Performance measurement and tuning of remote acquisition

Lukasz Makowski

February 2, 2016

Location

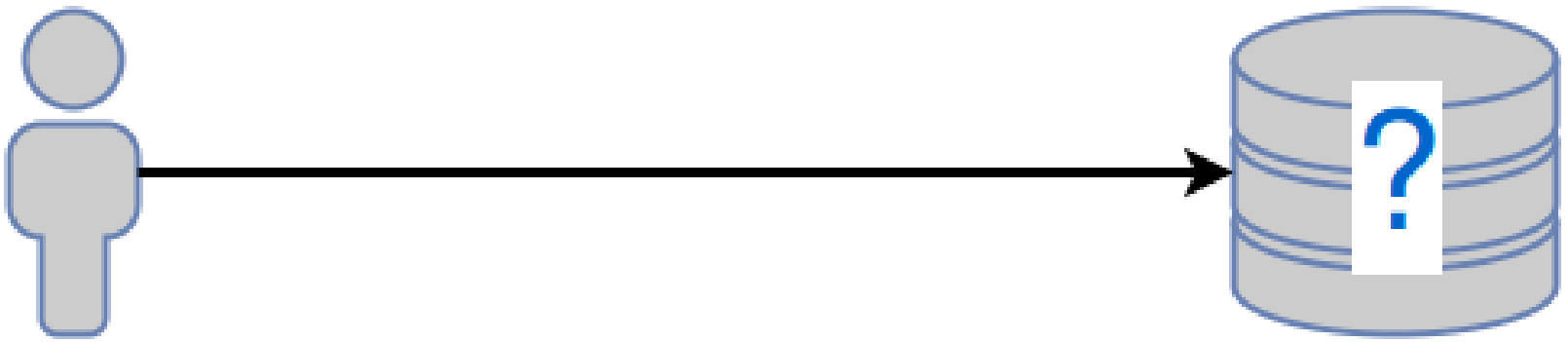
Netherlands Forensic Institute
Supervisor : Ruud Schramp

Agenda

1. Remote acquisition - research motivation introduction
2. Research scope and questions posed
3. Approach & methods taken
4. Results
5. Future work

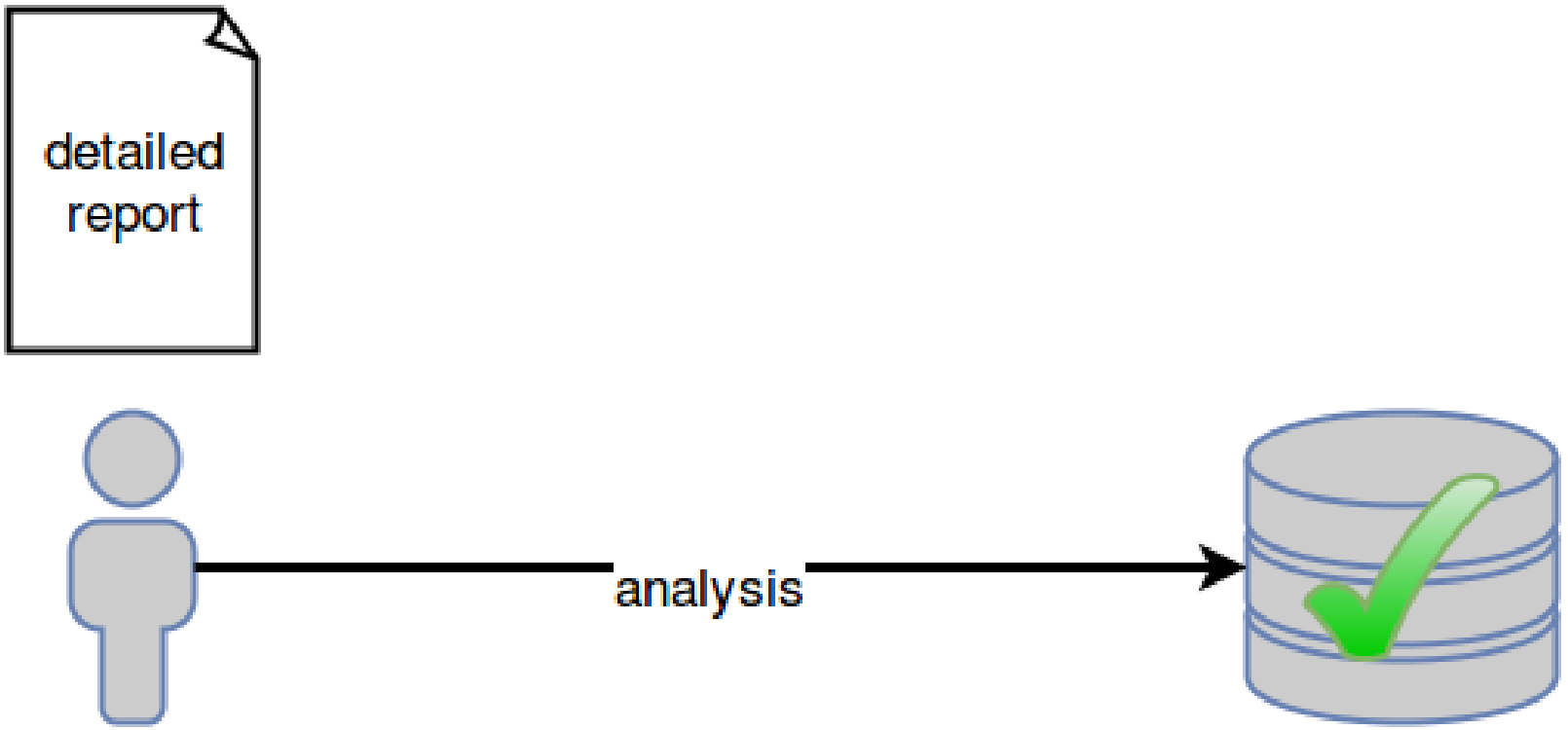
Forensic acquisition

"Old-school" approach:

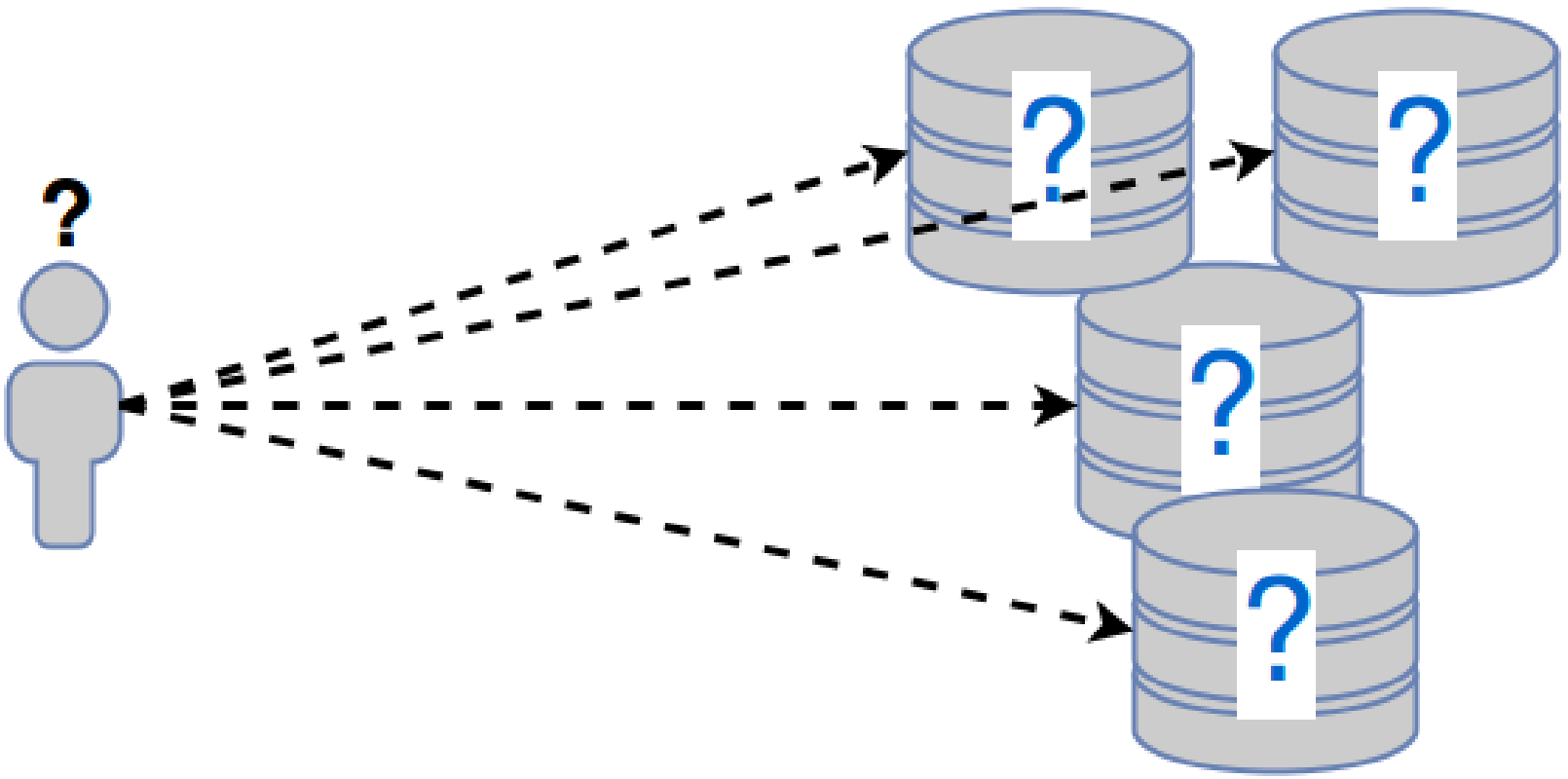


Forensic acquisition

"Old-school" approach:



Forensic acquisition



Forensic acquisition

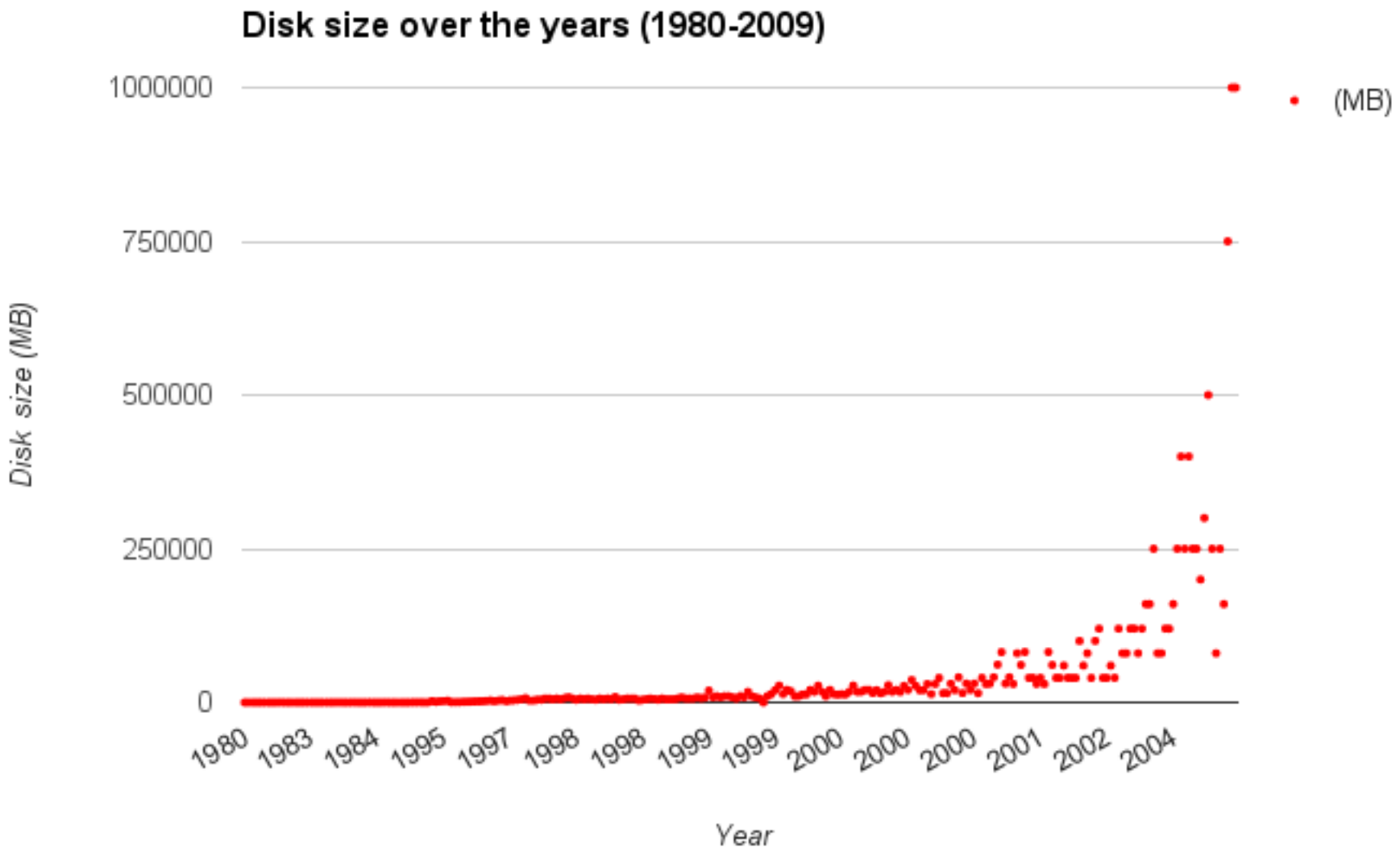
The bottlenecks in the current process:

