university of groningen

SPATIALLY DISTRIBUTED INFORMATION

From a Silly Chat to the Smart Grid

Marco Aiello

SPATIALLY DISTRIBUTED INFORMATION

Marco Aiello

Inaugural Talk (Oratie)

October 16th, 2012, 14:00

Aula of the University of Groningen

SPATIALLY DISTRIBUTED INFORMATION

From a Silly Chat to the Smart Grid

Marco Aiello

Johann Bernoulli Institute for Mathematics and Computer Science
University of Groningen

Prologue

In 1973 at the Stanford Artificial Intelligence Lab funded by John McCarthy, Mario Aiello, my father, was using a simple Unix application called *talk* to tease my mother, Luigia Carlucci Aiello, while working. The application would allow to share text live on two consoles connected to the same mainframe, a `DEC PDP11`. As my mother was typing, he could add and remove her text using a very naive form of distribution. The consoles were simply textual terminals connected via a very short network to the same machine and would share resources without much control. A basic example of what today we would call a *client-server* architecture and possibly one of the first examples of computer mediated interaction. At the time I was one year old and little would I know that many years later I would be using networked software for social interaction. Take `Skype`, for instance, it derives from the same basic idea. Having multiple users being able to interact via the exchange of messages over a computer network. So what has changed in the last forty years and where are we heading? In the next thirty minutes I will try to answer this question.

Distributed Information Systems

A first obvious answer is that we went from one extremely expensive computer connecting a couple of consoles available to the few experts to ubiquitous cheap and mobile devices for all, connected to a worldwide network. Just think that the Internet today covers about 35% of the world population with percentages close to 90% for the Netherlands and peaks of over 97% for Iceland.

Incidentally, rapid technological evolution and widespread availability are somewhat prototypical for the field. In 1965, Gordon Moore noticed that transistor counts were doubling every year and predicted that the trend would stay valid for a decade:

> *"The complexity for minimum component costs has increased at a rate of roughly a factor of two per year. Certainly over the short term this rate can be expected to continue, if not to increase. Over the longer term, the rate of increase is a bit more uncertain, although there is no reason to believe it will not remain nearly constant for at least 10 years. That means by 1975, the number of components per integrated circuit for minimum cost will be 65,000. I believe that such a large circuit can be built on a single wafer."*

This empirical observation, which is known as *Moore's Law*, has been actually valid for the past fifty years and has been shown to apply not only to transistor counts, but also to disk storage, network capacity, pixels per dollar and many other components. Perhaps less known, though quite relevant when looking at the field from the perspective of a researcher, is the *Second Moore's Law,* which states that to sustain the first law of Moore, the capital cost (including R&D) also increases exponentially over time. So the changes that computer technology have brought to our society do come at the cost of heavy investments. Something that I can't resist but mentioning today to the members of the governing bodies of our university. But let's go back to the Internet as it will help us understand more generally what a distributed information system is and what are its characteristics.

The Internet is a global network infrastructure based on open standards allowing autonomous computational hosts to coordinate by asynchronous communication. Just think of sending out emails, browsing the Web and performing a Skype call. The device you are using to do so is coordinating with at least another device somewhere on the Internet to allow to perform your tasks. Underlying infrastructure and services make sure that data from your device is delivered to the cooperating devices to achieve your global goal. A *Distributed Information System* is thus one which performs computation by means of exchanging messages among autonomous computational entities. The system has a global state representing the progress of the computation which is a function of the state of the individual components and the network.

Then there are a number of factors that define the qualities of the distributed systems, such as, the number of components that can be part of it, its *scalability*, the *resilience* it has to failures, the way in which messages are passed around in terms of *reliability, synchronicity* and *timing*. Thought the one I find most naturally defining a distributed information system is what is known as *transparency*. Transparency is the abstracting away from resources and distribution services for the components participating in the system. Something that allows you to make a Skype call without knowing where the responding party is in the world and how he's connected to the Internet or the telephone lines. Something that allows you to use an email address to send a message ignoring any other detail of your recipient's way of consuming your content. Something that allows us to build simple and light programs that are very powerful. Something that allows people to perceive software as a light entity, as Italo Calvino did when referring to computers in his 1985 lecture on Lightness (Leggerezza) to be presented at Harvard. Such transparency of a distributed system is so essential that it has been even used as the definition itself of a distributed system. Leslie Lamport, who has greatly contributed to the field, in an email sent on 28 May 87 while at DEC writes:

