P1

DAD

CREA 1 FIGLIO, e se esce con un errore RIGENERA IL FIGLIO, altrimenti in caso di corretta terminazione da parte di un amministratore, termina

CREATE

P2

WEB SERVER

S.O.

OVR

NCS

IP & PORT

ACCEPTER

CREATE

ELEMENT

CODA

MAX CON

meta slot della connessioni

GESTISCONO le OP RICHIESTE

CORE ELAB. SYS.

THREADS

THREAD lavorano in maniera esclusiva per la DURATA dello svolgimento dell'OP

STATO e CAMBIO CON

WEB HTML

HTML FILES

IRSC

IMAGE ELABORATION

CACHE_IMG

FILE-SYS

FILE shredder

THREAD LOW PRIORITY & TIMING x log IMG

DB PIPE

SAT OUT

ERR PIPE

LOG SYS

TERMINALE FILE LOG

FLUSSO THREAD

SERVING

ELABORA RICHIESTA

1 HOME

2 MOVE ON SITE

3 RICEVUTA RICHIESTA Elabora RISP

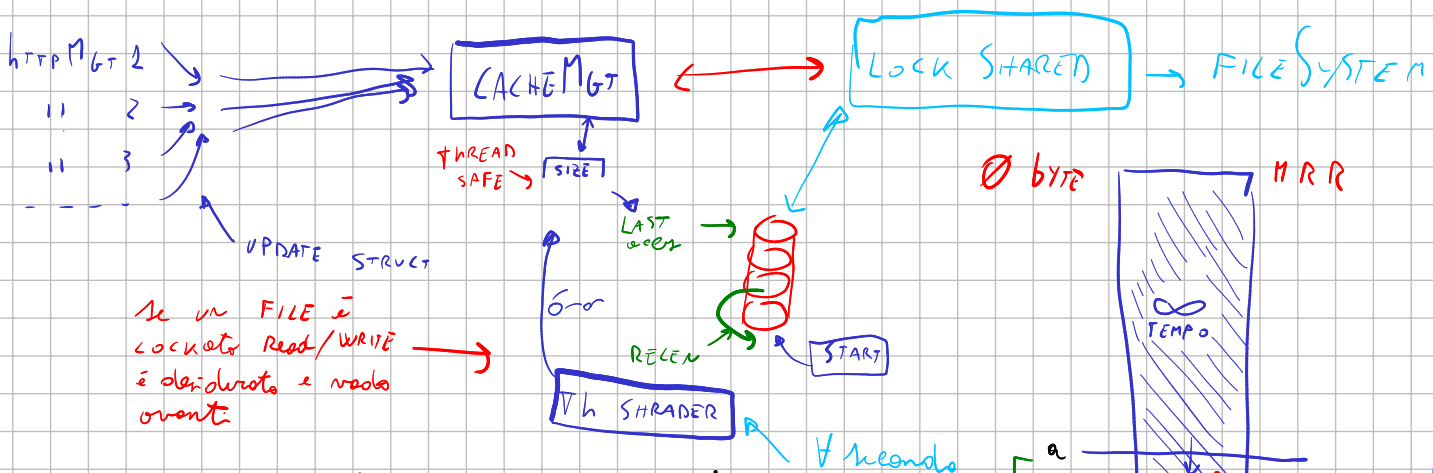
4 KEEP ALIVE

5 CLOSE_CLIENT

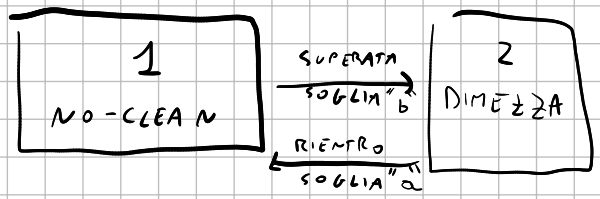
SEND

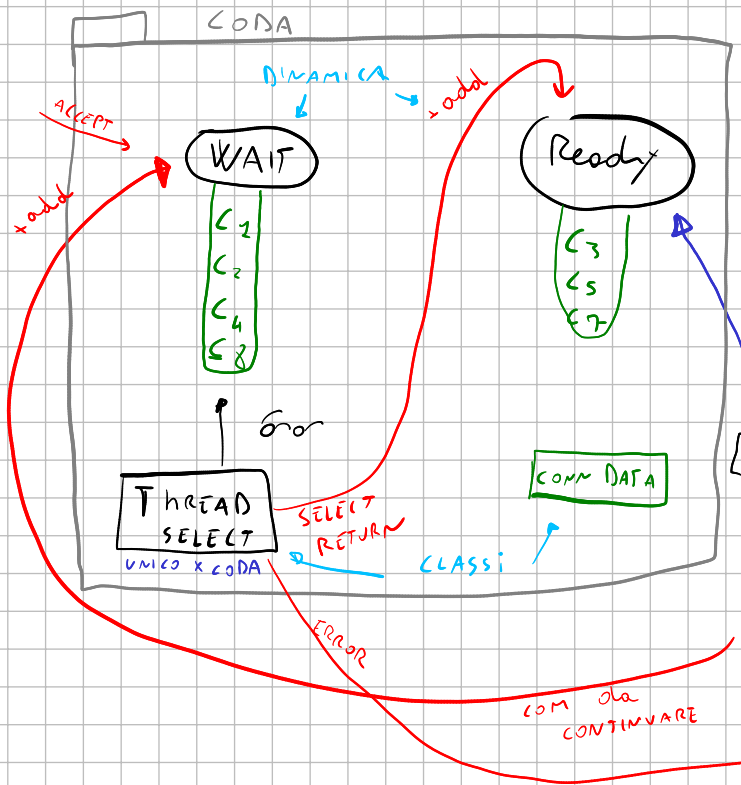
RIMETTI IN CODA per attesa

END



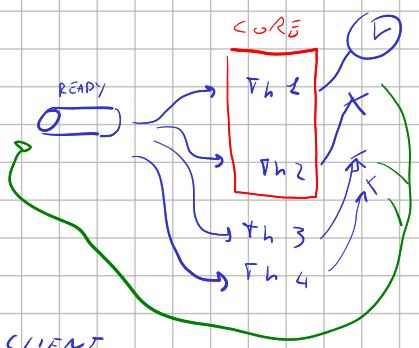
GESTIONE FILE SHREDDING
MACCHINA a STATI





