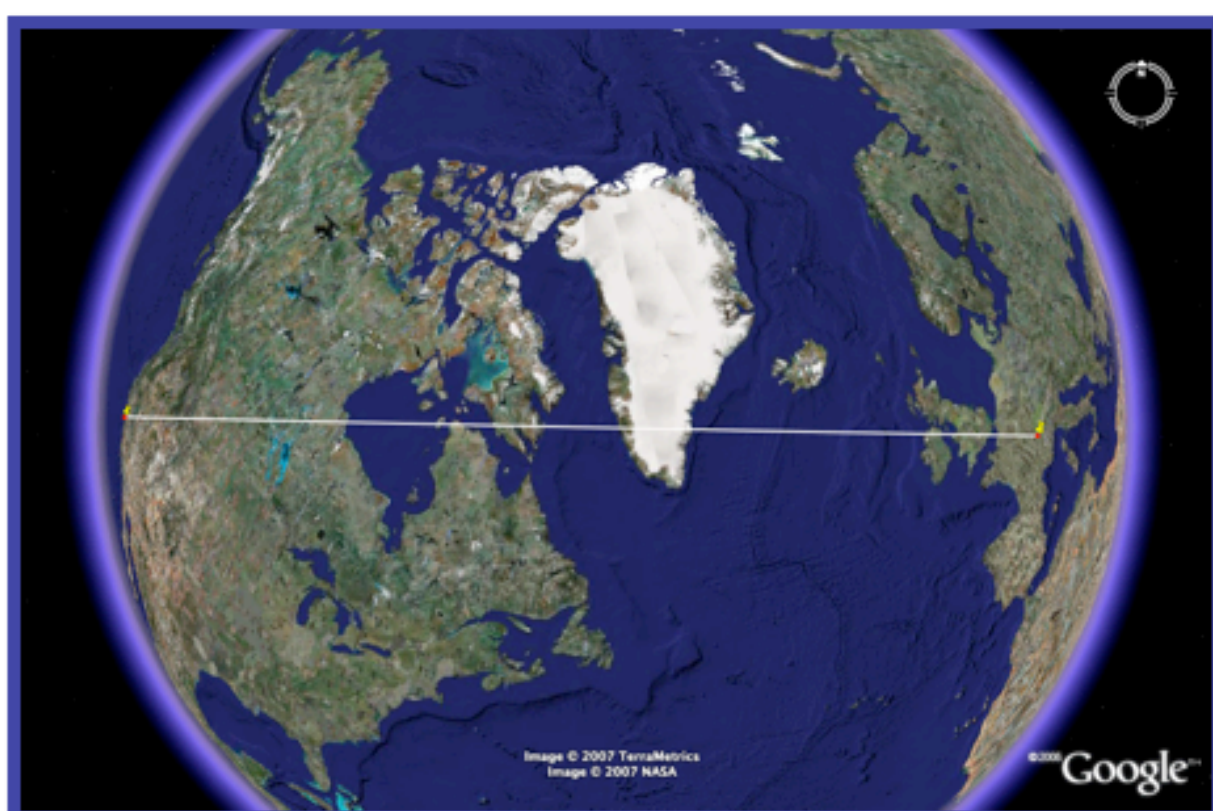




Intelligent Pre-fetching

Current transfer protocols are tuned to work with large pipelined data buffers where bandwidth is the key parameter. On the other hand, interactive data analysis requires access to scattered records in remote files and latency is the main factor. To alleviate the latency problem, an efficient pre-fetching/cache algorithm has been implemented in recent versions of ROOT.

High latency and numerous reads create a problem



Remote access example

- Distance CERN-SLAC: 9393Km
- Maximum speed: 2.9×10^6 Km/s
- Lowest latency (RTT): 62.6 ms
- Measured latency : 166 ms