> *"A distributed system is one in which the failure of a computer you didn't even know existed can render your own computer unusable."*

Transparency though comes at a cost. To make it simple for applications to be able to "move" on networks, to connect to resources they find on the fly when they need them, to send messages reliably, we need to take great care in designing our systems. The *middleware* is the software layer that provides an uniform view for software applications being part of distributed information system. It is the key component to implement the transparency and allow for information to be distributed. The more transparent we want our systems, the better equipped has to be the middleware, because, among other, one of the characteristics of a distributed system is that components may fail. I'm sure you have all experienced an application quitting unexpectedly, a phone call dropping suddenly, an email not being sent out even though you can swear you've hit that send button. Now if one machine can fail, then take many machines put them on the same network and consider what is the chance of any of them failing at any given moment. So what a well designed distributed information system has to be able to do is to anticipate possible failures, be able to detect them and recover from them. Simple failures are called *crashes*, that is, a machine stops responding and this is easy to detect, but some more nasty cases exist, for instance when a machine is still executing and responding, but it is maliciously producing messages that are not correct. Such case is called a *byzantine failure*. The term comes from a 1980 paper of Lamport and others where they talk of a Byzantine army needing to coordinate an attack on an enemy city. The army is divided into troops which station at different locations, each led by a Byzantine general. The generals can only coordinate by sending a messenger on horse to one another with a message. If the troops attack together, they will defeat the city resistance, on the other hand, if they attack uncoordinated they will be defeated. A byzantine general can be a traitor and will send out confusing messages to break the coordination and allow the city to survive the attack. From a distributed system point of view, the interesting question becomes: under which conditions one can tolerate having byzantine failures? and how many of these can one have as a proportion of the whole army while still being able to function correctly? It turns out that under the condition that the messengers running from one general to another actually make it within a given time, the system can still function if there are at most one less than a third generals being malicious traitors. Byzantine generals are an example of how far research has gone in defining the formal properties of a distributed system. In making sure that when we design a system, we know what we can expect from it, we know what faults we will be able to catch and recover from, we know how to trade off between having consistent data across the whole system and making sure that updates do not take too long.

Today Computation is Distributed

If we agree that distributed systems require formal methodologies to be designed and artful engineering skills to work properly, and if we also agree that distributed systems are among the tools that we work daily with, then I am ready to formulate my first of three claims.

> ***Proposition I:*** *Today it is hard to find a computing system that is not distributed.*

Today your PC assumes that it can go on-line regularly to get software updates, your watch can take signals from satellites to be more precise. Even your washing machine or drier can be made capable of communicating with your home automation system, with a remote billing system, or with a maintenance system. On February 17, 2012 Apple has reached the target of selling 25 billion apps on its store. These are applications that come from a network and run on a networked device. Most of them, if not all, rely on networking for updates, storing user data, pushing notifications, allowing for on-line gaming or using remote databases and services. Cars use GPS, radio and GSM networks to position themselves, give traffic information and provide servicing requests. Cisco estimates that the number of devices connected to IP networks will be twice as high as the global population in 2015. There will be two networked devices per capita in 2015. Driven IP traffic will reach 11 gigabytes per capita in 2015, up from 3 gigabytes per capita in 2010.

It is thus a fact that most meaningful and sizable systems have one form of distribution or another. But why is this so? The fact that networks are widely and cheaply available is just a prerequisite for this, but not the justification. The fact is that building software that relies on components remotely available allows to build systems that are modular, scalable, efficient and robust. A typical example of this type of construction is based on the *three-tier architecture* pattern. A data-tier is concerned with the permanent storage of information, say a database server maintaining information about student marks. A logical-tier has access to the data servers, so that it can perform logical operations which are useful for the application, for instance, confirming that a particular student has obtained enough credits to be eligible for a diploma. Finally, the presentation-tier is concerned with providing access to the system for the end-user. Typically these can be Web interfaces or mobile device Apps.