Forensic acquisition

The bottlenecks in the current process:

- quantity : regular disk size increases

Forensic acquisition



Data source : http://www.mkomo.com/cost-per-gigabyte

Forensic acquisition

The bottlenecks in the current process:

- quantity : regular disk size increases

Forensic acquisition

The bottlenecks in the current process:

- quantity : regular disk size increases
- staffing : forensic experts cannot be easily multiplied :(

Forensic acquisition

The bottlenecks in the current process:

- quantity : regular disk size increases
- staffing : forensic experts cannot be easily multiplied :(
- legal : court approval takes time

Forensic acquisition

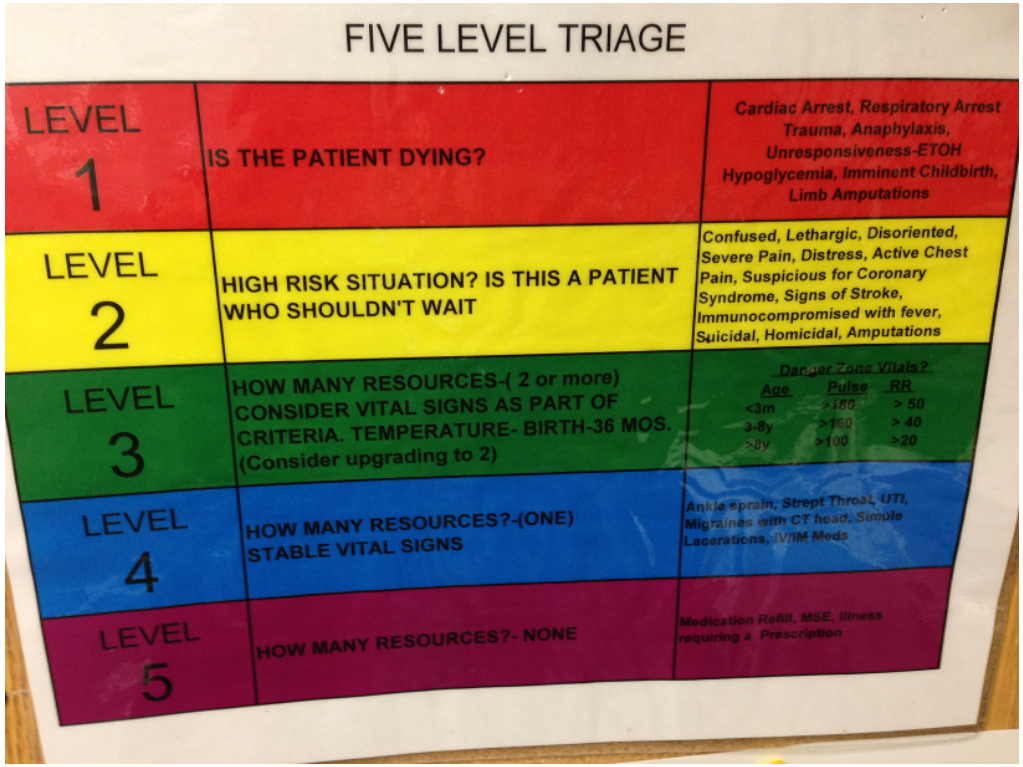
The bottlenecks in the current process:

- quantity : regular disk size increases
- staffing : forensic experts cannot be easily multiplied :(
- legal : court approval takes time

But there is a possible solution! (at least to the first two points ...)

Forensic triage - the cure for pain?

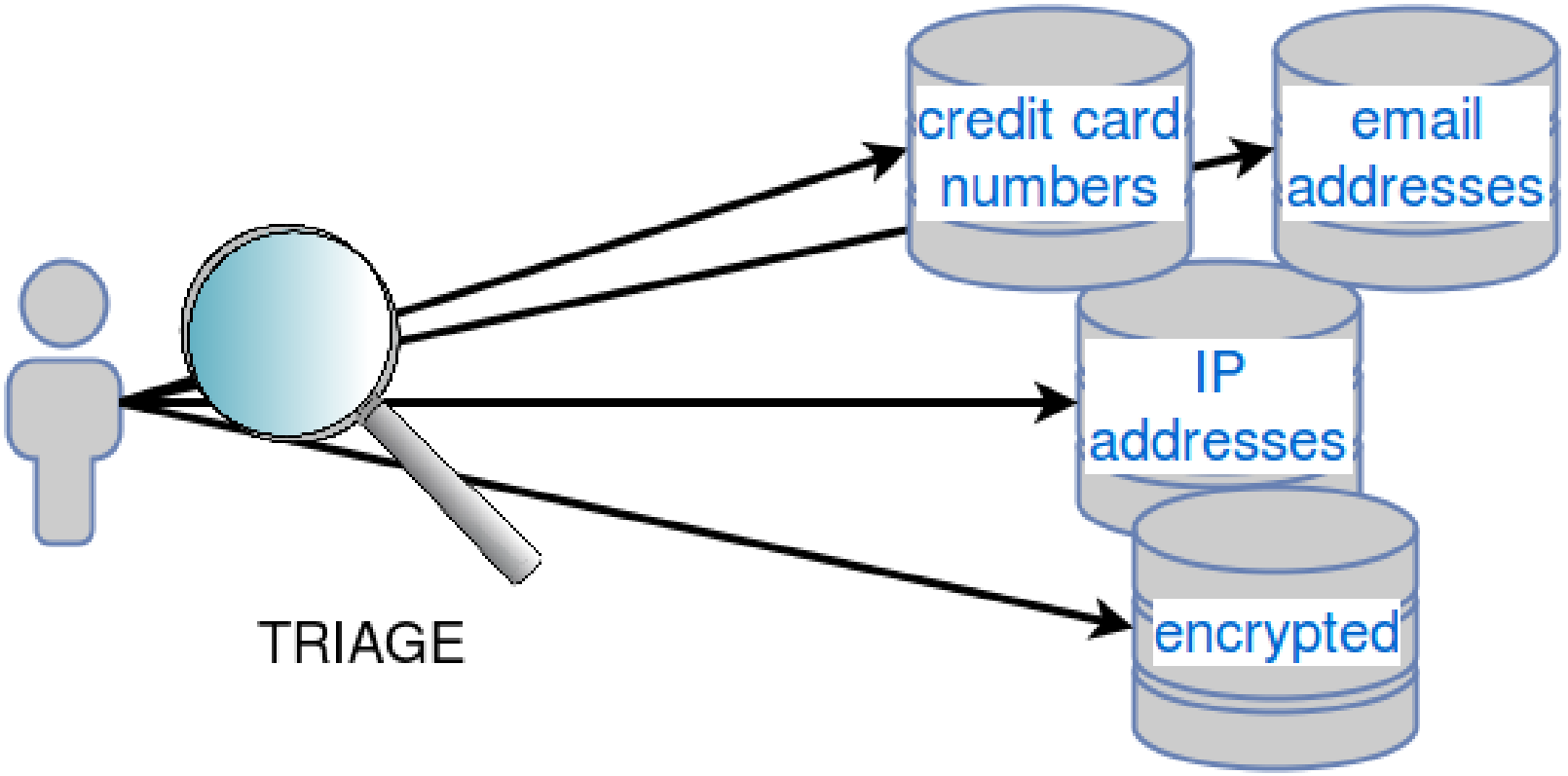
Triage is the process of determining the priority of patients' treatments based on the severity of their condition. This rations patient treatment efficiently when resources are insufficient for all to be treated immediately.



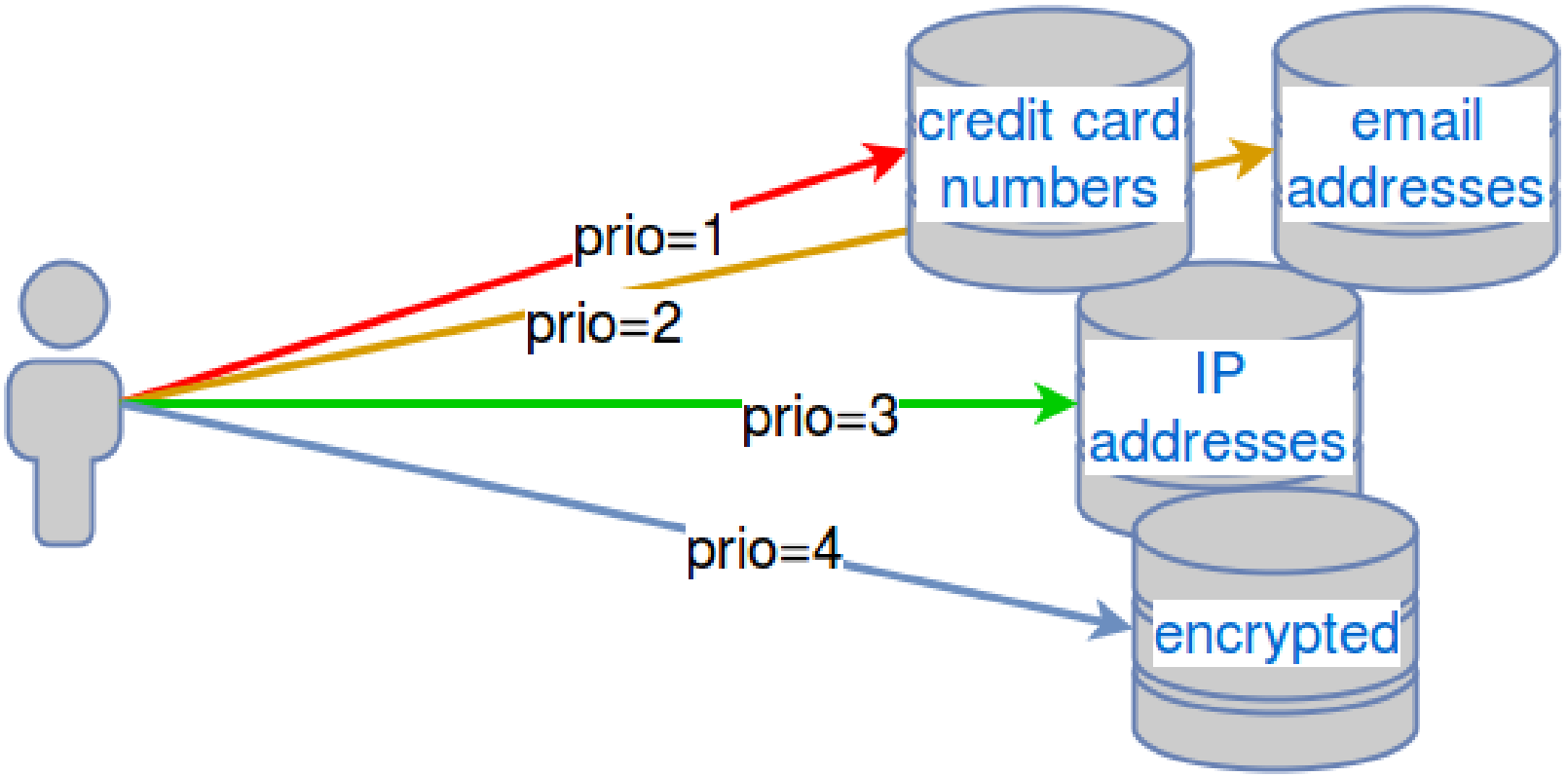
Source : https://en.wikipedia.org/wiki/Triage

Source : https://cartadvocate.files.wordpress.com/2015/03/img_3788.jpg

Forensic triage - the cure for pain?