C_i : oggetti connessioni, per tenere METADATI

MUTUA ESCLUSIONE MEDIANTE TOKEN



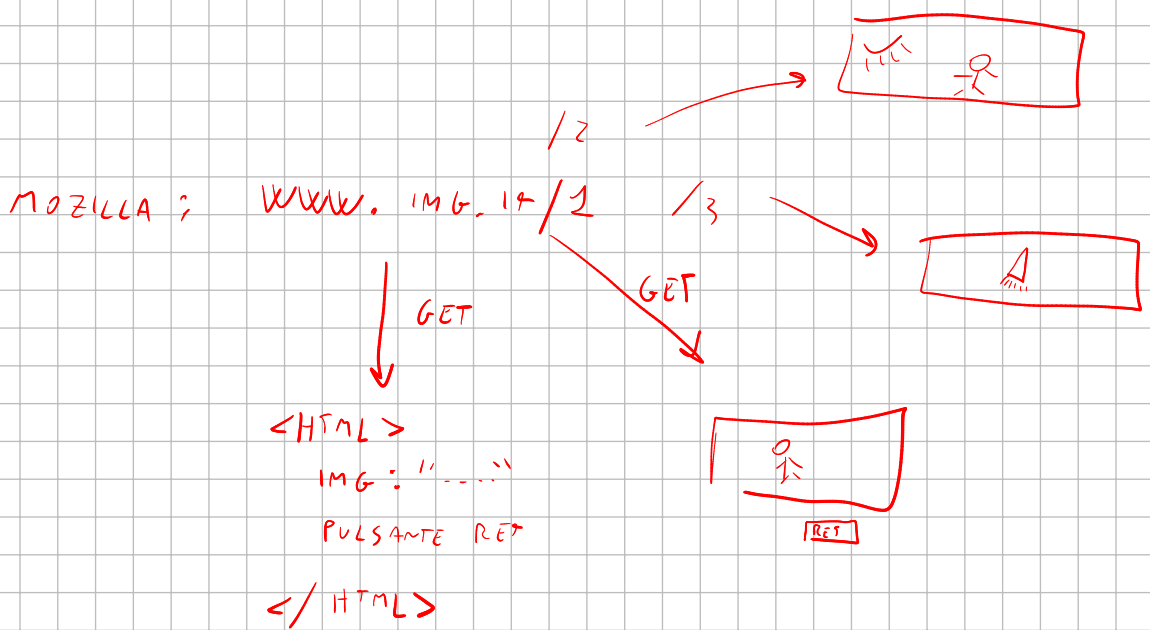
CLIENT o KEEP-ALIVE
da capire

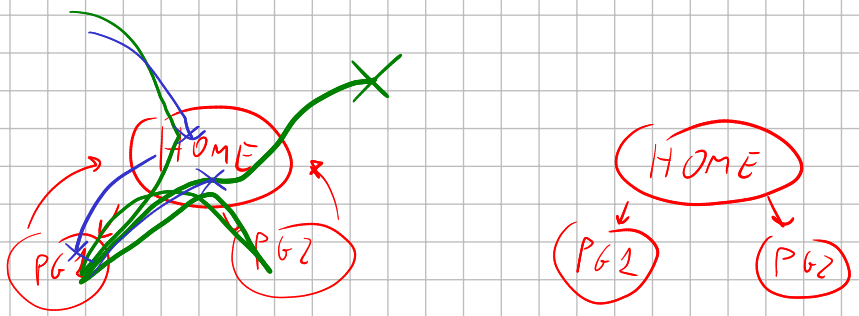
HOME

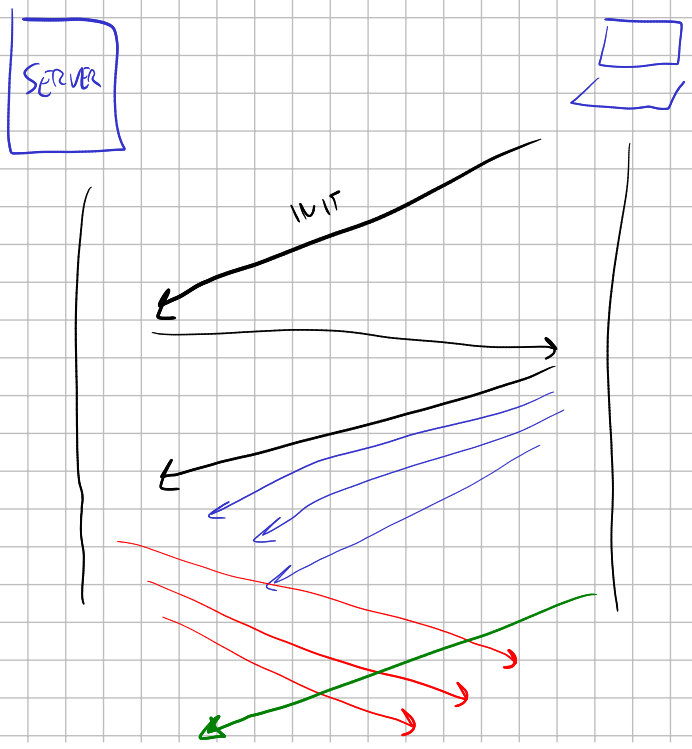
- LEONE - NAVE - AEREO .....

