5-2011

Towards Optimal Knowledge Processing: From Centralization Through Cyberinsfrastructure To Cloud Computing

Octavio Lerma

Eric Gutierrez
The University of Texas at El Paso, ejgutierrez@miners.utep.edu

Chris Kiekintveld
The University of Texas at El Paso, cdkiekintveld@utep.edu

Vladik Kreinovich
The University of Texas at El Paso, vladik@utep.edu

Comments:
Technical Report: UTEP-CS-11-20
Published in *International Journal of Innovative Management, Information & Production (IJIMIP)*, 2011, Vol. 2, No. 2, pp. 67-72.

Towards Optimal Knowledge Processing: From Centralization Through Cyberinsfrastructure To Cloud Computing

Octavio Lerma[1], Eric Gutierrez[2],
Chris Kiekintveld[2], and Vladik Kreinovich[1,2]

[1]Computational Sciences Program
[2]Department of Computer Science
University of Texas at El Paso
500 W. University
El Paso, TX 79968, USA
lolerma@episd.org, ejgutierrez@miners.utep.edu,
cdkiekintveld@utep.edu, vladik@utep.edu

Abstract

One of the most efficient way to store and process data is *cloud computing*, when we store the data so as to minimize the expenses and increase the efficiency. In this paper, we provide an analytical solution to the corresponding optimization problem.

1 Cloud Computing: Why We Need It and How Can We Make It Most Efficient

Why cloud computing. In many application areas (bioinformatics, geosciences, etc.) we need to process large amounts of data, which requires fast computers and fast communication. Historically, there have been limits on the amount of the information that can be transmitted at a high speed, and these limits affected information processing.

A few decades ago, we could only send the results of data processing fast. As a result, the best strategy to speed up computations was to move all the data into a central location, close to the high performance computers for processing this data.

In the last decades, it became equally fast to move big portions of databases needed to answer a certain query. This enabled the users to switch to a *cyberinfrastructure* paradigm, when there is no longer need for time-consuming moving of data to a central location: the data is stored where it was generated, and

when needed, the corresponding data is moved to processing computers; see, e.g., [2, 3, 6, 7, 8] and references therein.

Nowadays, moving the whole databases becomes almost as fast, so there is no longer need to store the data where it was produced – it is possible to store the data where it will be best for future data processing. This idea underlies the paradigm of *cloud computing*.

What is the most efficient way of cloud computing. The main advantage of cloud computing is that we can make computations more efficient by finding optimal placement of the servers that store and/or process the corresponding data. So, in developing cloud computing schemes, it is important to be able to solve this optimization problem. In this paper, we consider the corresponding problem of optimal data storage in cloud computing.

Comment. This server placement problem is very similar to the type of problems faced by Akamai and other companies that do web acceleration via caching; we therefore hope that our solution can be of help in web acceleration as well.

2 Towards a Precise Formulation of the Problem: First Approximation

What we want and what we need. We usually *know* the geographic density $\rho_u(x)$ describing possible users of this particular database (e.g., a database containing geophysical data), and we know the number of duplicates D that we can afford to store. We *need to determine* the storage density $\rho_s(x)$, i.e., number of copies per geographic region, so as to minimize the average communication delay.

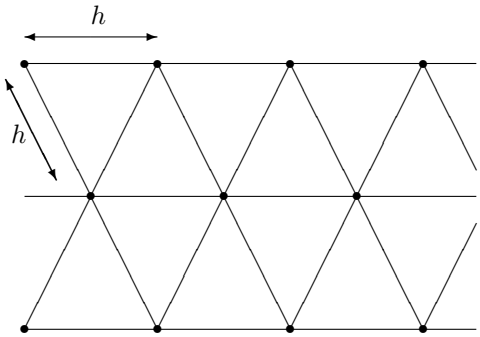
First approximation model: main assumption. In the first approximation, we can measure the travel delay by the average travel distance.

Derivation of the corresponding model. How can we describe this distance in terms of the density $\rho_s(x)$? When the density is constant, we want to place the servers in such a way that the largest distance r to a sensor is as small as possible. (Alternatively, if r is fixed, we want to minimize the number of servers for which every point is at a distance $\le r$ from one of the servers. In geometric terms, this means that every point on a plane belongs to a circle of radius r centered on one the sensors – and thus, the whole plane is covered by such circles. Out of all such coverings, we want to find the covering with the smallest possible number of sensors.

