MADALGO
CENTER FOR MASSIVE DATA ALGORITHMICS

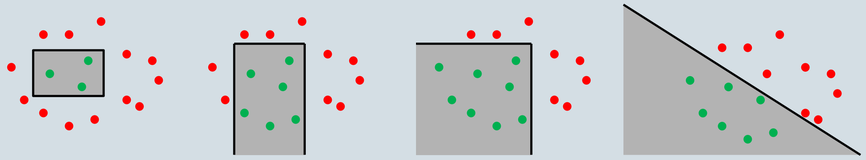

Peyman Afshani
Aarhus University

Cache-oblivious Geometric Range Reporting

Range Reporting

Range reporting is a fundamental problem in computational geometry.

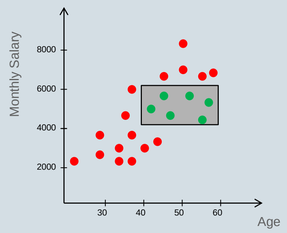
In the problem, the goal is to store a set of *n* points in a data structure to allow users to ask for the subset of points that is contained in a geometric query region.



orthogonal range reporting | *three-sided reporting* | *dominance reporting* | *halfspace range reporting*

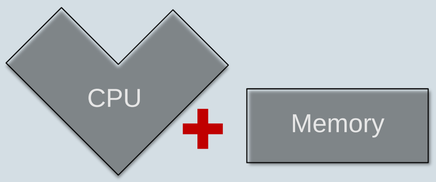
The variants of range reporting problem depend on the shape of the query region.

Finding all the cities within 60km of Århus

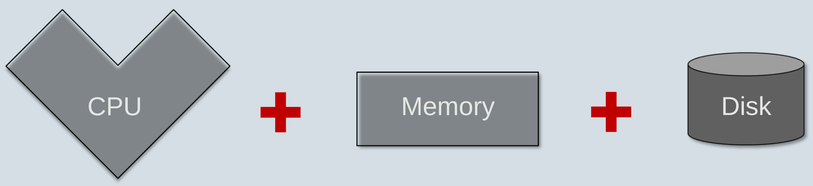


Finding all the 40 to 60 years old employees who earn between 4000 and 6000

Range reporting has many applications in various fields, such as geographical information systems and databases.



The problem was traditionally studied in main memory models which represent the computer as being composed of a main memory and a processing unit.



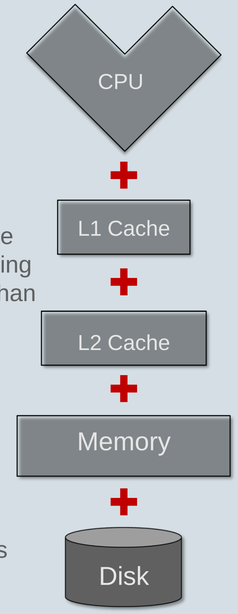
The *I/O model* tries to reflect the architecture of the computers more accurately by including a "disk" which is a slow but conceptually unlimited storage facility. It assumes communication between the disk and the main memory is done using blocks of size *B*. This became a successful model, however, the increased complexity of the model made designing algorithms more difficult.

The Cache-oblivious Model

The cache-oblivious model can model an arbitrary number of memory levels, thus, it captures the architecture of the modern computers more accurately.

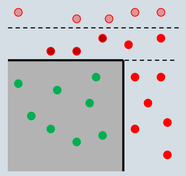
The algorithms are designed in a main memory model, while the analysis is done in the I/O model. Because of this, designing algorithms in this model is much easier than other multi-level models.

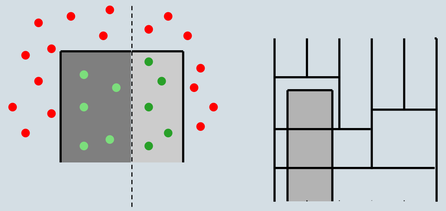
As the algorithm is oblivious to the memory hierarchy, the analysis can be extended to *any* two levels of memory hierarchy, meaning, if it performs well for two levels of memory, then it performs equally well on *all* levels of memory.



Cache-oblivious Range Reporting

Dominance reporting was solved optimally in two-dimensions, using O(*n*) space. The queries could be answered with $O(\lg_B n + K/B)$ I/Os [Arge, Zeh'06]. *K* is number points in the output.





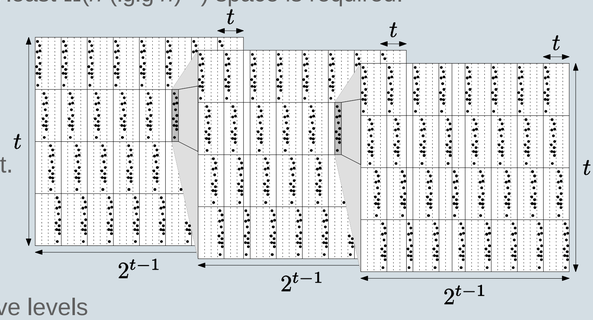
It was not known how to solve three-sided queries optimally, despite many attempts [Agarwal et al.'03, Arge et al.'05, Arge, Zeh'06]. The query bound of $O(\lg_B n + K/B)$ could be achieved with O(*n* lg *n*) space.

In the I/O model, they could be answered with O(*n*) space and optimal query time, and finding an equivalent data structure with O(*n*) space in the cache-oblivious model was left open.

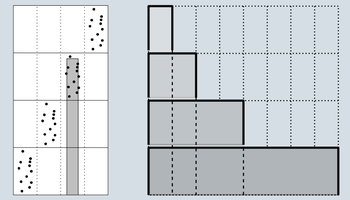
Lower Bound for Three-sided Queries

We proved that it is impossible to answer queries with $O(\lg_B n + K/B)$ I/Os using linear space. We show that at least $\Omega(n\ (\lg\lg n)^{\varepsilon})$ space is required.

We first create a set of points using a highly recursive layout.



Queries from different recursive levels do not intersect.

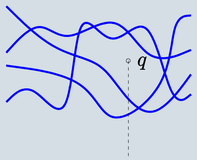


The lower bound is proved using various combinatorial properties of the queries and the points, combined with a recursive boosting argument.

This provides a separation result between the cache-oblivious and I/O models.

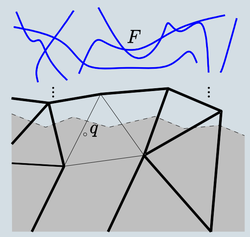
A Framework for Cache-oblivious Range Reporting

We study the general problem of processing a set *F* of functions in a data structure such that the functions passing below a query point *q* can be reported efficiently. Three-sided queries, 3-d dominance queries, 3-d halfspace queries, and 2-d circular queries can be modeled in this framework.



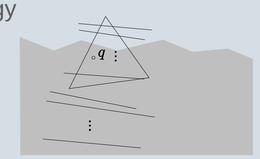
It turns out, we must solve the approximate counting problem in the cache-oblivious model first.

Shallow cuttings are powerful tools used frequently in halfspace range reporting. We employ them in a novel way, improving all the previous results on many problems, including approximate halfspace range counting.



Combining various techniques with a clever recursive strategy we obtain optimal approximate counting data structures.

This gives efficient cache-oblivious data structures that use O(*n* lg *n*) space and with optimal query bounds.



A very interesting problem is to find the right space bound. Our results imply this is between $\Omega(n\ (\lg\lg n)^{\varepsilon})$ and O(*n* lg *n*).