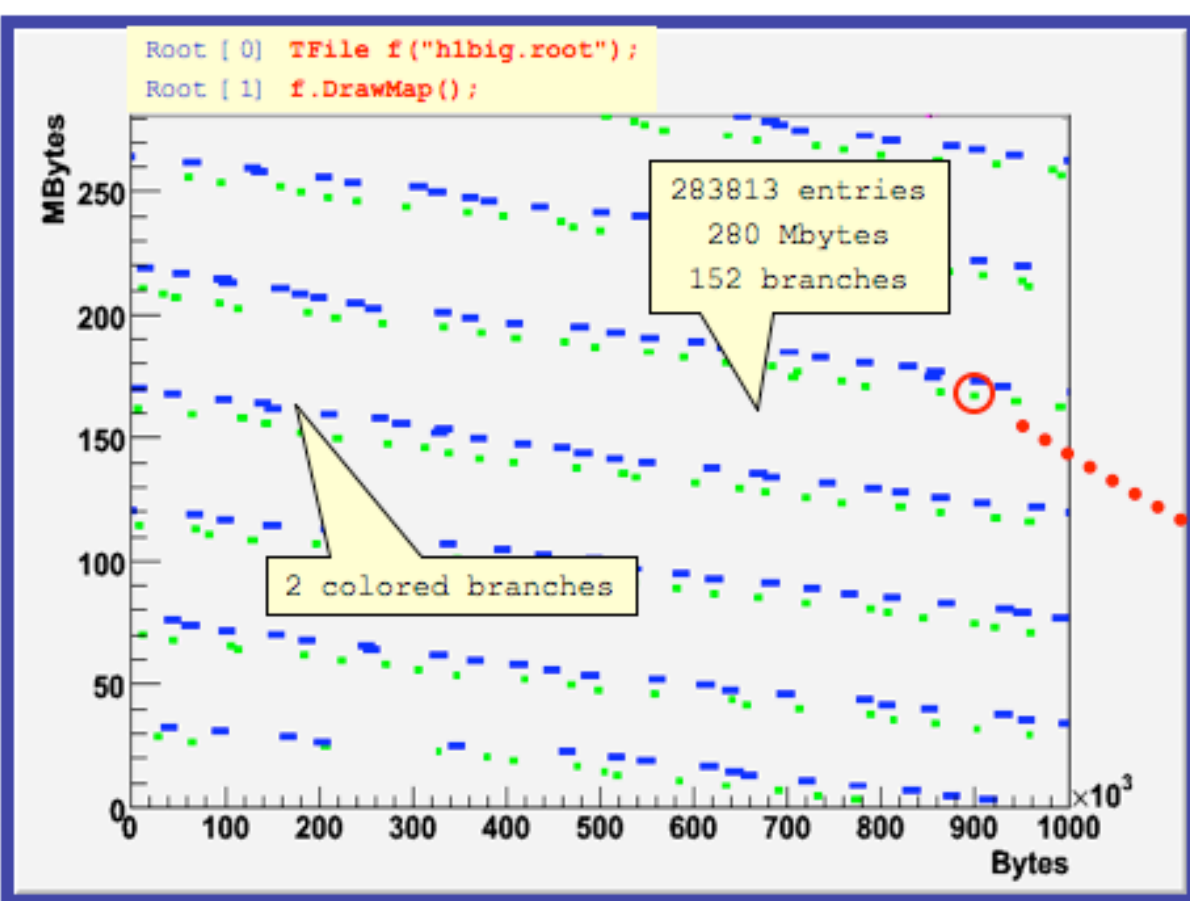
Latency is proportional to distance and can not be reduced!!!