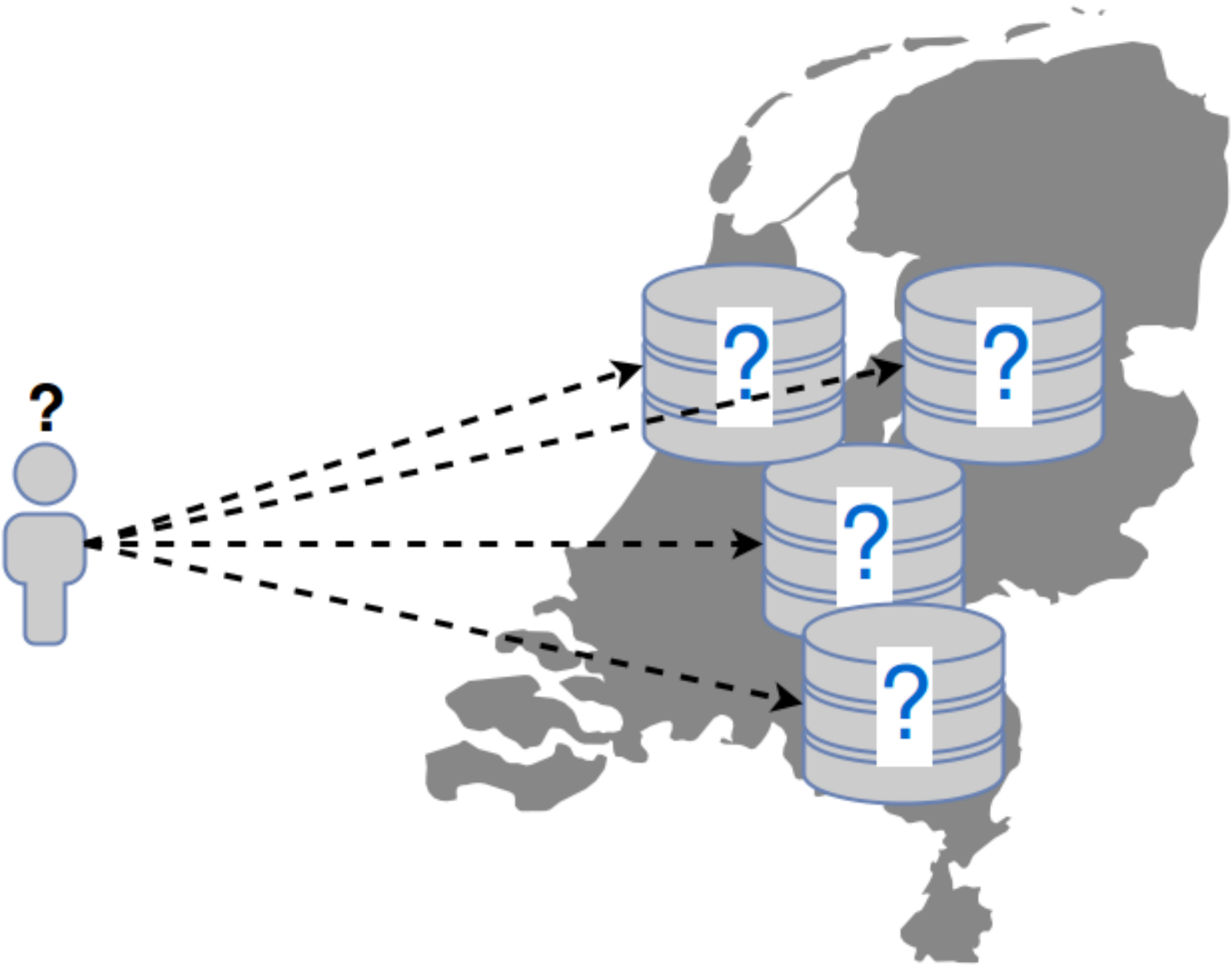


Forensic triage - the cure for pain?



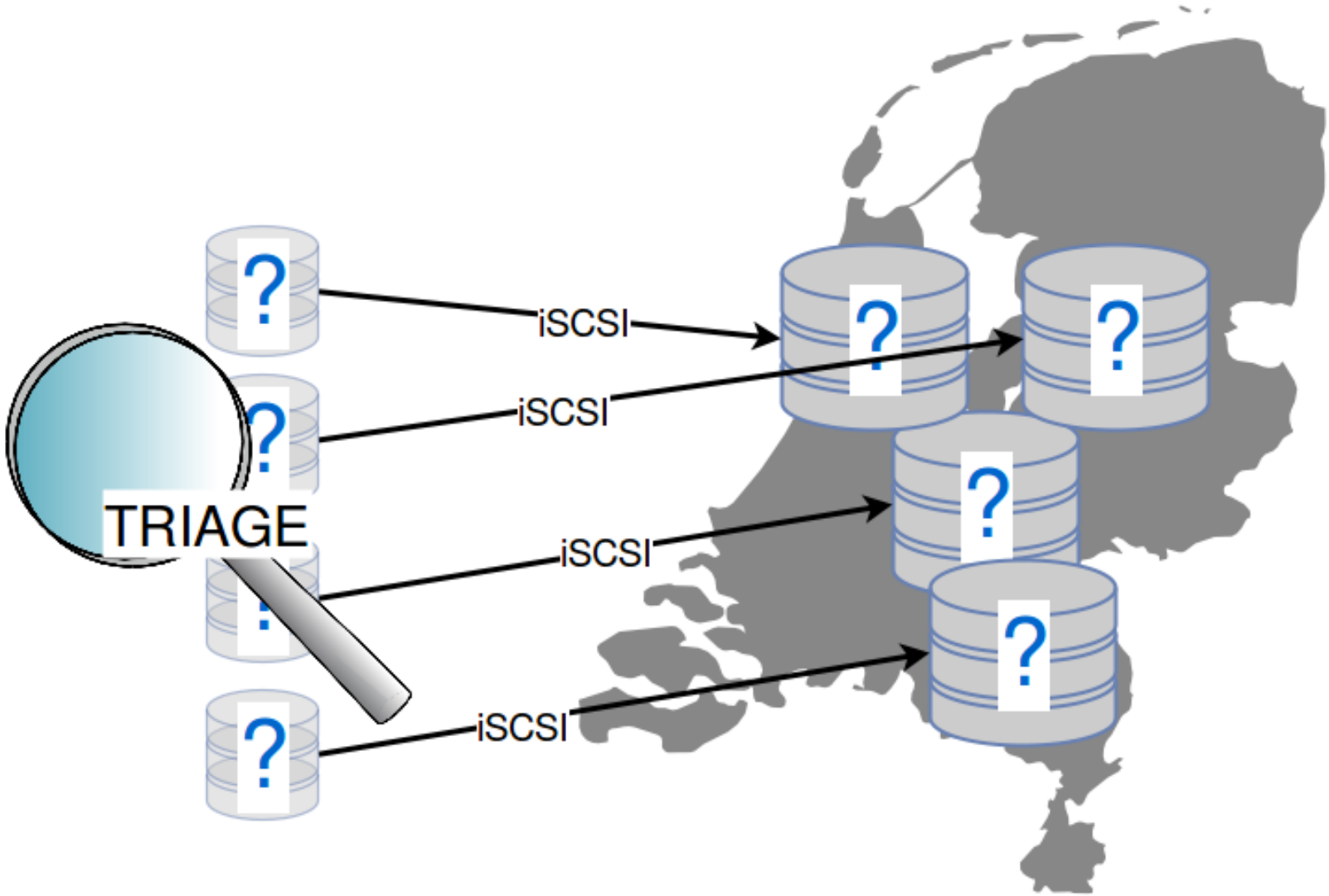
Remote triage

Remote triage - problem:



Remote triage

Remote triage - approach:



Remote triage

Remote triage' issues:

Remote triage

Remote triage' issues:

- WAN links introduce whole subset of problems (delay, bandwidth, packet loss, ...)

Remote triage

Remote triage' issues:

- WAN links introduce whole subset of problems (delay, bandwidth, packet loss, ...)
- iSCSI uses TCP in transport layer (TCP limitations inherited)

Remote triage

Remote triage' issues:

- WAN links introduce whole subset of problems (delay, bandwidth, packet loss, . . .)
- iSCSI uses TCP in transport layer (TCP limitations inherited)
- iSCSI is not well suited to WAN links

Remote triage - issues

Essentially the problem can be synthesized to simple question :

Remote triage - issues

Essentially the problem can be synthesized to simple question :
How to make the remote triage as efficient as possible?

Remote triage - issues

Areas where the speed-up can be potentially achieved:

Remote triage - issues

Areas where the speed-up can be potentially achieved:

- TCP protocol tuning

Remote triage - issues

Areas where the speed-up can be potentially achieved:

- TCP protocol tuning
- iSCSI stack tuning

Remote triage - issues

Areas where the speed-up can be potentially achieved:

- TCP protocol tuning
- iSCSI stack tuning
- Acquisition I/O optimisation

Remote triage - issues

Areas where the speed-up can be potentially achieved:

- TCP protocol tuning
- iSCSI stack tuning
- Acquisition I/O optimisation

  Yes. . . TCP and iSCSI options left in the defaults

Research scope

Acquisition I/O optimisation :

Research scope

Acquisition I/O optimisation :

- Is it feasible to enhance a transfer rate for acquisition performed on the iSCSI block device?

Research scope

Acquisition I/O optimisation :

- Is it feasible to enhance a transfer rate for acquisition performed on the iSCSI block device?
- Which techniques an application can use to improve on the transmission rate?

Research scope

Acquisition I/O optimisation :

- Is it feasible to enhance a transfer rate for acquisition performed on the iSCSI block device?
- Which techniques an application can use to improve on the transmission rate?
- How a link delay influences the experiment?

Research scope

Researching on potential I/O optimisation methods:

Research scope

Researching on potential I/O optimisation methods:

- prefetching (implies the usage of cache)

Research scope

Researching on potential I/O optimisation methods:

- prefetching (implies the usage of cache)
  - read-ahead

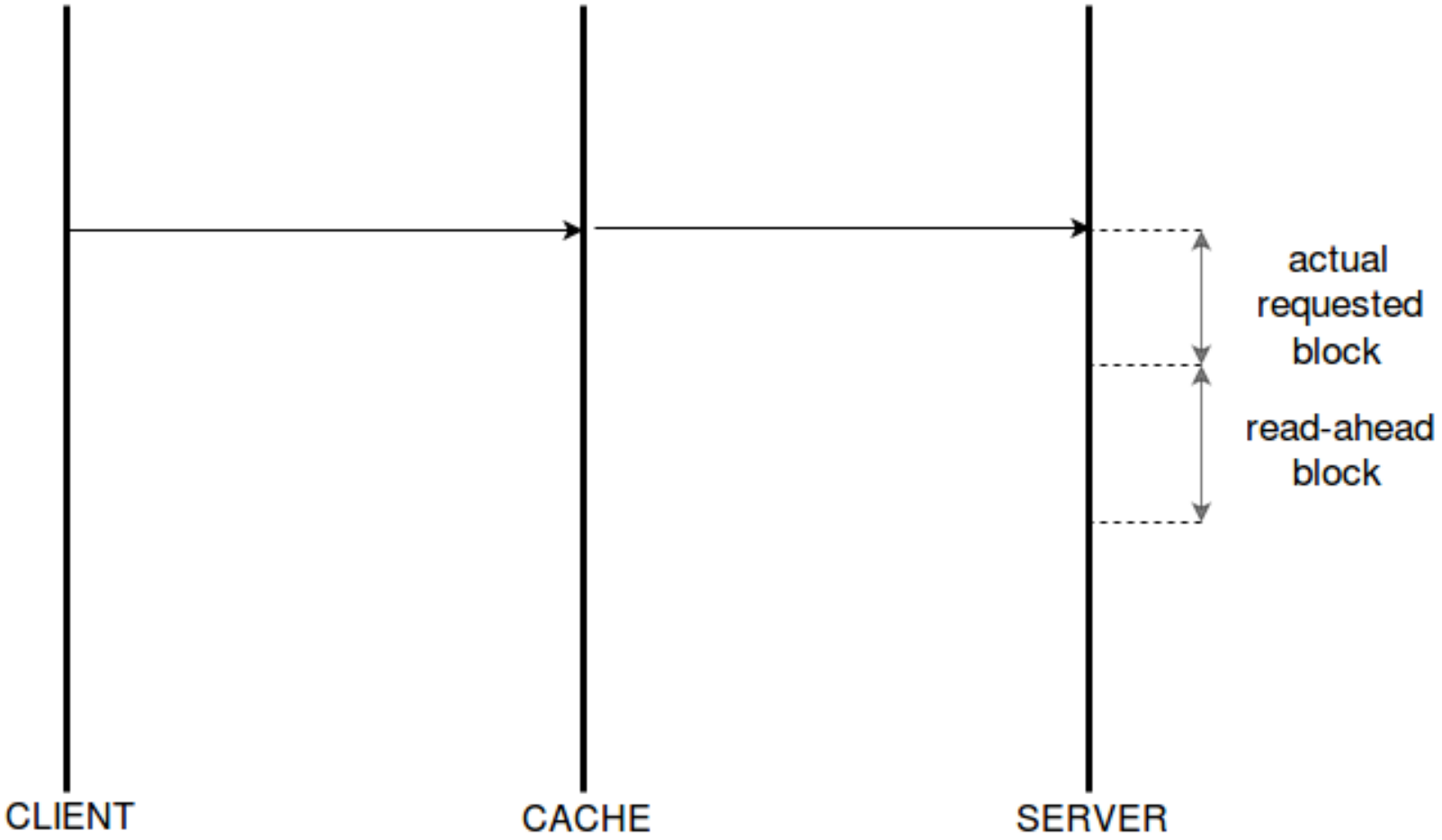
Research scope

Researching on potential I/O optimisation methods:

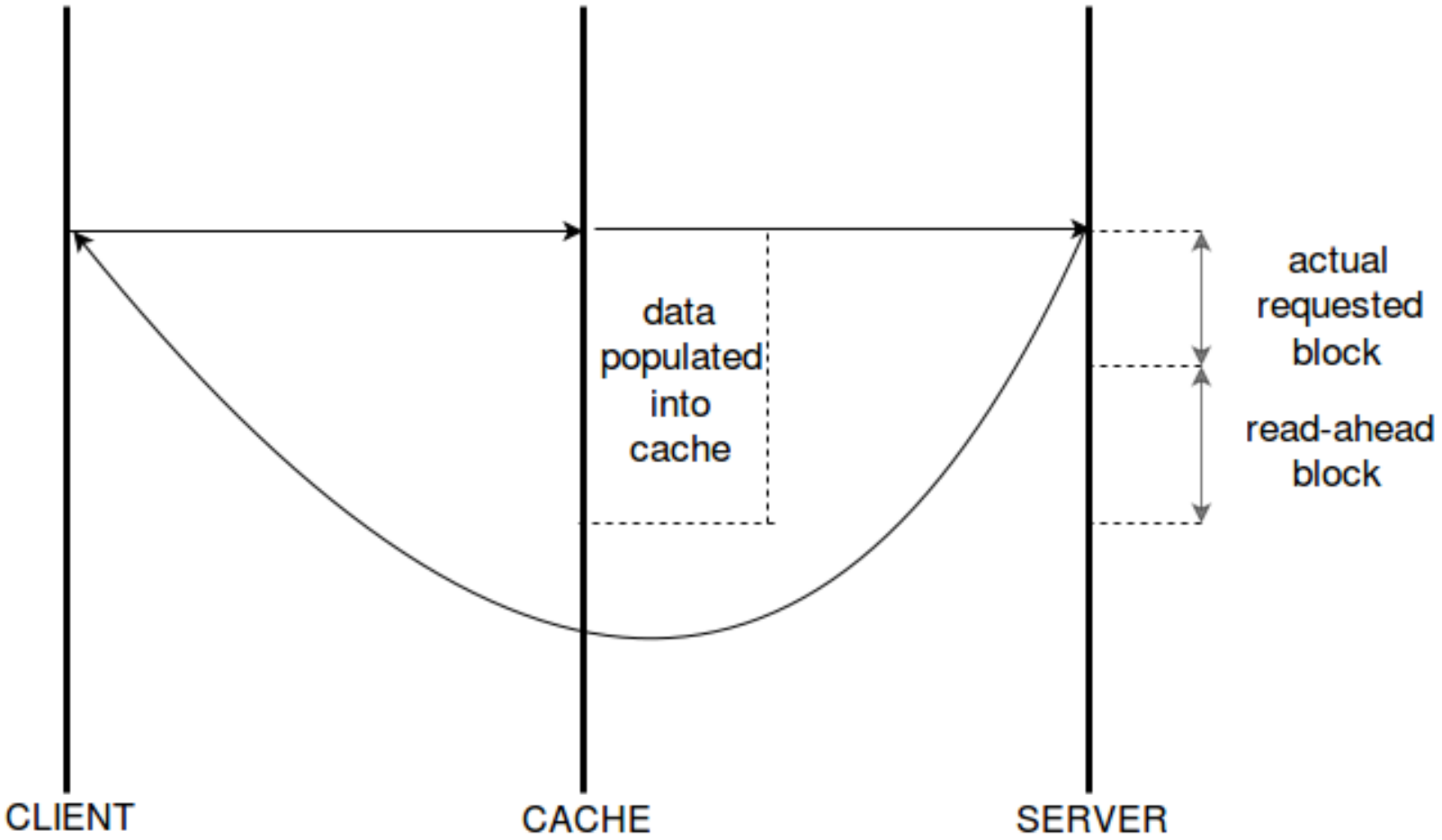
- prefetching (implies the usage of cache)
  - read-ahead
  - read-behind

Research scope - prefetching

Read-ahead : read block-size $\rightarrow$ cache MISS $\rightarrow$ read block-size+read-ahead

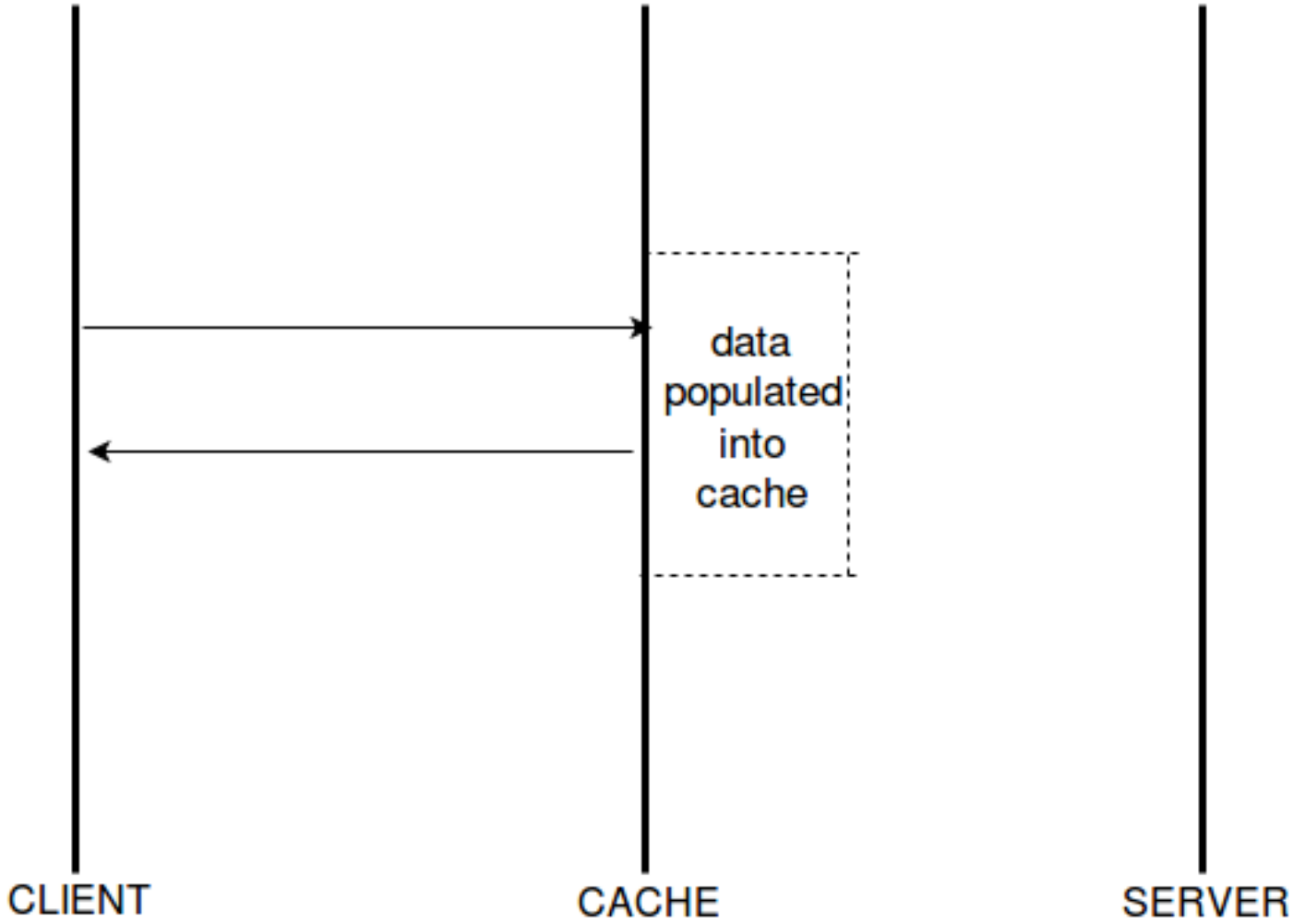


Research scope - prefetching



Research scope - prefetching

Read-ahead : read block-size $\rightarrow$ cache HIT



Research scope

Researching on potential I/O optimisation methods:

- prefetching (implies the usage of cache)
  - read-ahead
  - read-behind

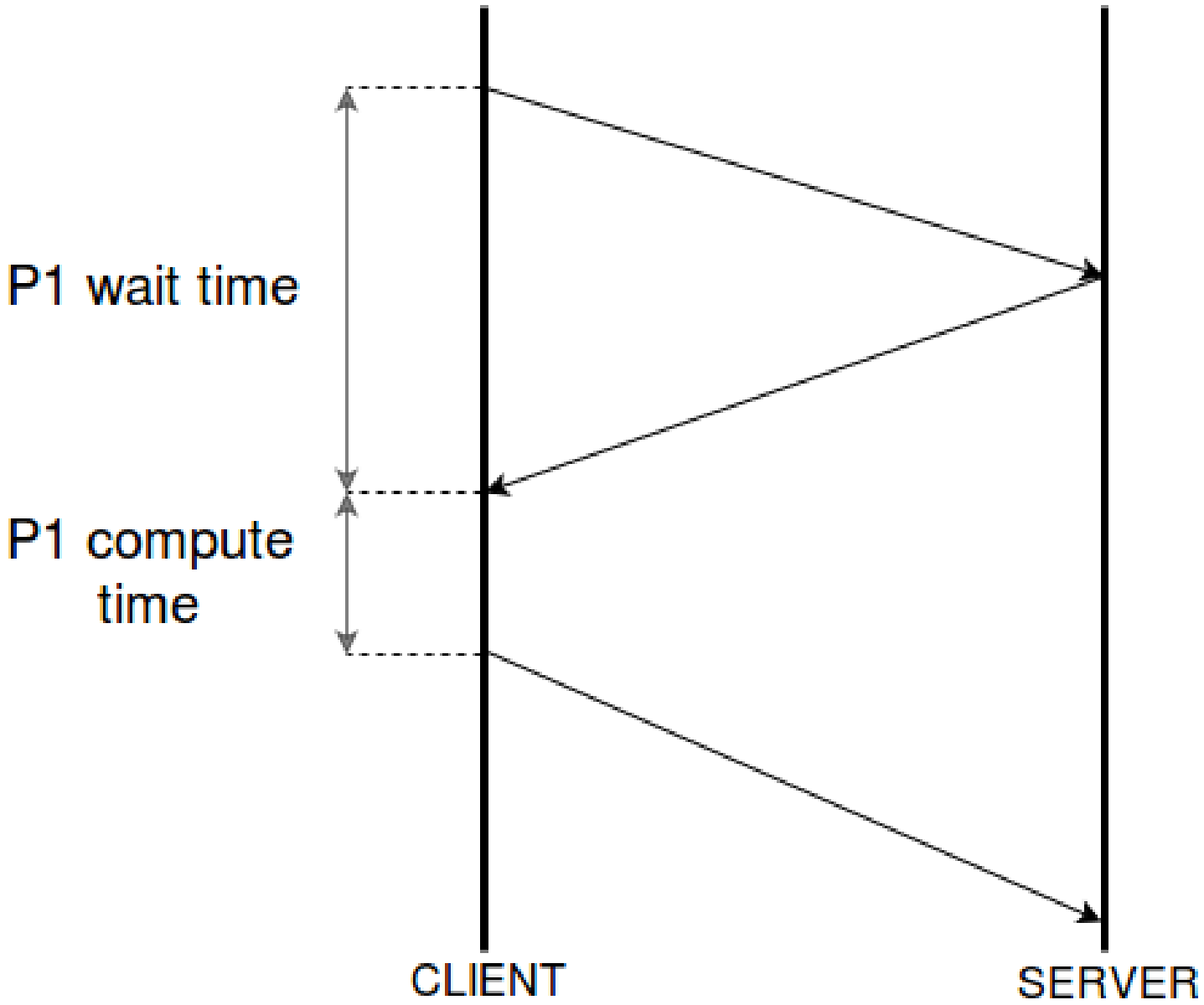
Research scope

Researching on potential I/O optimisation methods:

- prefetching (implies the usage of cache)
  - read-ahead
  - read-behind
- parallelism

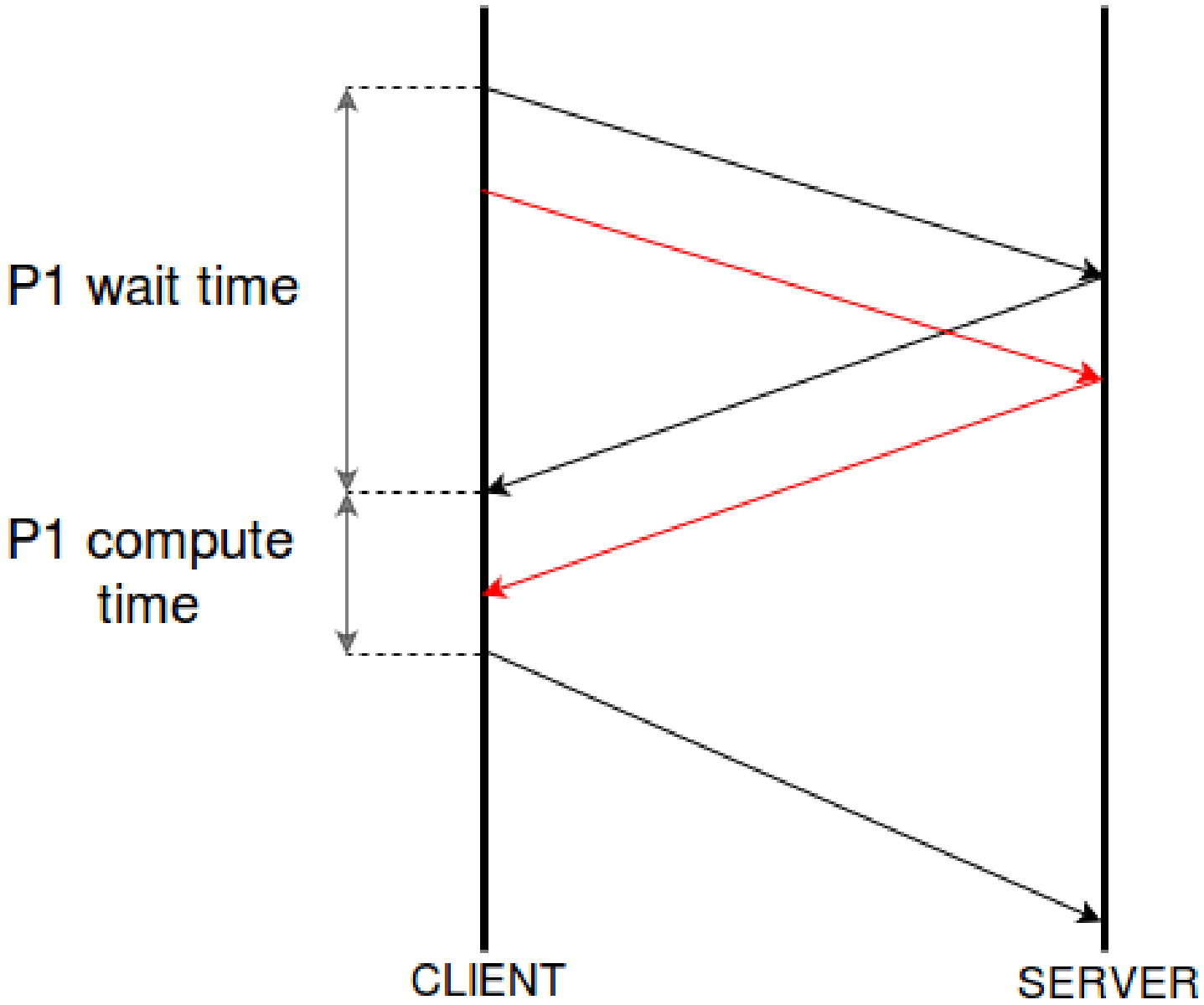
Research scope - parallelism

Single process, waiting for the reply

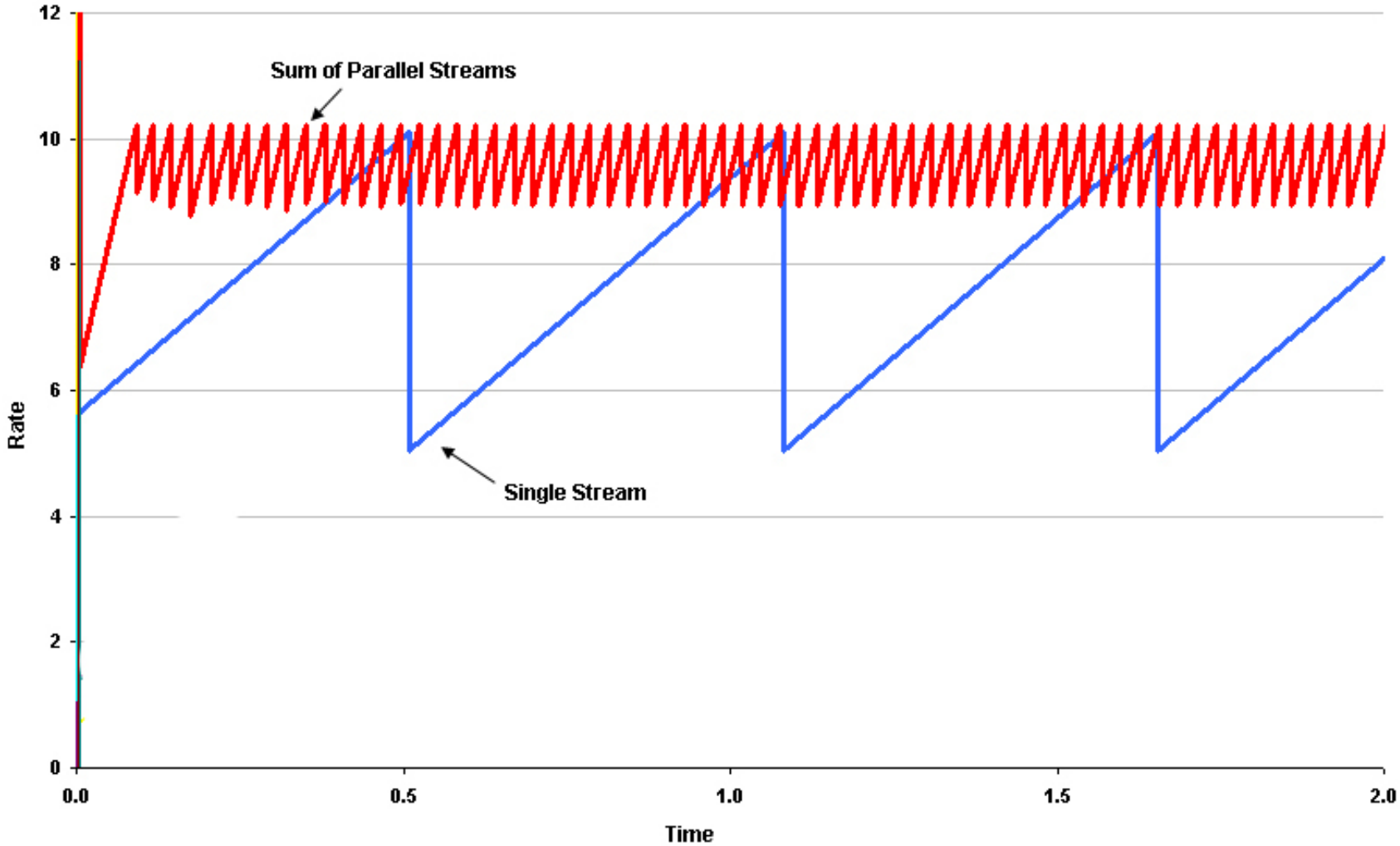


Research scope - parallelism

More processes, an attempt to utilise the wait time



Research scope - parallelism



Source : http://www.potaroo.net/ispcol/2005-06/fig4.jpg

Methods - creating `triage.py`

Goals:

Methods - creating `triage.py`

Goals:

- Repeatable triage process (tests)

Methods - creating `triage.py`

Goals:

- Repeatable triage process (tests)
- Two modes : sequential & parallel

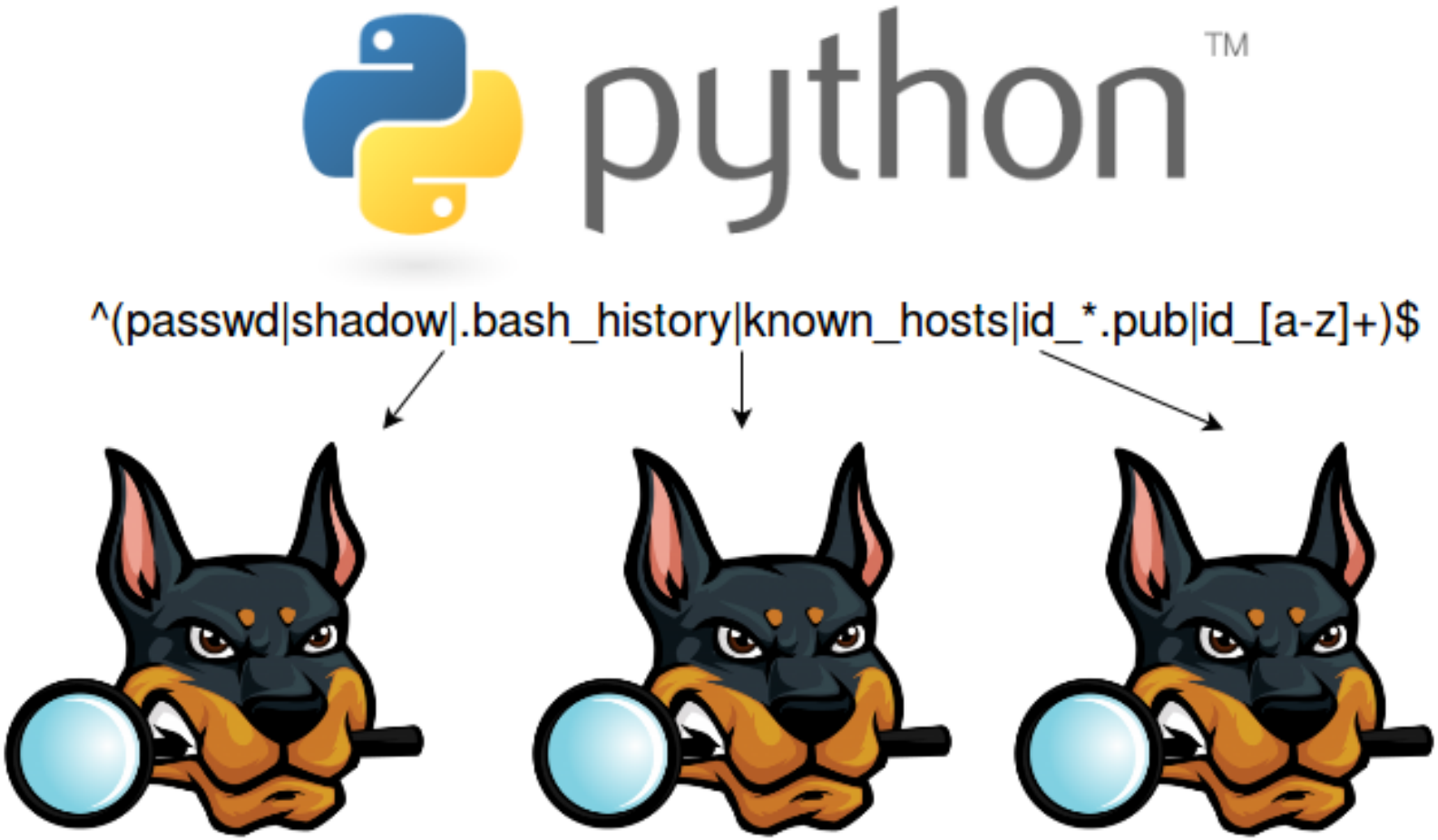
Methods - creating `triage.py`

Goals:

- Repeatable triage process (tests)
- Two modes : sequential & parallel
- Adjustable parallel workers number