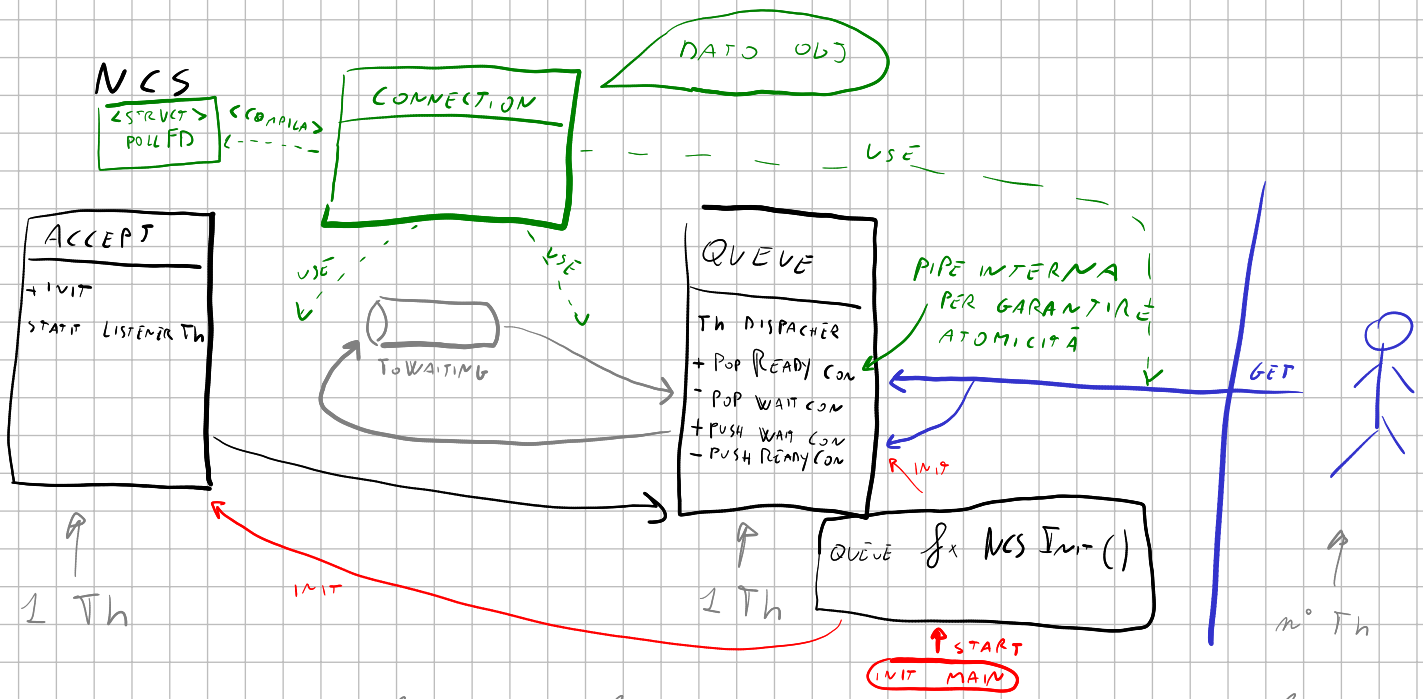
↑

HREF: "..."









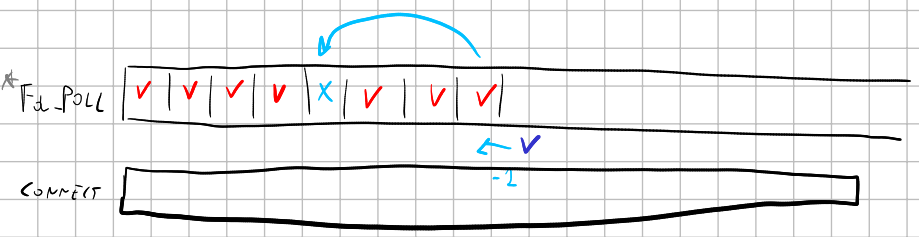
La QUEUE usa le PIPE al suo interno per sfruttare la possibilità che il S.O. mette a disposizione dei fare altre BLOCCANTI

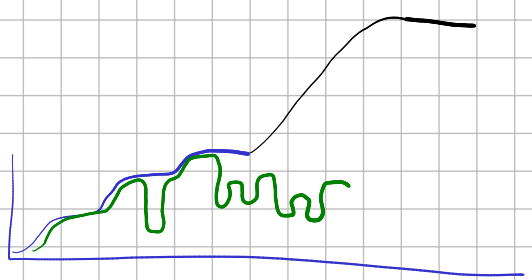
Il Th DISPATCHER guarda soltanto l'uscita della PIPE TOWAKING, e gli altri FD da guardare, se arriva qualcosa di nuovo li sveglia, la legge e la rimette in POLL dopo averla gestita, gli altri estremi della PIPE, anche se codificate dentro la CLASSE (per semplicità di PROGETTO) vengono SEMPRE seguiti da altri flussi di esecuzione

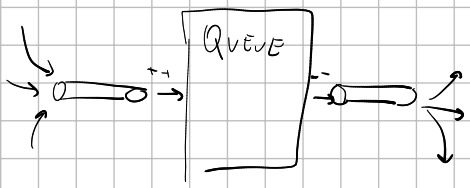
EVENTI POLL di interesse per la KEEP-ALIVE

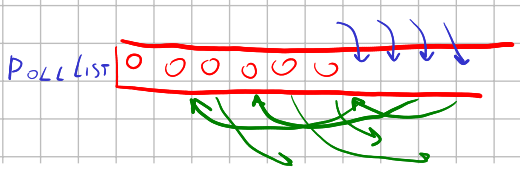
POLLRDHUP (since Linux 2.6.17)
Stream socket peer closed connection, or shut down writing half of connection. The _GNU_SOURCE feature test macro must be defined (before including any header files) in order to obtain this definition.

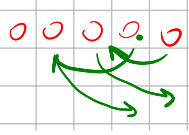
POLLHUP
Hang up (only returned in revents; ignored in events). Note that when reading from a channel such as a pipe or a stream socket, this event merely indicates that the peer closed its end of the channel. Subsequent reads from the channel will return 0 (end of file) only after all outstanding data in the channel has been consumed.