Scattered reads

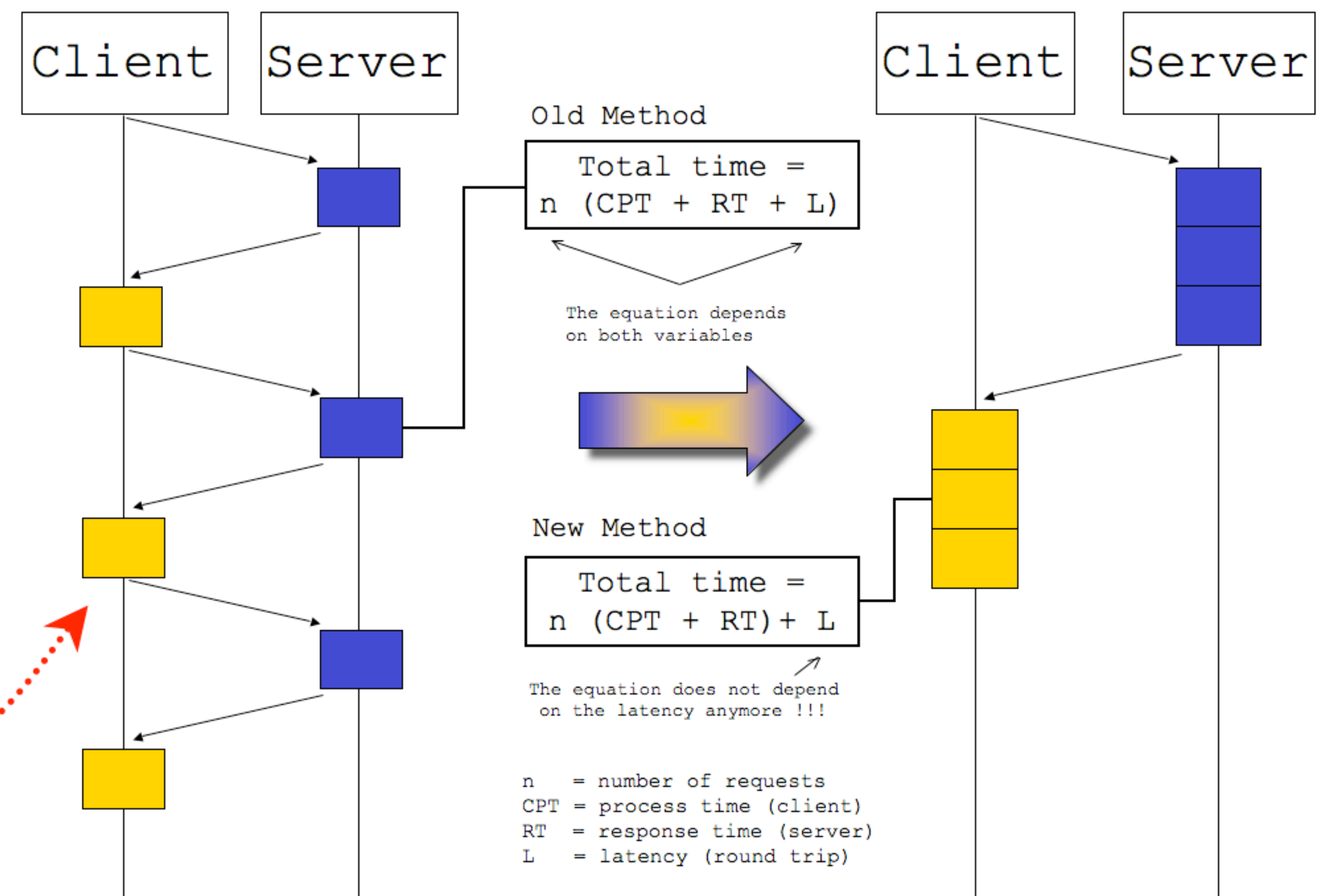
Trees represent data in a very efficient way

- Data is grouped in branches.
- We read only subsets of the branch buffers.