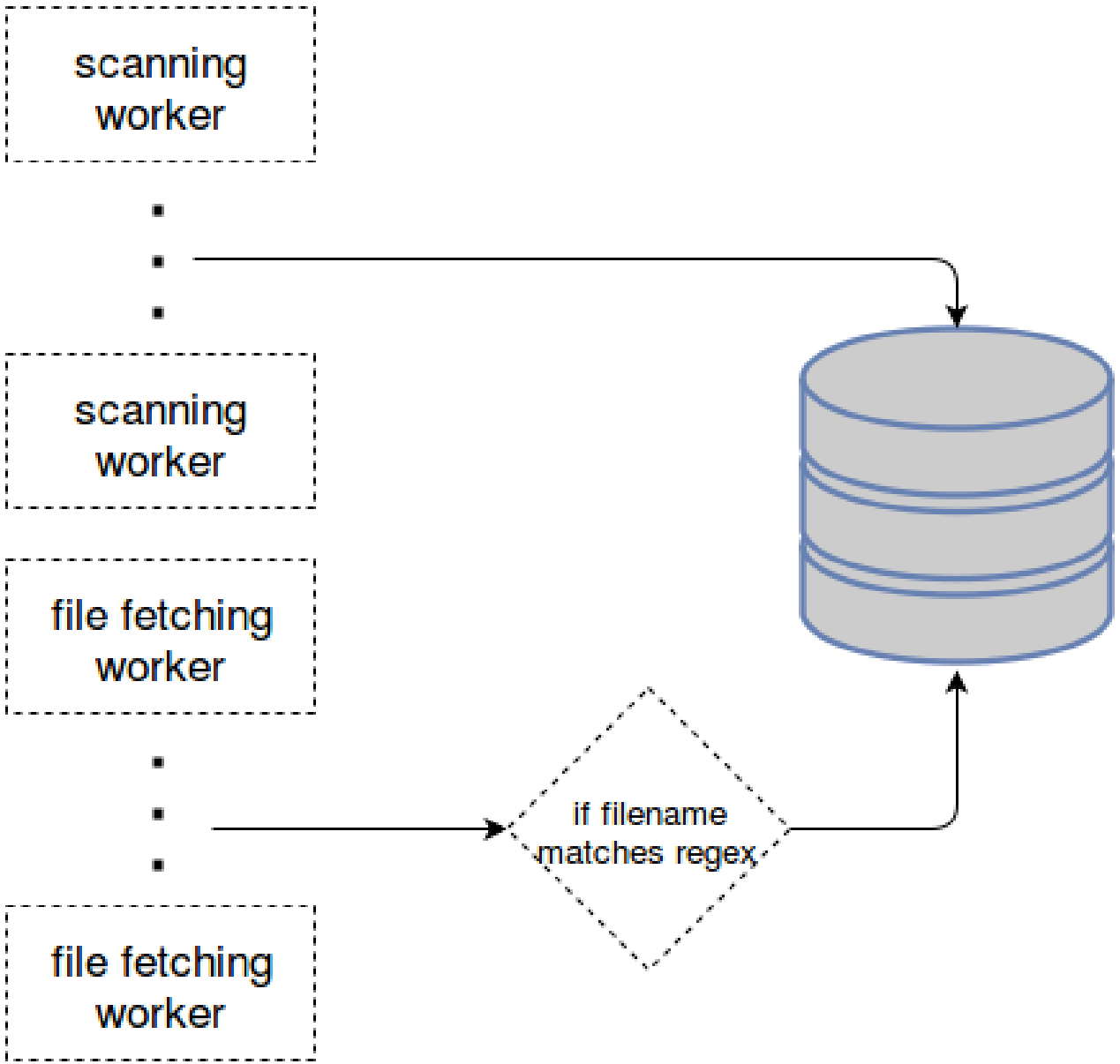
Methods - creating `triage.py`

Solution:



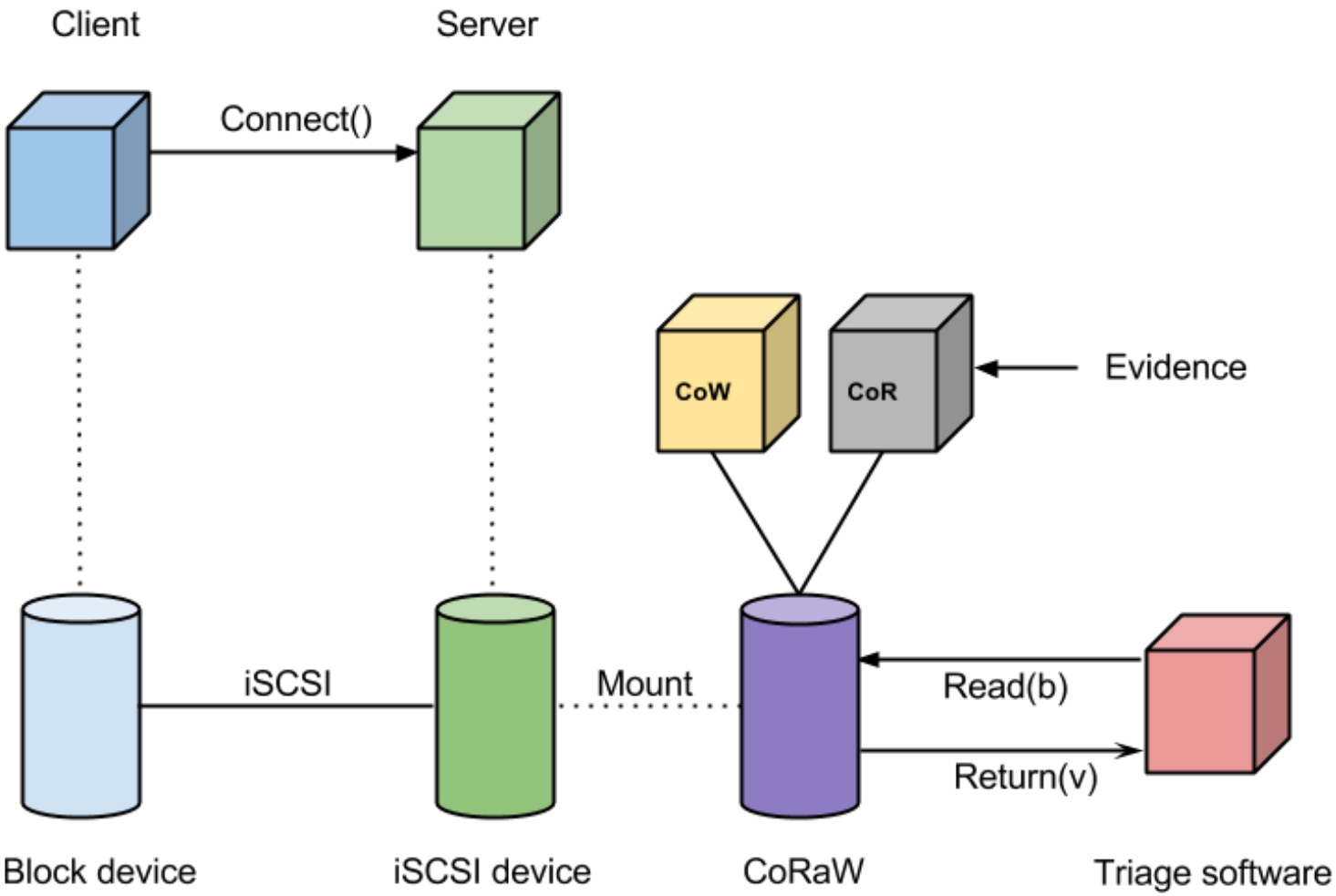
Methods - parallelism

Multiprocessing. Making The SleuthKit (TSK) parallel.



Methods - prefetching

Cache implementation : Fusecoraw[1]



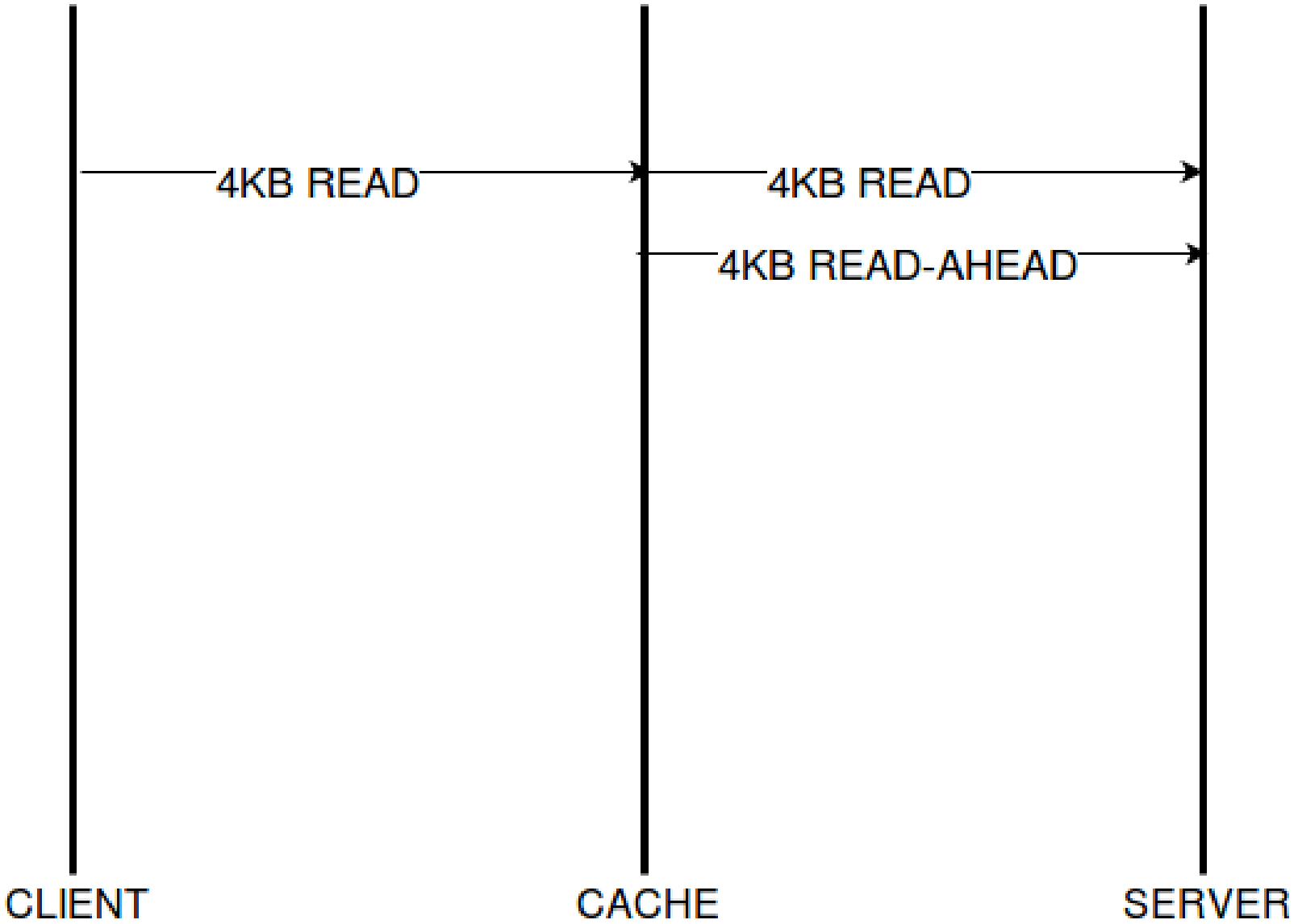
[1] https://homepages.staff.os3.nl/~delaat/rp/2013-2014/p71/report.pdf