The art of building good distributed systems lies in making wise decisions on where to store the data, how many copies of it are kept around in the system and what is their level of consistency. Take for instance FaceBook, who currently enjoys about one billion active users worldwide. It is estimated that it has more than 100.000 servers to manage its data base and that individual items are replicated tens of times, that is, the text that you posted last week is copied to at least ten servers. Now, what is a good policy to maintain such data consistent as more posts add up on your profile? The choice of making sure that all posts are represented following the real time of their writing is hard to impossible to implement without rendering the system slow and unusable.

Going back to the first proposition, I stress that it doesn't stand to say that monolithic single machine systems don't exist, that there is no electronic device without an antenna or a network interface, they still exist in the form of desk calculators, quartz watches, and I am not sure what else. But these have a role in our everyday life that is as relevant as the role of a greenhouse tomato in high cuisine.

Being Served

After the completion of my PhD on the logical foundations of spatial representation, I've been intrigued by the problem of how to build distributed systems that are scalable, modular and able to autonomically adapt to their execution environments. The first challenge was presented to me in 2002. Say that you want to organize a trip which involves traveling, relocation, booking artistic events, paying for the services you reserve, having insurance, and so on. Say also that you want to be free to choose from any available service provider. How can one then support a user with a tool to create an optimal travel package? The idea is that every travel related service has a software presence on a network and offers its functionalities by allowing the exchange of standardize messages. Then the task is to interact with these services, obtain relevant data from them, and try to find a package satisfying the user preferences and compatible with the available offers. As simple as the problem may seem, there are actually many challenges to be addressed. One is that the system has no control over its environment: service may come and leave, the pricing and availability of service offers are not known a priori and may also change while creating the travel package. Another challenge is to allow the user to leave out details of the trip or state preferences. Finally, when some decisions are made it is not always possible to retract them, for instance, if a cheap flight has been booked, it may cost money to cancel the reservation.

The distributed information systems for which the components are black boxes available through standardized interfaces on a network and for which there are facilities to search for them are known as *Service-Oriented Architectures*. These follow the simple pattern of `publish-find-bind.' That is, when a software component needs a remote functionality, it finds it and then binds it to itself. Service providing components have to make themselves known by publishing their presence and their interfaces. The extreme power of these architectures is to allow the process of connecting components to happen as late as at execution time.

After having considered the travel domain, when I arrived in Groningen, I was presented with new challenges and got engaged in various research projects ranging from customizable business processes for Dutch municipalities, to infrastructures to make homes smart, to making buildings energy aware and saving. Here let me just briefly touch on how one can make homes smarter by taking advantage of service-oriented architectures. The main idea is to look at homes as something more than just walls and windows, but as a collection of autonomous physical components that have a presence on a network and are ready to interoperate with one another. The fridge can talk to the iPad to inform of available ingredients, a gas alarm system can ask a window to open for mitigating the effects of a gas leakage, and so on. Now, if we accept that devices are networked and organized as a service-oriented architecture, how do we make the home smart? My approach has been to add an intelligent component to the home that is able to query for the state of all other devices and based on this decide how to act on behalf of the user. The advantage for the user is that he now needs simply to express his wishes to the home, and doesn't have to operate all the single elements. Say that you want to organize a party. You just inform the home of this desire, and the home will check whether you are stocked on groceries and drinks, it will turn on the lights in the living room to the desired configuration, it will start the right music, close the bathroom door, if open, and so on. And to make the sensation of your wishes come true even stronger, why not allowing the user to simply think about the wish rather than having to input it to the home in some other way? If reading the mind is not yet possible, what is possible is to use non invasive readings of brain activity to drive an interface with a few dozen predefined wishes. Besides being rather appealing to the general public, such a solution is also very useful for people who have major physical impediments in order to render them autonomous and in control.

One way I like to look at building distributed systems, and perhaps information systems in general, is as solutions to support users in achieving their goals with lesser or no effort and in many cases to achieve goals that would not otherwise be reachable. In 2007, invited to deliver a keynote at an international event in Vienna, I formulated the following claim, which I repeat here as my second proposition:

Proposition II: *Distributed information systems foster user laziness.*

I do not mean that designing and engineering distributed systems is an activity that can be carried out being lazy, on the contrary. Though, I do mean that systems built in this way can support the user by eliminating unnecessary effort for them. It can also eliminate wasteful use of other resources, such as energy.

The Next Big Thing: The Smart Grid

A great way to being wrong is to make predictions. I will now take this risk while formulating my third and last claim. To do so, I resort to a personal recollection.