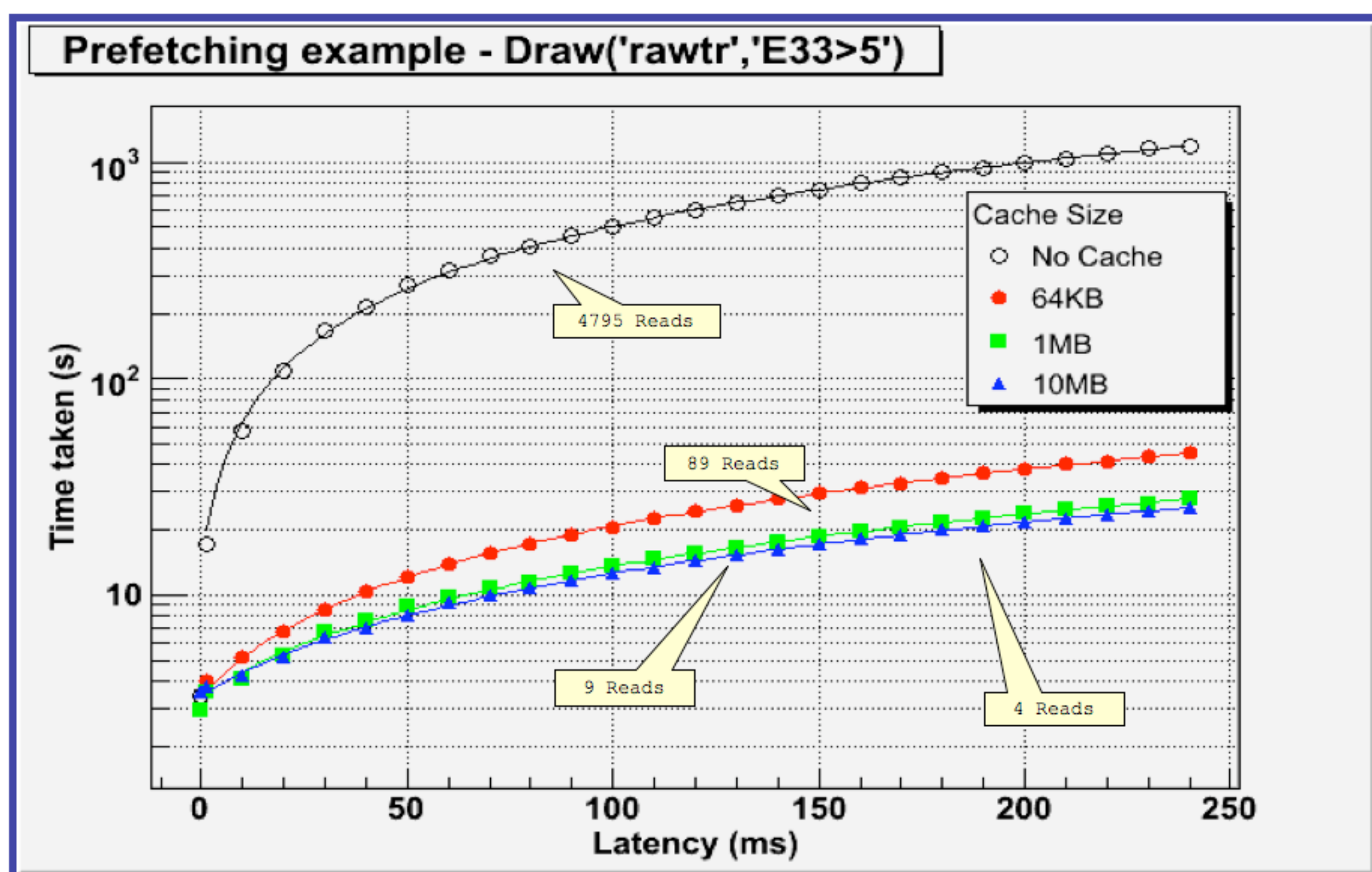
We need buffers that are not contiguous but we know their position and size.



Read multiple buffers with one request



A simulation with a simple example



Data access can be between 10 and 100 times faster!!!

