msg — send, receive
- sem — post, wait