Methods - prefetching

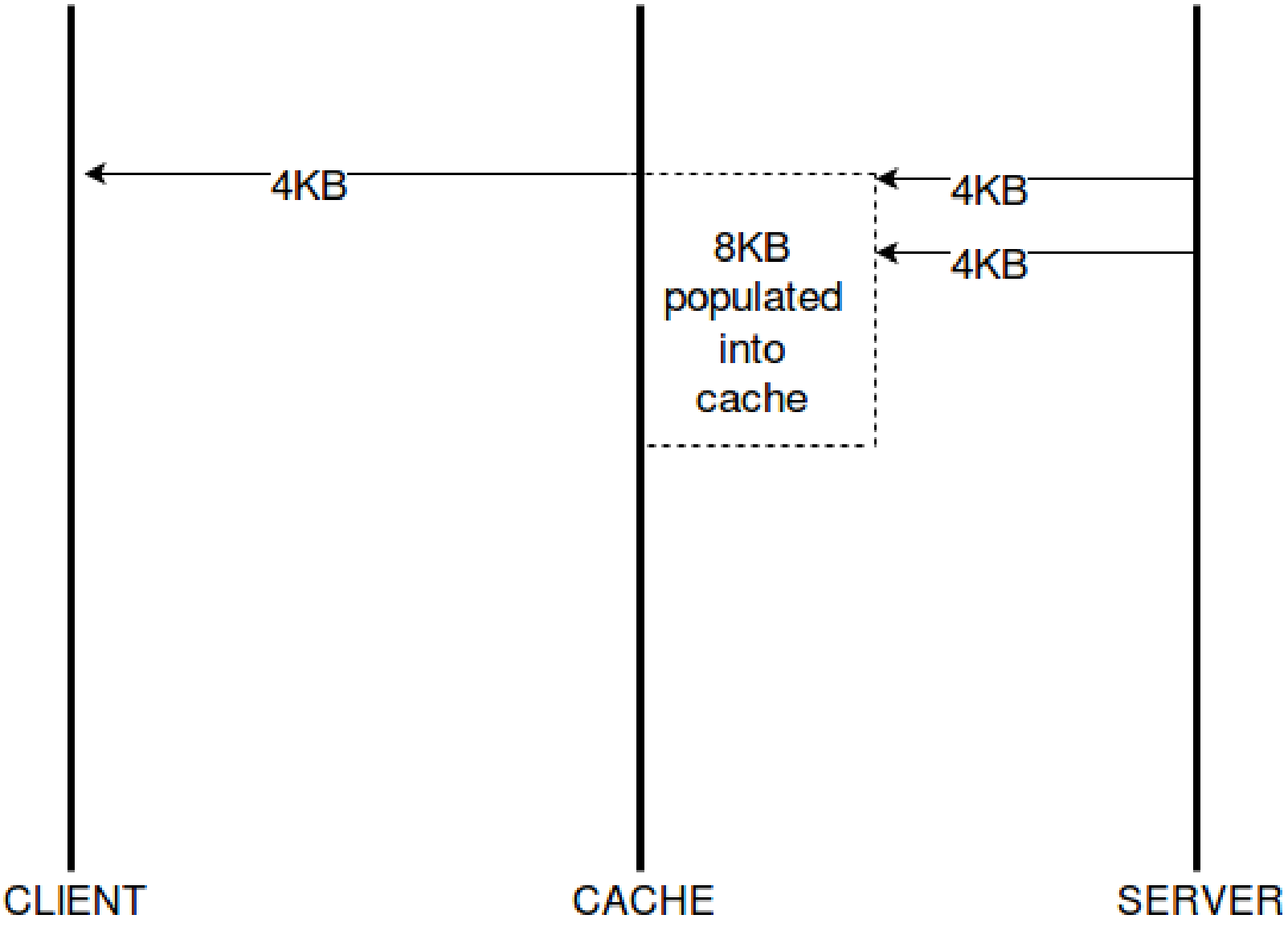
Expanding fusecoraw with read-ahead, read-behind functionality.
Simplified approach.

Methods - prefetching

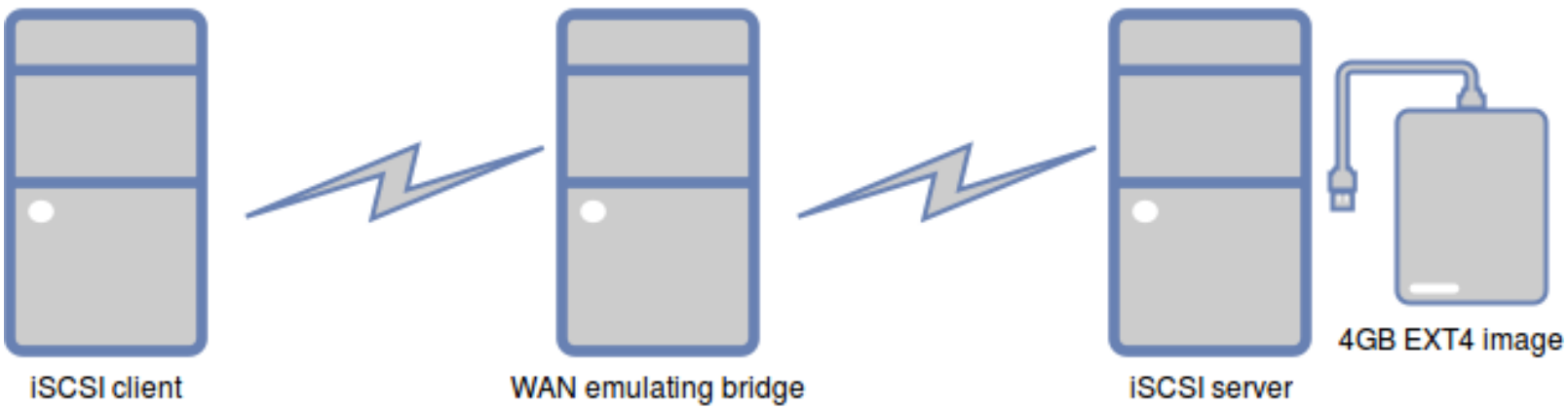
Reads issued to the FUSE filesystem are being extended by the additional `read()`.



Methods - prefetching

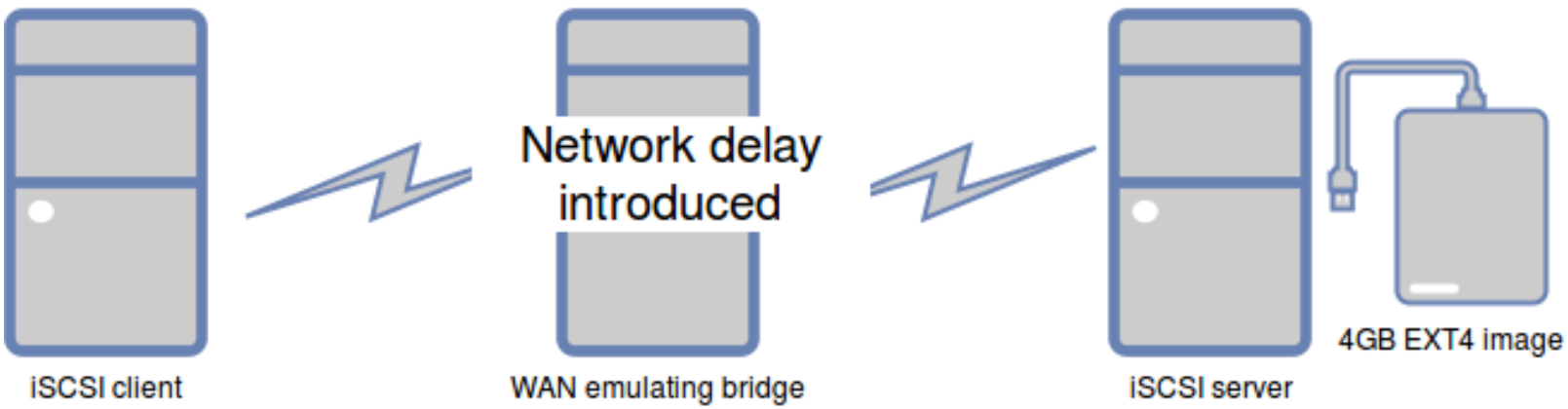


Methods - Lab setup



Methods - Lab setup

Constant delay applied : 0, 10, 20 [ms]



Experiments performed

test performed / relative delay (ms)	prefetching	parallelism	repetitions
0	X	X	3
10	X	X	3
20	X	X	3

Table : Test sets summary

Experiments performed

Chosen metrics:

- Average throughput (`tcpdump` + `tcptrace`)
- Elapsed time (`GNU time`)