A real-life measurement

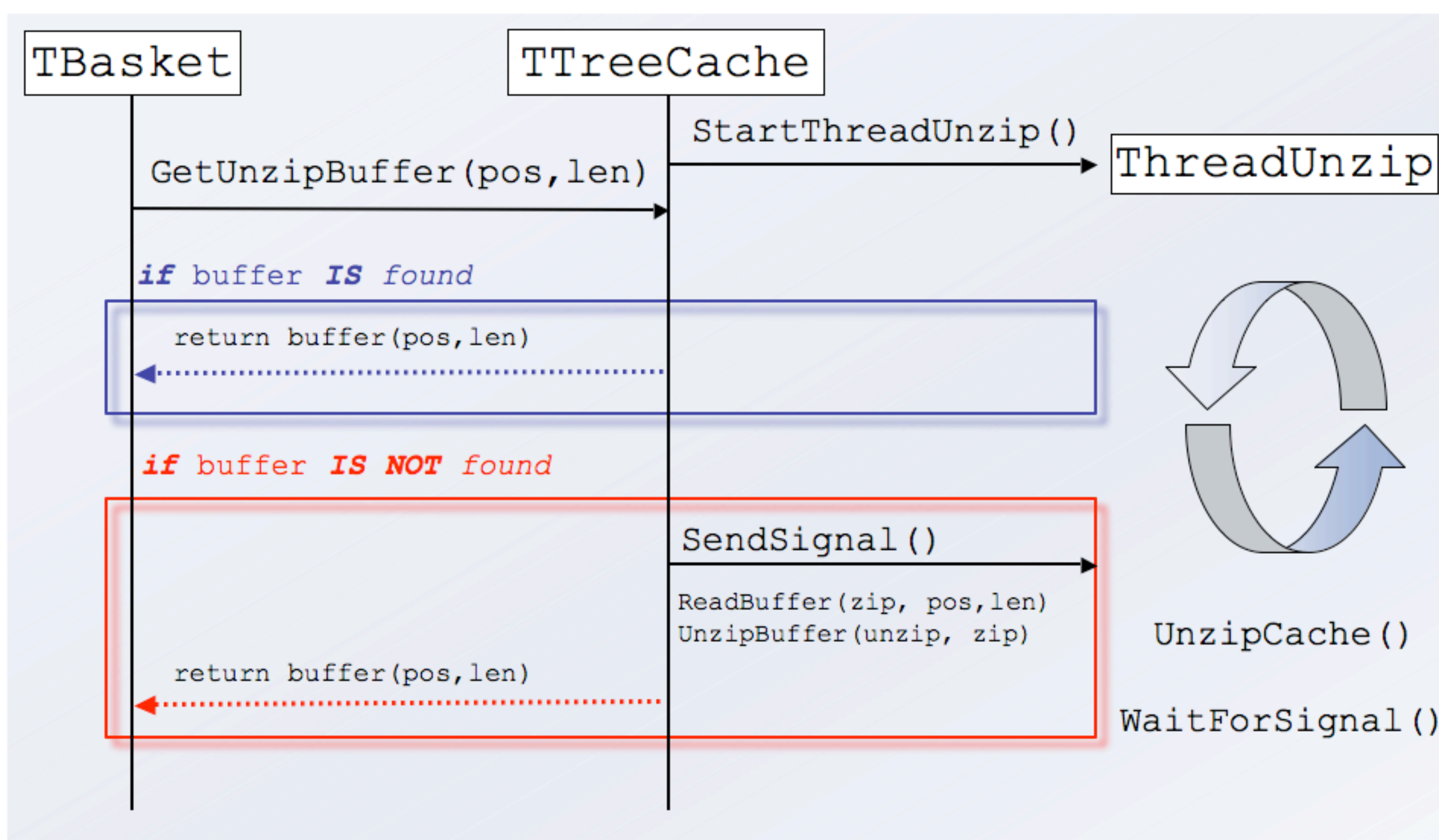
- The file is on a CERN machine (LAN 100MB/s).
- A is a local read (same machine).
- B is on a LAN (100 Mb/s - PIV 3 Ghz).
- C is on a wireless network (10Mb/s - Mac duo 2Ghz).
- D is in Orsay (LAN 100Mb/s, WAN 1Gb/s - PIV 3 Ghz).
- E is in Amsterdam (LAN 100Mb/s, WAN 10Gb/s-AMD64).
- F is on ADSL (8Mb/s - Mac duo 2Ghz).
- G is in Caltech (10Gb/s).
- The results are in real-time seconds

	Latency (ms)	0KB	64KB	10MB
A	0.0	3.4	3.4	3.4
B	0.3	8.0	6.0	4.0
C	2.0	11.6	5.6	4.9
D	11.0	124.7	12.3	9.0
E	22.0	230.9	11.7	8.4
F	72.0	743.7	48.3	28.0
G	240.0	>1800.0	125.4	9.9*

* TCP/IP Jumbo frames (9KB) and a TCP/IP window size of 10 Mbytes. We hope to bring down this number to about 4 seconds by reducing the number of transactions when opening the file.

Parallel Unzipping

Taking advantage of multi-core machines



Since we know which buffers to read, an additional thread can unzip them in advance.

Balance between:

- Size of the buffer
- Performance gain
- Number of cache misses

Ideal size:

- Around 10%

Gain (overall):

- Close to 13%