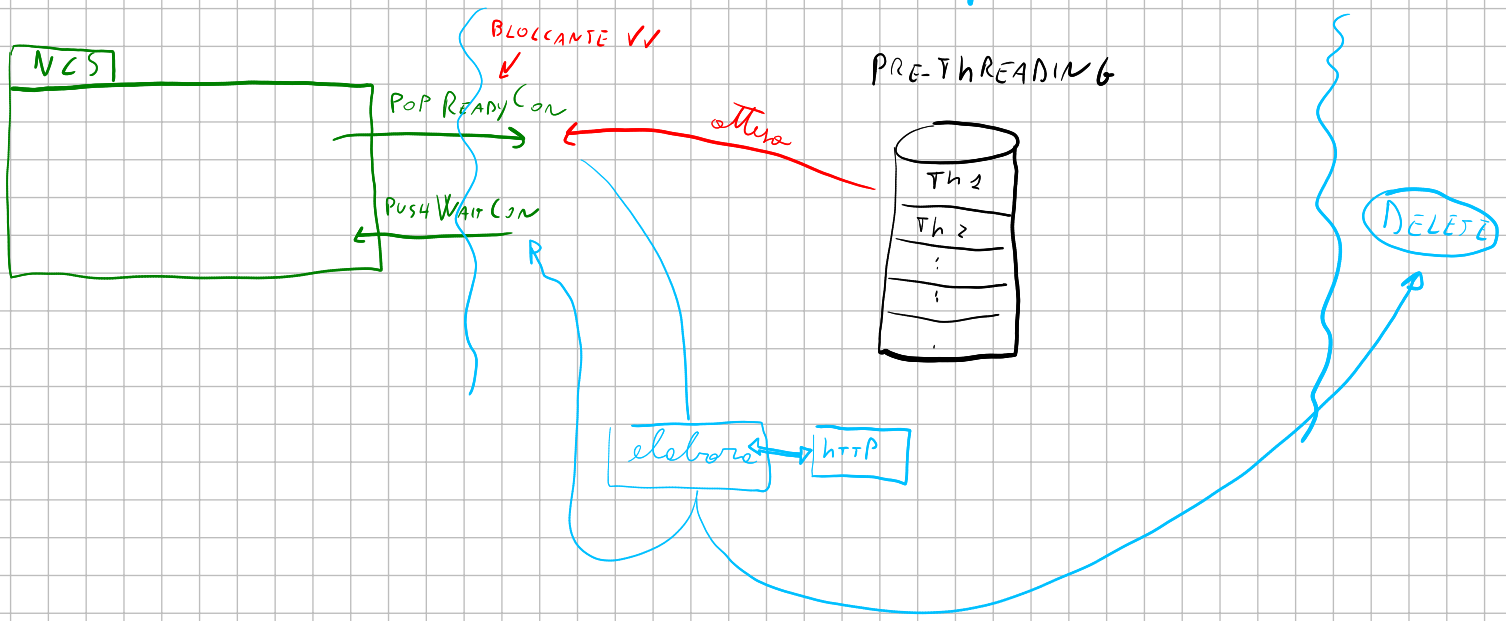


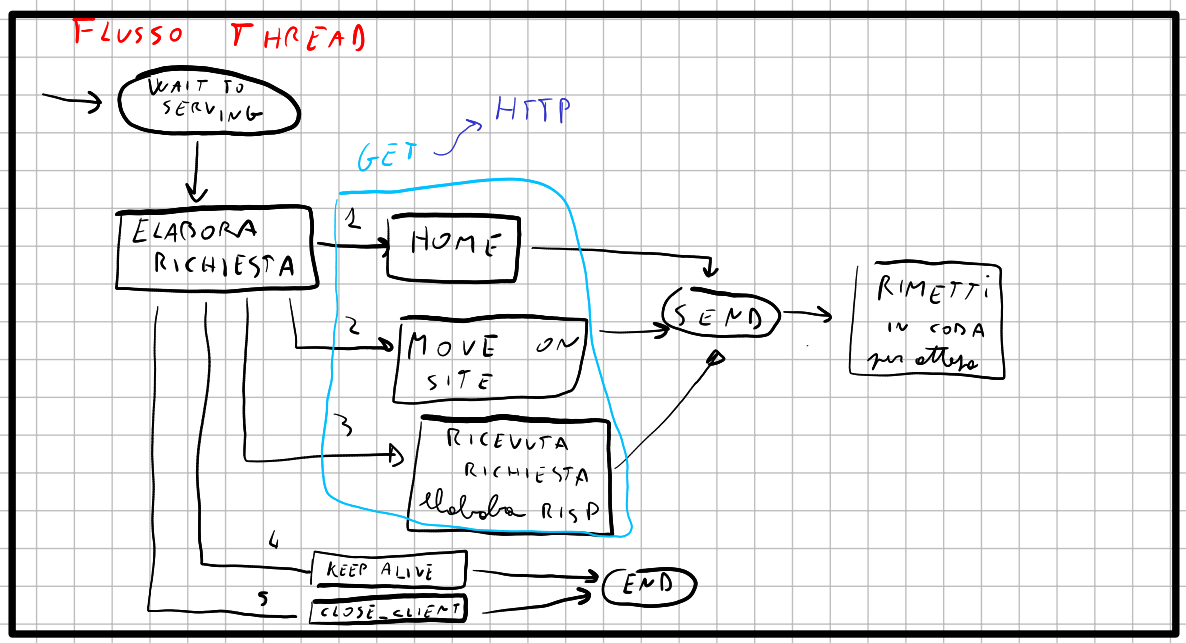




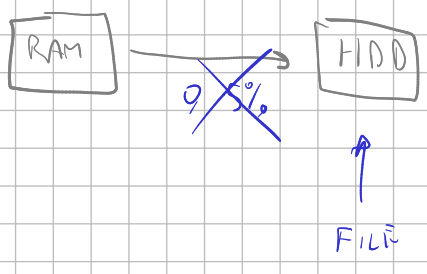


CORE ELABORATION SYSTEM

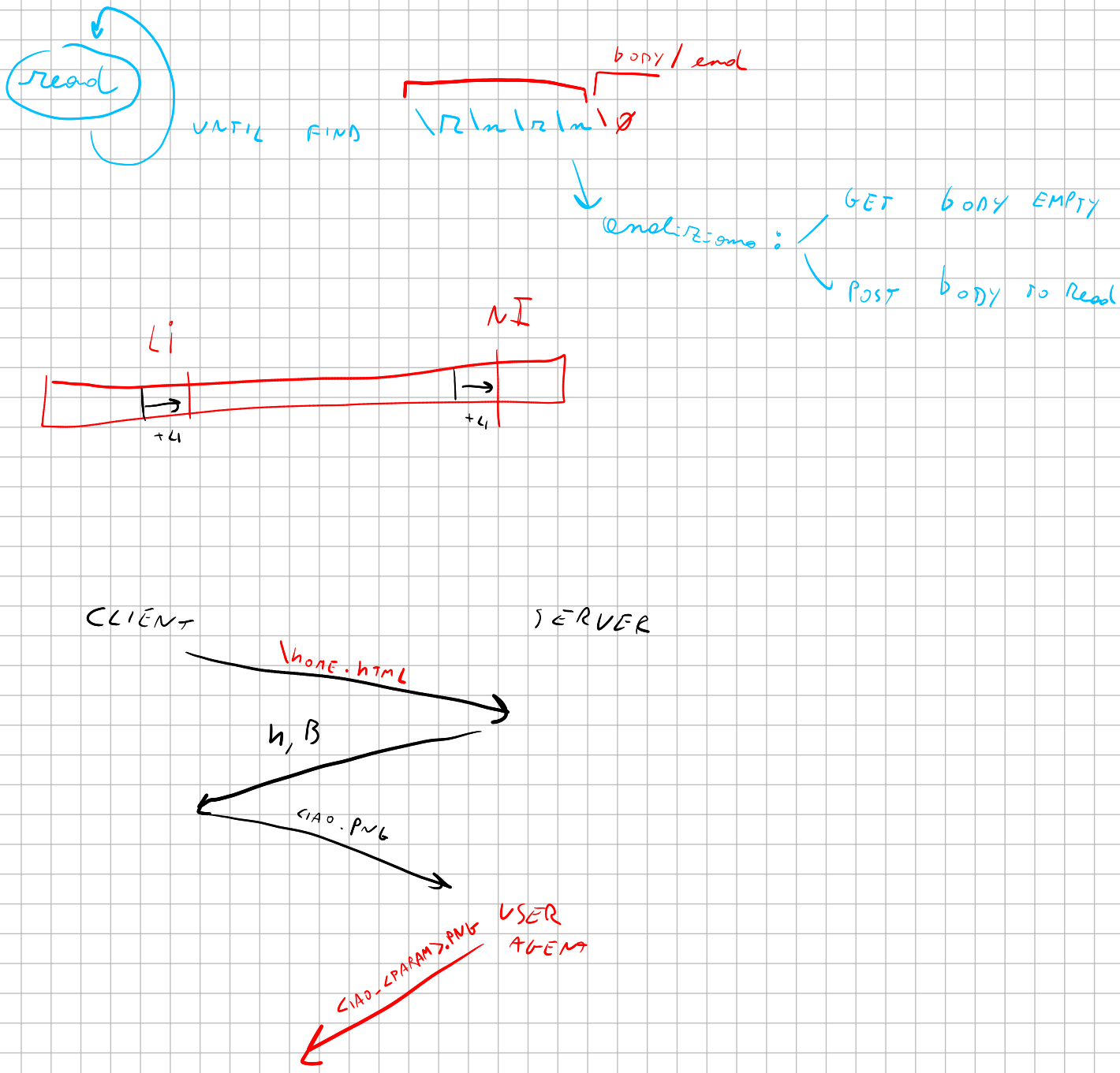




RIFLETTERE se i THREAD devono sapere se c'è stata una BROKEN-PIPE sulla connessione in elaborazione e interrompere l'elaborazione in anticipo