Experiments performed

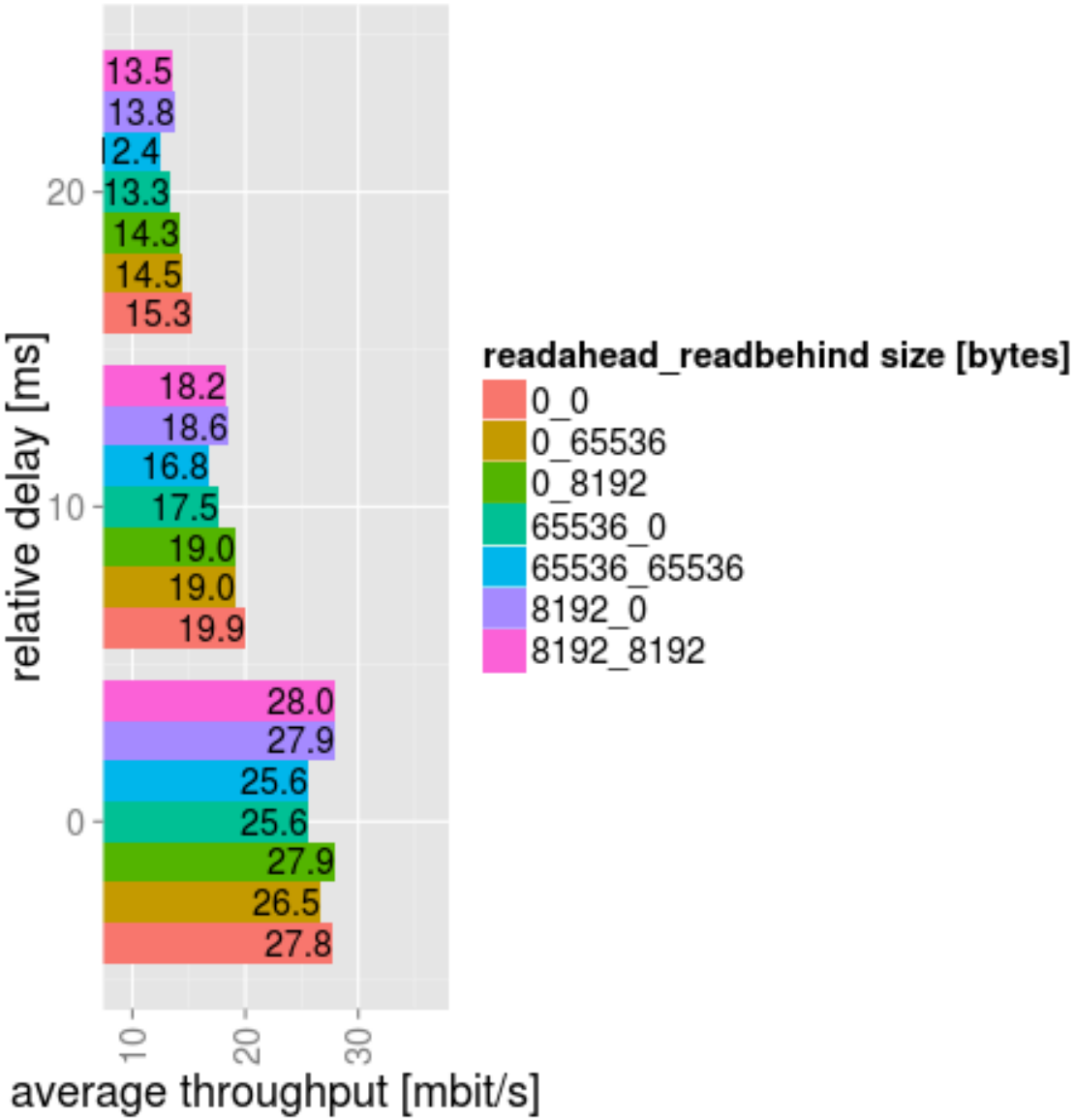
Prefetching

read ahead \ read behind	0	8192	65536
0	X	X	X
8192	X	X	-
65536	X	-	X

Table : Chosen read-ahead and read-behind values

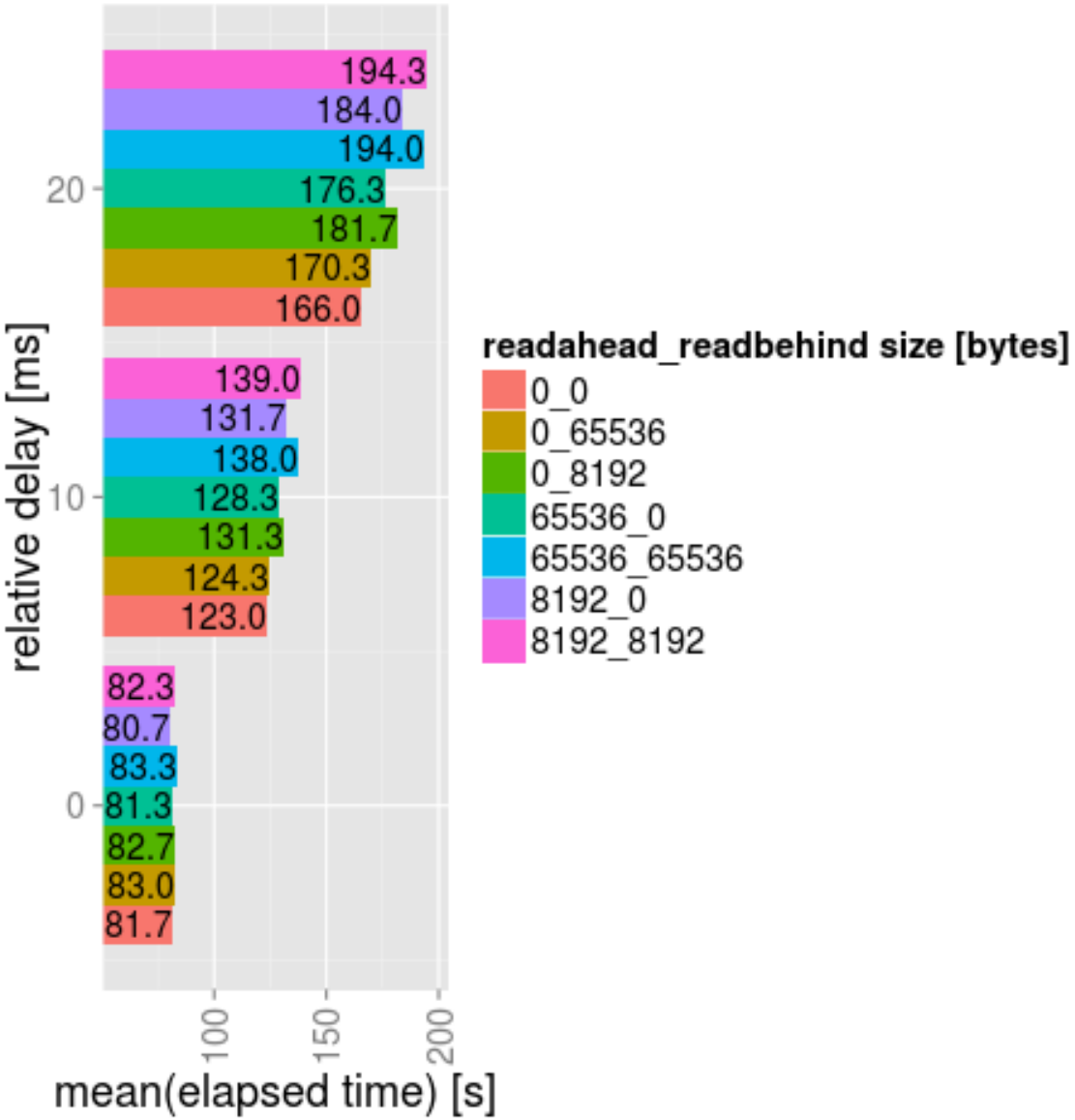
Results

Prefetching (Read-ahead & read-behind)



Results

Prefetching (Read-ahead & read-behind)



Results

Prefetching tests observations

Results

Prefetching tests observations

- Average throughput *may* indicate the triage process speed-up, but . . .

Results

Prefetching tests observations

- Average throughput *may* indicate the triage process speed-up, but . . .
- It's better to look at the execution time

Results

Prefetching tests observations

- Average throughput *may* indicate the triage process speed-up, but . . .
- It's better to look at the execution time
- When no delay was introduced; read-ahead of 8KiB, had the smallest mean execution time

Results

Prefetching tests observations

- Average throughput *may* indicate the triage process speed-up, but . . .
- It's better to look at the execution time
- When no delay was introduced; read-ahead of 8KiB, had the smallest mean execution time
- With the delay; I/O without prefetching had the smallest time metric

Experiments performed

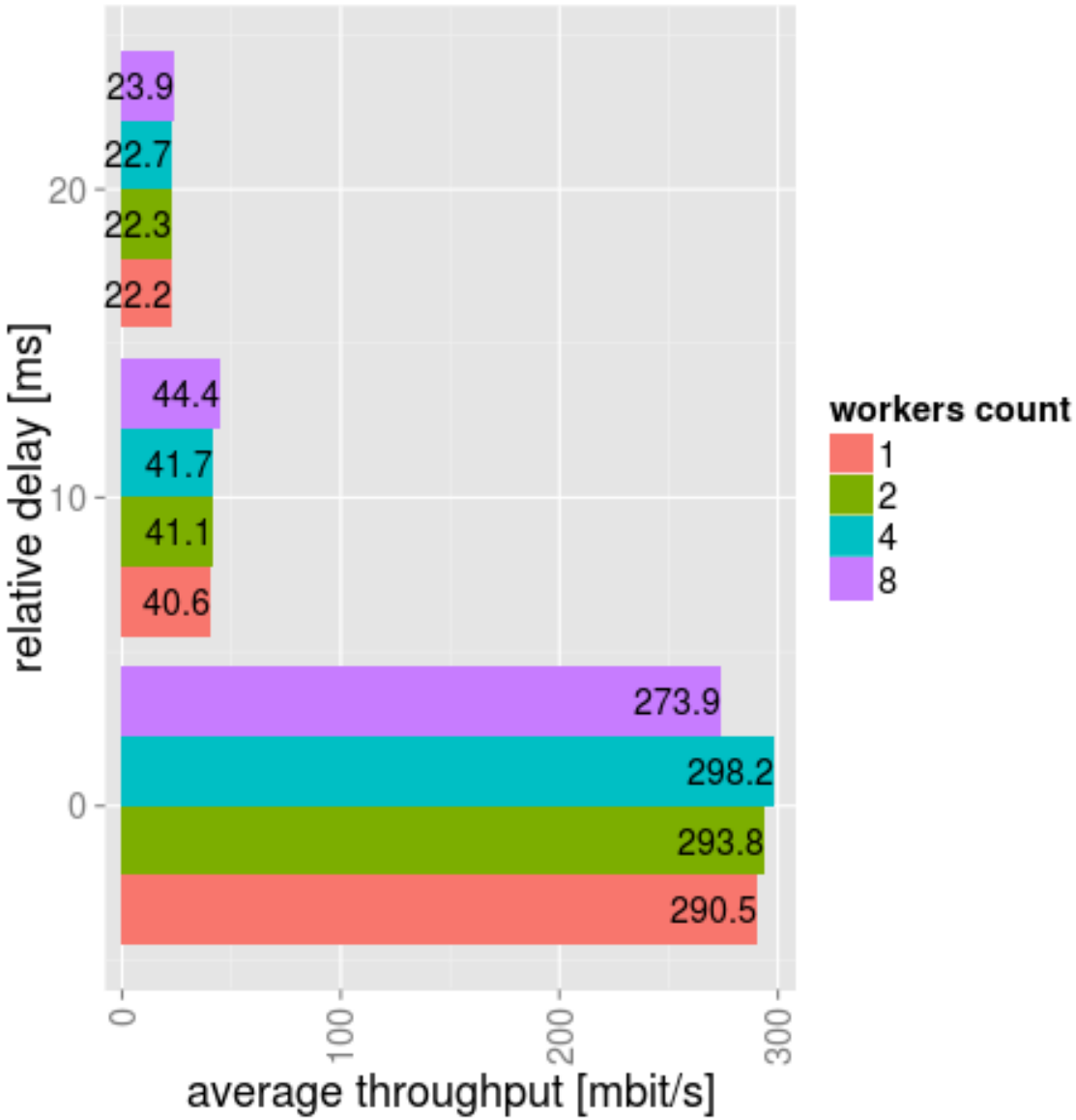
Parallelism

directory scanner \ file fetcher	1	2	4
1	X	-	-
2	-	X	-
4	-	-	X

Table : triage.py workers setup

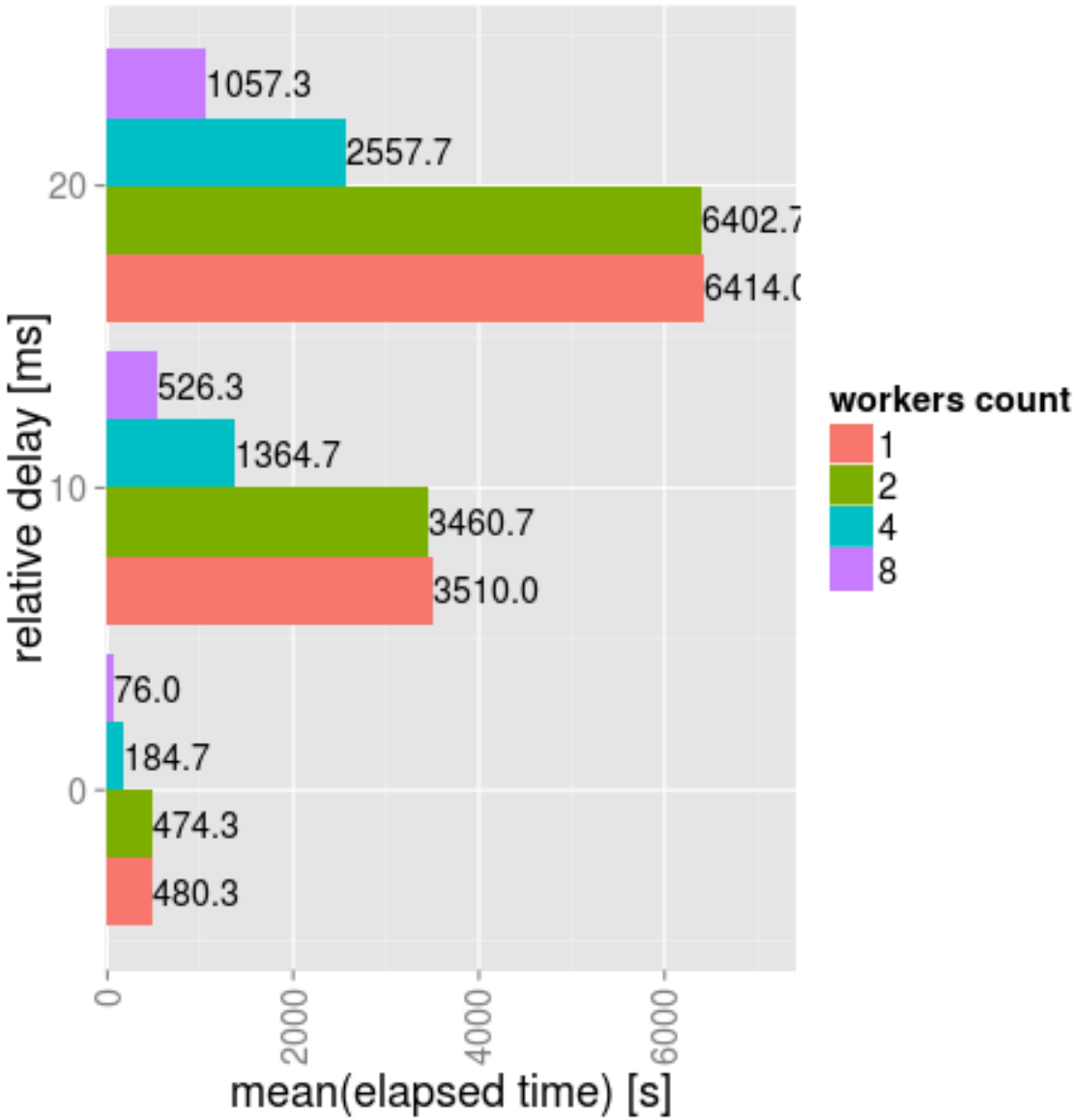
Results

Parallelism



Results

Parallelism



Results

Parallelism test observations

Results

Parallelism test observations

- Elapsed time barchart suggests that 8 workers perform surprisingly well for the delayed link

Results

Parallelism test observations

- Elapsed time barchart suggests that 8 workers perform surprisingly well for the delayed link
- However, the throughput chart does not record expected speed-up (the differences are small)

Results

Parallelism test observations

- Elapsed time barchart suggests that 8 workers perform surprisingly well for the delayed link
- However, the throughput chart does not record expected speed-up (the differences are small)
- Probably the external factor which influenced the test occurred (caching?)

Lessons learnt

Lessons learnt

- OS tries to be your best friend. It optimises/caches whenever it can. Not necessarily bad, but it has to be understood while designing the tests.

Lessons learnt

- OS tries to be your best friend. It optimises/caches whenever it can. Not necessarily bad, but it has to be understood while designing the tests.
- Trying to abstract the research from the components it will eventually need to rely on, is close to agreeing that its results may become "abstract".

Future work

Future work

- Follow up on the I/O optimisation techniques (extend presented tests)

Future work

- Follow up on the I/O optimisation techniques (extend presented tests)
- Try to reuse tuning knowledge from the papers which investigated iSCSI sequential writes over the delayed links

Future work

- Follow up on the I/O optimisation techniques (extend presented tests)
- Try to reuse tuning knowledge from the papers which investigated iSCSI sequential writes over the delayed links
- Assess chosen iSCSI implementation against *Analysis of iSCSI Short Blocks Access* paper criteria

Future work

- Follow up on the I/O optimisation techniques (extend presented tests)
- Try to reuse tuning knowledge from the papers which investigated iSCSI sequential writes over the delayed links
- Assess chosen iSCSI implementation against *Analysis of iSCSI Short Blocks Access* paper criteria
- Is getting the work done without TCP possible? Exploring ATA over Ethernet (AoE) feasibility for the remote acquisition

Q&A

Questions?