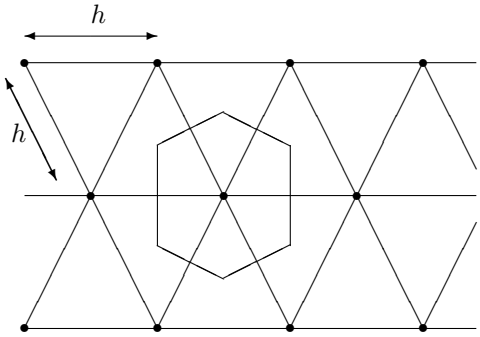
It is known that the smallest such number is provided by an equilateral triangle grid, i.e., a grid formed by equilateral triangles; see, e.g., [4, 5].

Let us assume that we have already selected the server density function $\rho_s(x)$. Within a small region of area A, we have $A \cdot \rho_s(x)$ servers. Thus, if we,

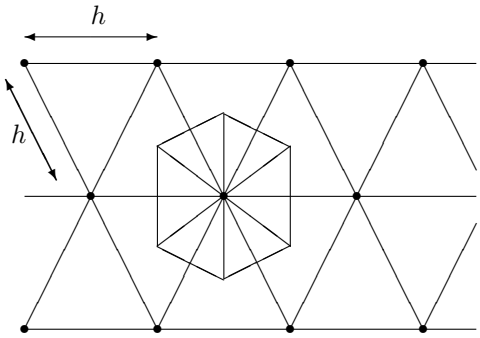
e.g., place these servers on a grid with distance h between the two neighboring ones in each direction, we have:



For this placement, the set of all the points which are closest to a given detector forms a hexagonal area:



This hexagonal area consists of 6 equilateral triangles with height $h/2$:



In each triangle, the height $h/2$ is related to the size s by the formula

$$\frac{h}{2} = s \cdot \cos(60^\circ) = s \cdot \frac{\sqrt{3}}{2},$$

hence

$$s = \frac{h}{\sqrt{3}} = h \cdot \frac{\sqrt{3}}{3}.$$

Thus, the area A_t of each triangle is equal to

$$A_t = \frac{1}{2} \cdot s \cdot \frac{h}{2} = \frac{1}{2} \cdot \frac{\sqrt{3}}{3} \cdot \frac{1}{2} \cdot h^2 = \frac{\sqrt{3}}{12} \cdot h^2.$$

So, the area A_s of the whole set is equal to 6 times the triangle area:

$$A_s = 6 \cdot A_t = \frac{\sqrt{3}}{2} \cdot h^2.$$

Each point from the region is the closest to one of the points from the server grid, so the region of area A is thus divided into $A \cdot \rho_s(x)$ (practically) disjoint sets of area $\frac{\sqrt{3}}{2} \cdot h^2$. So, the area of the region is equal to the sum of the areas of these sets:

$$A = (A \cdot \rho_s(x)) \cdot \frac{\sqrt{3}}{2} \cdot h^2.$$

Dividing both sides of this equality by A, we conclude that

$$1 = \rho_s(x) \cdot \frac{\sqrt{3}}{2} \cdot h^2,$$

and hence, that

$$h = \frac{c_0}{\sqrt{\rho_s(x)}},$$

where we denote

$$c_0 \stackrel{\text{def}}{=} \sqrt{\frac{2}{\sqrt{3}}}.$$

The largest distance r to a server is thus equal to

$$\frac{h}{2} = \frac{c_0}{2 \cdot \sqrt{\rho_s(x)}}.$$

The average distance $\overline{\rho}$ is proportional to r – since when we re-scale the picture, all the distances –including the average distance – increase proportionally. Since the distance r is proportional to $(\rho_s(x))^{-1/2}$, the average distance near the location x is thus also proportional to this same value: $\overline{\rho}(x) = \text{const} \cdot (\rho_s(x))^{-1/2}$ for some constant.

At each location x, we have $\sim \rho_u(x)$ users. Thus, the total average distance – the value that we would like to minimize – is equal to $\int \overline{\rho}(x) \cdot \rho_u(x)\, dx$ and is, thus, proportional to

$$\int (\rho_s(x))^{-1/2} \cdot \rho_u(x)\, dx.$$

So, minimizing the average distance is equivalent to minimizing the value of the above integral.

We want to find the server placement $\rho_s(x)$ that minimizes this integral under the constraint that the total number of server is D, i.e., that $\int \rho_s(x) = D$.

Resulting constraint optimization problem. Thus, we arrive at the following optimization problem:

- We know the density $\rho_u(x)$ and an integer D;
- under all possible functions $\rho_s(x)$ for which $\int \rho_s(x)\, dx = D$, we must find a function that minimizes the integral $\int (\rho_s(x))^{-1/2} \cdot \rho_u(x)\, dx$.