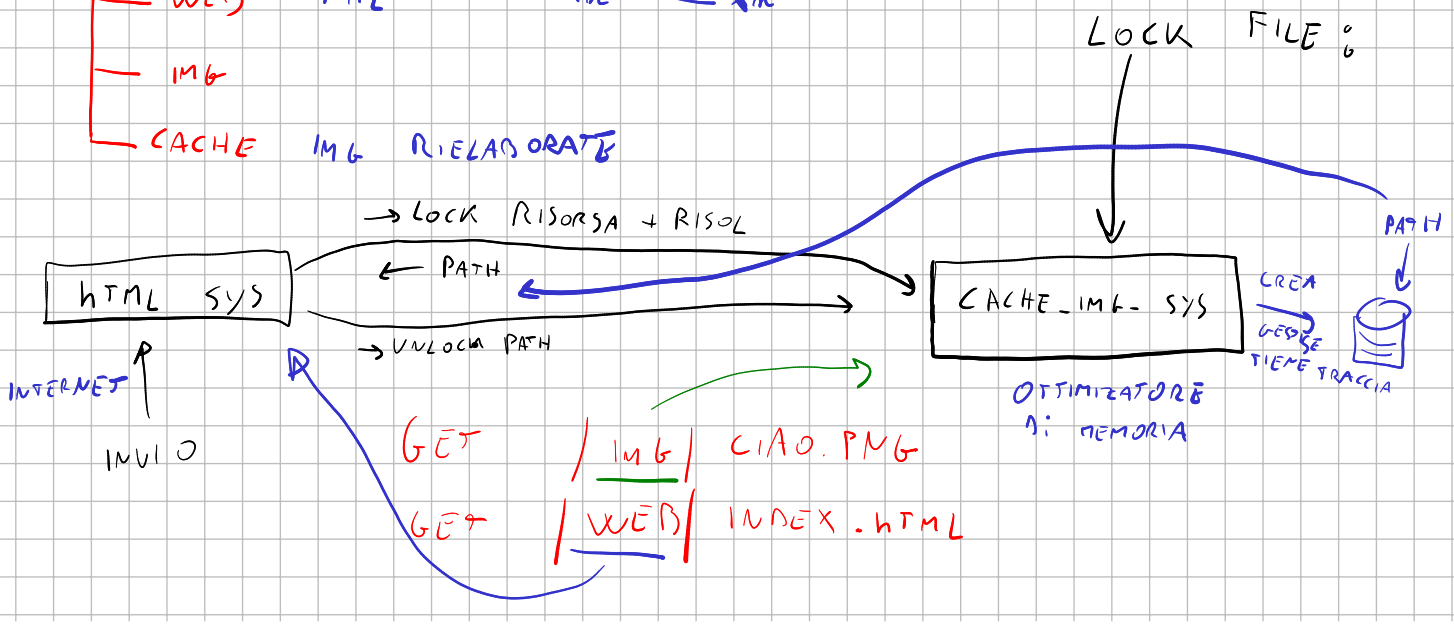


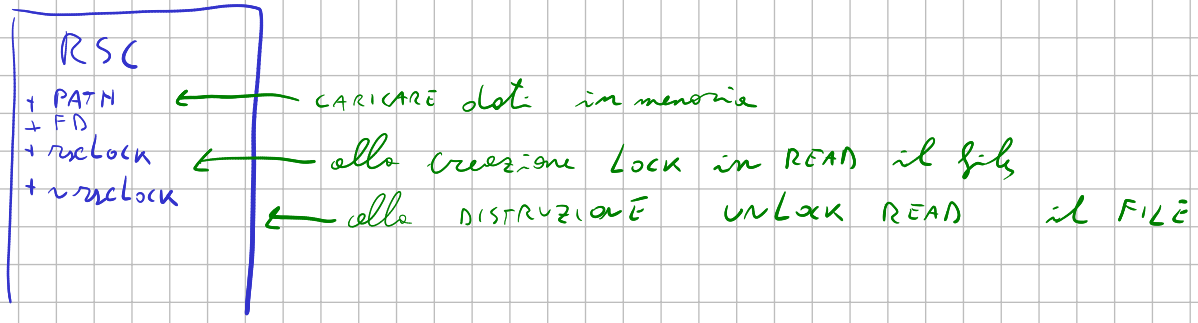
ENGINE	ELABORATION	SYSTEM	ECS
WORKING	SERVE	SYSTEM	WSS
ELABORATION	SYSTEM		ES
CORE	ENGINE	SYSTEM	CES ←
CORE	ELABORATION	SYSTEM	