3 Towards Optimal Server Placement in Cloud Computing: First Approximation

Solving the constraint optimization problem. A standard way to solve a constraint optimization problem of optimizing a function $f(X)$ under the constraint $g(X) = 0$ is to use the Lagrange multiplier method, i.e., to apply unconstrained optimization to an auxiliary function $f(X) + \lambda \cdot g(X)$, where the parameter λ (called *Lagrange multiplier*) is selected in such a way so as to satisfy the constraint $g(X) = 0$.

With respect to our constraint optimization problem, this means that we need to select a density $\rho_s(x)$ that optimizes the following auxiliary expression:

$$\int (\rho_s(x))^{-1/2} \cdot \rho_u(x)\, dx + \lambda \cdot \left(\int \rho_s(x)\, dx - D \right).$$

Having an unknown function $\rho_s(x)$ means, in effect, that we have infinitely many unknown values $\rho(x)$ corresponding to different locations x. Optimum is attained when the derivative with respect to each variable is equal to 0. Differentiating the above expression with respect to each variable $\rho_s(x)$, and equating the result to 0, we get the equation

$$-\frac{1}{2} \cdot (\rho_s(x))^{-3/2} \cdot \rho_u(x) + \lambda = 0,$$

hence $\rho_s(x) = c \cdot (\rho_u(x))^{2/3}$ for some constant c.

The constant c can be determined from the constraint $\int \rho_s(x)\, dx = D$, i.e., that

$$\int c \cdot (\rho_u(x))^{2/3}\, dx = c \cdot \int (\rho_u(x))^{2/3}\, dx = D.$$

Thus,

$$c = \frac{D}{\int (\rho_u(x))^{2/3}\,dx},$$

and we arrive at the following solution.

Solution to the problem. Once we know the user density $\rho_u(x)$ and the total number of servers D that we can afford, the optimal server density $\rho_s(x)$ is equal to

$$\rho_s(x) = D \cdot \frac{(\rho_u(x))^{2/3}}{\int (\rho_u(y))^{2/3}\,dy}.$$

Discussion. In line with common sense, the optimal server density increases when the user density increases, i.e.:

- in locations where there are more users, we place more servers, and
- in locations where there are fewer users, we place fewer servers.

However, when the user density decreases, the server density decreases slower – because otherwise, if we took the server density simply proportional to the user density, the delays in areas with few users would have been huge.

Comment. From the mathematical viewpoint, this analysis is similar to the analysis of a security-related optimization problem, in which, instead of placing servers, we need to place sensors; see [5].

4 Towards A More Realistic Model

First idea. In the above first approximation, we only took into account the time that it takes to move the data to the user. This would be all if the database was not changing. In real life, databases need to be periodically updated. Updating also takes time. Thus, when we find the optimal placement of servers, we need to take into account not only expenses on moving the data to the users, but also the expenses of updating the information.

Towards a precise formulation of this idea. How do we estimate these expenses? In a small area, where the user distribution is approximately uniform, the servers are also uniformly distributed, i.e., they form a grid with distance $h = 2r$ between the two neighboring servers [4, 5]. Within a unit area, there are $\sim 1/r^2$ servers, and reaching each of them from one of its neighbors requires time proportional to the distance $\sim r$. The overall effort of updating all the servers can be obtained by multiplying the number of servers by an effort needed to update each server, and is thus proportional to $1/r^2 \cdot r \sim 1/r$. We already know that $r \sim (\rho_s(x))^{-1/2}$, thus, the cost of updating all the servers in the vicinity of a location x is proportional to $(\rho_s(x))^{1/2}$. The overall update cost

can thus be obtained by integrating this value over the whole area. Thus, we arrive at the following problem.

Resulting optimization problem:

- We know the density $\rho_u(x)$, an integer D, and a constant C that is determined by the relative frequency of updates in comparison with frequency of normal use of the database;
- under all possible functions $\rho_s(x)$ for which $\int \rho_s(x)\,dx = D$, we must find a function that minimizes the expression

$$\int (\rho_s(x))^{-1/2} \cdot \rho_u(x)\,dx + \int C \cdot (\rho_s(x))^{1/2}\,dx.$$

Solving the problem. To solve the new optimization problem, we can similarly form the Lagrange multiplier expression

$$\int (\rho_s(x))^{-1/2} \cdot \rho_u(x)\,dx + \int C \cdot (\rho_s(x))^{1/2}\,dx + \lambda \cdot \left(\int \rho_s(x)\,dx - D \right),$$

differentiate it with respect to each unknown $\rho_s(x)$, and equate the resulting derivative to 0. As a result, we get an equation

$$-\frac{1}{2} \cdot (\rho_s(x))^{-3/2} \cdot \rho_u(x) + \frac{1}{2} \cdot C \cdot (\rho_s(x))^{-1/2} + \lambda = 0.$$

This is a cubic equation in terms of $(\rho_s(x))^{-1/2}$, so while it is easy to solve numerically, there is no simple analytical expression as in the first approximation case.