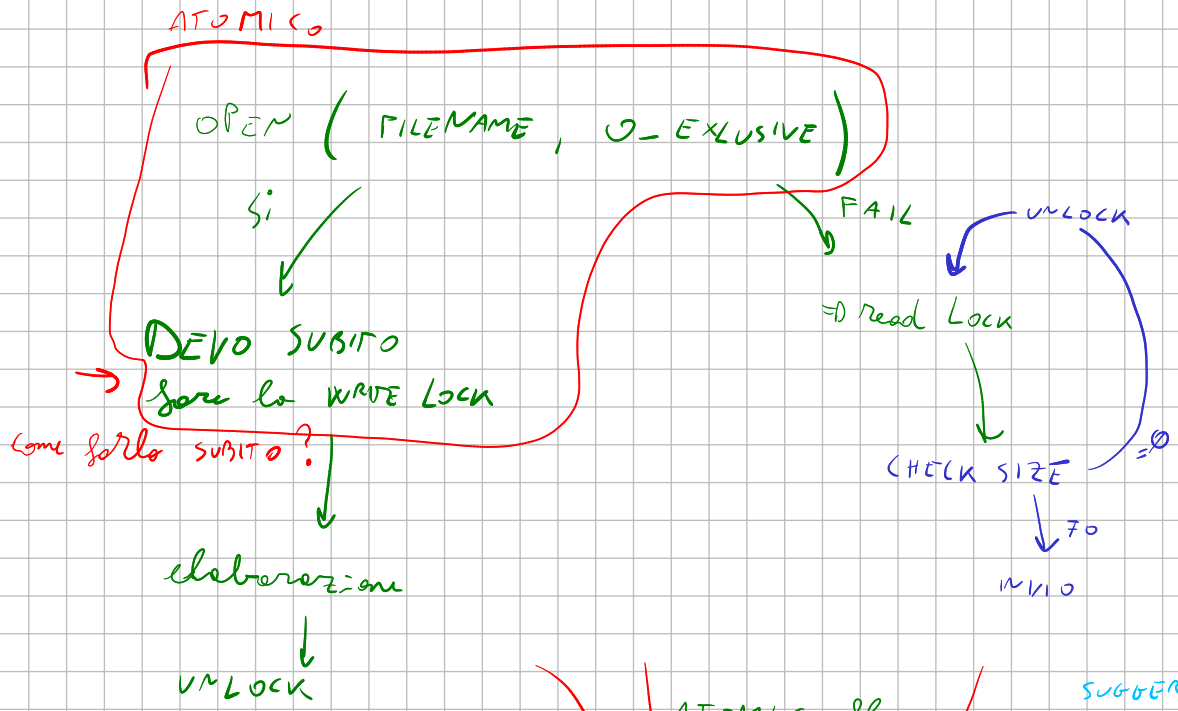
SOTTO SIS HTML

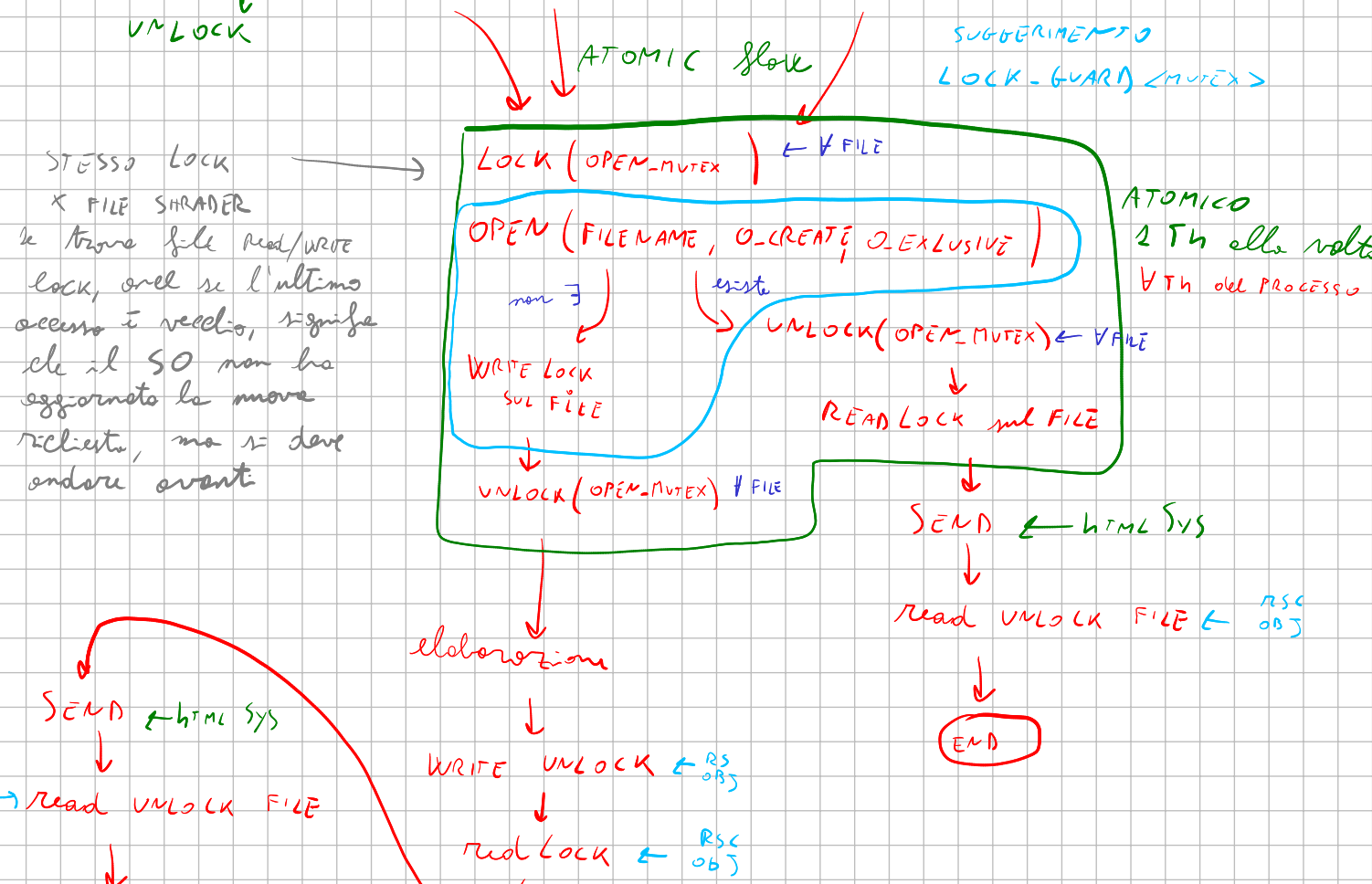
- ROOT SERVER
  - WEB html — \n — \n
  - IMG
  - CACHE IMG RIELABORATE

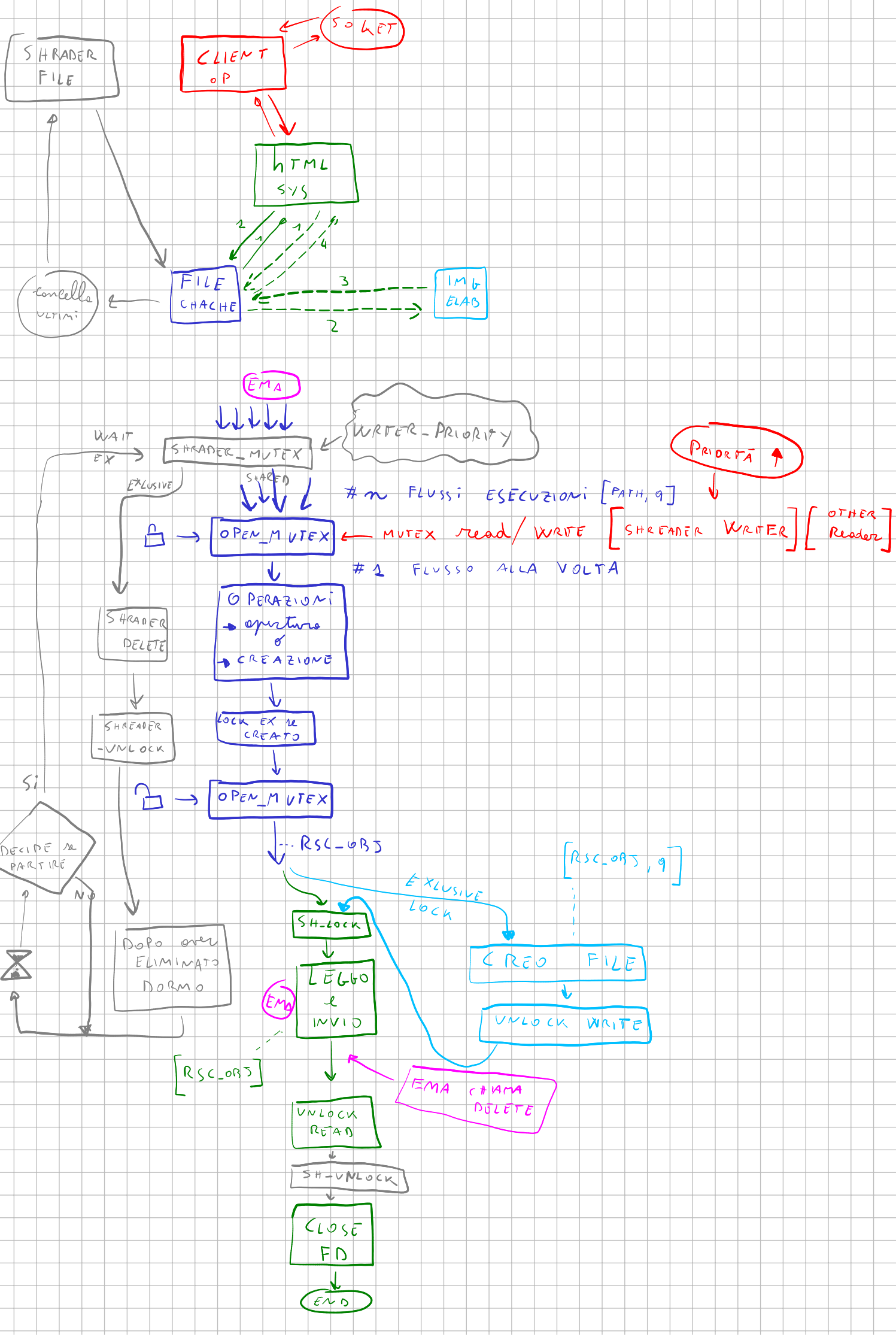


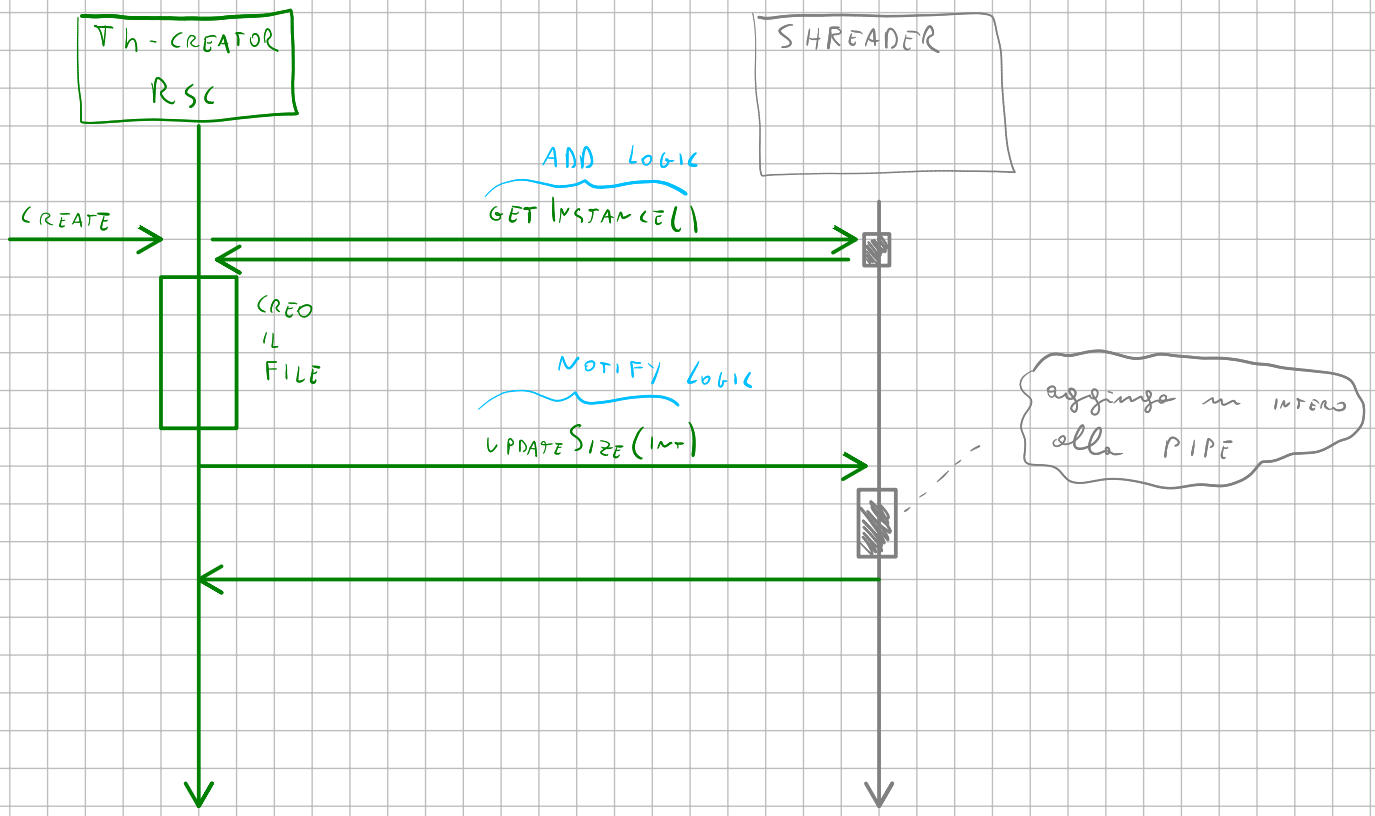


LOCK WRITE sul FILE lo fa il CACHE MGT in creazione fino alla fine dell'elaborazione, poi lo rilascia