The resulting solution $\rho_s(x)$ depends on the choice of the Lagrange multiplier λ, i.e., in effect, we have $\rho_s(x) = \rho_s(x, \lambda)$. The value λ can be determined from the condition that $\int \rho_s(x, \lambda)\,dx = D$.

Second idea. The second idea is that usually, a service provides a time guarantee, so we should require that no matter where a user is located, the time for this user to get the desired information from the database should not exceed a certain value. In our model, this means that a distance r from the user to the nearest server should not exceed a certain given value r_0. Since $r \sim (\rho_s(x))^{-1/2}$, this means, in turn, that the server density should not decrease below a certain threshold ρ_0.

This is an additional constraint that we impose on $\rho_s(x)$. In the first approximation model, it means that instead of the formula $\rho_s(x) = c \cdot (\rho_u(x))^{2/3}$ – which could potentially lead to server densities below ρ_0 – we should have $\rho_s(x) = \max(c \cdot (\rho_u(x))^{2/3}, \rho_0)$.

The parameter c can be determined from the constraint

$$\int \rho_s(x)\,dx = \int \max(c \cdot (\rho_u(x))^{2/3}, \rho_0)\,dx = D.$$

Since the integral is an increasing function of c, we can easily find the solution c of this equation by bisection (see, e.g., [1]).

Combining both ideas. If we take both ideas into account, then we need to consider only those roots of the above cubic equation which are larger than or equal to ρ_0; if all the roots are $< \rho_0$, we take $\rho_s(x) = \rho$.

The resulting solution $\rho_s(x)$ depends on the choice of the Lagrange multiplier λ, i.e., in effect, we have $\rho_s(x) = \rho_s(x, \lambda)$. The corresponding value λ can also be similarly determined from the equation $\int \rho_s(x, \lambda)\, dx = D$.

Acknowledgments

This work was supported in part by the National Center for Border Security and Immigration, by the National Science Foundation grants HRD-0734825 and DUE-0926721, and by Grant 1 T36 GM078000-01 from the National Institutes of Health.

References

[1] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, MIT Press, Cambridge, Massachusetts, 2009.

[2] A. Gates, V. Kreinovich, L. Longpré, P. Pinheiro da Silva, and G. R. Keller, "Towards secure cyberinfrastructure for sharing border information", In: *Proceedings of the Lineae Terrarum: International Border Conference*, El Paso, Las Cruces, and Cd. Juarez, March 27–30, 2006.

[3] G. R. Keller, T. G. Hildenbrand, R. Kucks, M. Webring, A. Briesacher, K. Rujawitz, A. M. Hittleman, D. J. Roman, D. Winester, R. Aldouri, J. Seeley, J. Rasillo, T. Torres, W. J. Hinze, A. Gates, V. Kreinovich, and L. Salayandia, "A community effort to construct a gravity database for the United States and an associated Web portal", In: A. K. Sinha (ed), *Geoinformatics: Data to Knowledge*, Geological Society of America Publ., Boulder, Colorado, 2006, pp. 21–34.

[4] R. Kershner, "The number of circles covering a set", *American Journal of Mathematics*, 1939, Vol. 61, No. 3, pp. 665–671.

[5] C. Kiekintveld and O. Lerma, "Towards optimal placement of bio-weapon detectors", *Proceedings of the 30th Annual Conference of the North American Fuzzy Information Processing Society NAFIPS'2011*, El Paso, Texas, March 18–20, 2011.

[6] L. Longpré and V. Kreinovich, "How to Efficiently Process Uncertainty within a Cyberinfrastructure without Sacrificing Privacy and Confidentiality", In: N. Nedjah, A. Abraham, and L. de Macedo Mourelle (Eds.), *Computational Intelligence in Information Assurance and Security*, Springer-Verlag, 2007, pp. 155–173.

[7] P. Pinheiro da Silva, A, Velasco, M. Ceberio, C. Servin, M. G. Averill, N. Del Rio, L. Longpré, and V. Kreinovich, "Propagation and Provenance of Probabilistic and Interval Uncertainty in Cyberinfrastructure-Related Data Processing and Data Fusion", In: R. L. Muhanna and R. L. Mullen (eds.), *Proceedings of the International Workshop on Reliable Engineering Computing REC'08*, Savannah, Georgia, February 20–22, 2008, pp. 199–234.

[8] A. K. Sinha (ed), *Geoinformatics: Data to Knowledge*, Geological Society of America Publ., Boulder, Colorado, 2006.