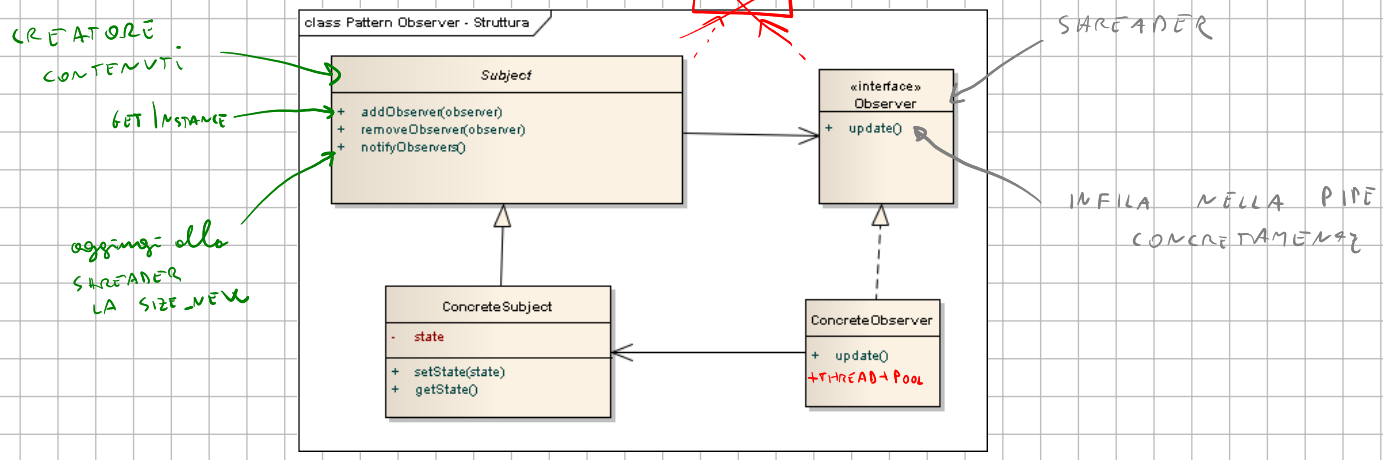





SHRADER FILE
CLIENT OP
SOCKET
HTML SYS
FILE CHACHE
IMG ELAB
cancella ULTIMI
1
2
3
4
EMA
WRITER_PRIORITY
WAIT EX
SHRADER_MUTEX
EXLUSIVE
SHARED
PRIORITÀ
n FLUSSI ESECUZIONI [PATH, q]
OPEN_MUTEX
MUTEX read/WRITE [SHREADER WRITER] [OTHER Reader]
1 FLUSSO ALLA VOLTA
OPERAZIONI
→ apertura
o
→ CREAZIONE
SHRADER DELETE
SHREADER -UNLOCK
LOCK EX se CREATO
OPEN_MUTEX
Si
DECIDE se PARTIRE
No
Dopo aver ELIMINATO DORMO
RSC_OBJ
[RSC_OBJ, q]
EXLUSIVE LOCK
SH-LOCK
LEGGO e INVIO
EMA
[RSC_OBJ]
CREO FILE
UNLOCK WRITE
EMA CHIAMA DELETE
UNLOCK READ
SH-UNLOCK
CLOSE FD
END



così non esiste

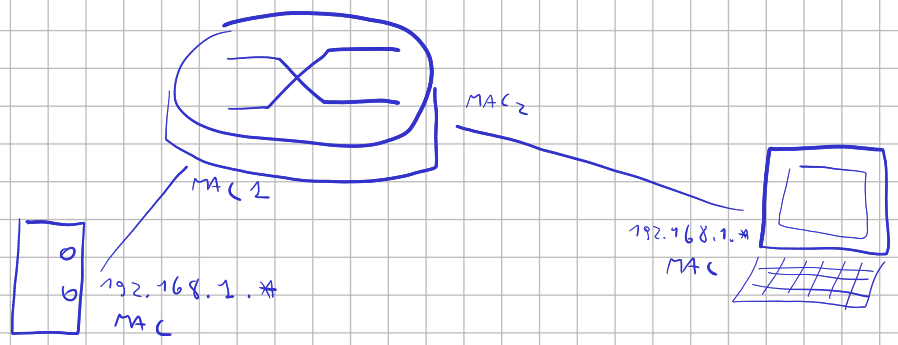


NOTIFY → UPDATE

ADD SIZE → PUT IN PIPE

INTERFACCIA

CONCRETO



PING dei sistemi da fare PRE-TEST

IMAGE_REWORK disattivato !!

SCRIPT

SERVER CONDITION

MEMORY ~~HDD~~ SSD

RAM: 16 GB

ETHERNET

RIAVVIO MANUALE 2 VOLTE (1 NGINX, 1 APACHE)

PAGE LOAD

1) TESTUALE: INDEX

2) IMAGINE: CAFFE, COMPUTER, SPACE SHIP, CAR

5 schede

BENCHMARK

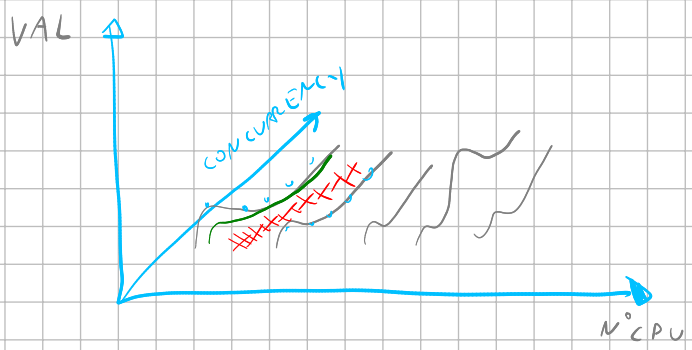
N° Req TOT: 100'000

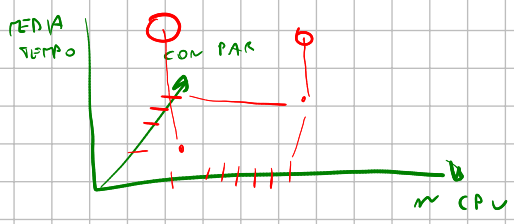
N° Req PARALLELE: 100, 200, 400, 800, 1600

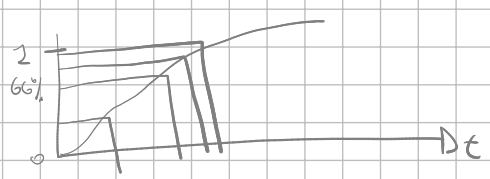
N° CPU: 1 ... n CPU

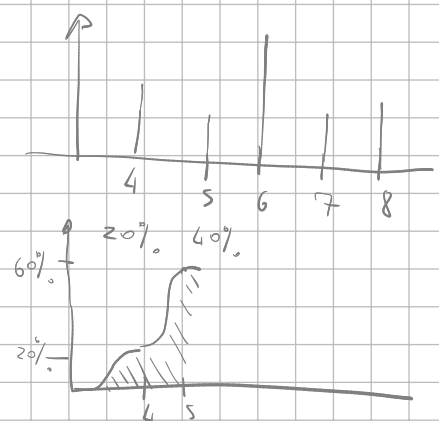
VAL TO read:

- HTML TRANSFERT Rate
- TOTAL TIME









MATLAB XML

PYTHON : RUN & SAVE

XML / .DAT

MATLAB

1°
:
n°

[TESTO]

n+1 DATO <SEP> DATO <sep> - - - - -

CONFICURAZIONE SERVER (SSD, HDD)

NOME DIR : NOSTRO

NO

SCRIPT

Y - M - G _ _ H - m - s _ _

<Resource> BENCKMARK.DAT

APACHE 2

POSSIBILI

STRUTTURA

N° CPU	N° PARALLEL CONNECTION	Request X SECOND [Req/s]	TIME X Request SINGOLO [s]	TRANSFER RATE [KB/s]
X	Y			