In 1995, I had just returned from a summer internship at Apple in Cupertino and I had set up my first Web server using Webstar and programming a novel set of Common Gateway Interfaces. I had a great business idea. Set up a server and sell Web presence to small and medium enterprises in Rome. Since I had just mastered the technology, all I needed was a fixed internet connection, a small server and to start the marketing. A fellow student and I took our best suit, made an appointment with a salesperson of Telecom Italia, a monopolist at the time, and we went to ask for prices for fixed connectivity. We didn't need much bandwidth as back then a web page was mostly composed of text and low resolution images, but we needed to be sure to have uptime close to 100%. The salesperson offered his basic solution to begin with, an ISDN connection of 64Kb per second---something close to the bandwidth that you get today as a punishment from your mobile provider when you have used up all your monthly quota---for the spectacular price of 100 million Italian lira per year, about 70.000 euros of today. In an instant he sank our great business idea. Without making a beep, we thanked him and departed. Though, we were left with a bitter after taste. And that feeling corroborated when a few years later I could get a much faster cable connection for my home for less than 50 euros a month and leave my computer with a file and web server running 24/7. What had happened to the telecom industry in the meantime? A radical change. The telecom sector, which was over 100 years old, was considered a natural monopoly. The telecom providers, mostly government owned, managed the service, network and equipment layers. All R&D was done internally and any innovation sprouted by so-called non-market drivers. Fransman identifies as causes of this paradigm shift the following ones: cross-country competition, political pressure and consumer pressure. For instance, in 1985 the European Community emanates the 'Liberalization Directives' under Article 90 of the Treaty of Rome determining the deregulation of telecommunications market for the next decade. Furthermore, Fransman notes

> *"by the end of 1995, [...] the now incumbent network operators [were] making the decision to leave more and more of the R&D related to the network and its elements to the specialist equipment suppliers. At the same time the incumbents decided to open their procurement, agreeing to buy from new suppliers in addition to their traditional suppliers."*

In other words, the suppliers were becoming innovators and ready to play on the telecom market in more than just one role. New players were appearing from unexpected areas. For instance, an electronics company like Olivetti together with Mannesmann in 1995 enters the Italian market creating Infostrada, the first competitor of Telecom Italia. In the same period the Internet and the Web were emerging as infrastructures for all and people started requesting access to them. This was in contrast with the general worry of the telecoms of having enough bandwidth. I remember absurd debates were technological experts were claiming that we could not provide enough bandwidth for all. Specialized magazines featured scary covers referring to the *"last mile problem"* of the Internet, claiming that it was easy to transport large amount of data on communication backbones, but then reaching every house with enough bandwidth would have been unfeasible. Interestingly, while I am writing this at home, I am watching a NBA game in HD streamed directly to my television via my iPad while my wife is making a Skype telephone call to her mother in Göttingen.

After the telecom sector, let me consider another critical infrastructure, the power grid. In the second half of the XIX century, Samuel Insull, the trusted right arm and successor of Thomas Edison, referred to the energy sector as a *"natural monopoly,"* something that is to a large extent true up to today. He continued

> *"every home, every factory and every transportation line will obtain its energy from one common source, for the simple reason that that will be the cheapest way to produce and distribute it."*

Such monopolies are organized in layers going from the power generation, to retailing via transmission and distribution. R&D is mostly performed internally and hardware equipment can be procured externally. Governments are indicating the necessity of unbundling and directives such as the EU 2009/72/EC indicate the way to achieve it. While retail prices are raising, user are embracing the possibilities of micro-generation by considering the installation of solar panels or micro-wind mills at their homes. The attractiveness resides in lowering the cost of the energy bill and potentially being able to make money from the over production. Though today there is not a free market for such produced energy. It must be sold back to the grid operator at a fixed price. In addition to the economic, there is also a social incentive. Most people do want to be "green" and make choices with a label of being sustainable. So one can expect consumer pressure for open markets to increase in the future. Does all this sound familiar?

My claim is that the power grid sector is today in a very similar situation to what the telecom sector was in the '90s and that we are ready for a major change in the way energy is produced and distributed.

Proposition III: *The next technologic revolution will be about energy.*

We will face a *"last mile problem"* again, but instead of having to do with Internet bandwidth, it will have to do with energy transportation at the low voltage level of households. As we include renewables at all scales in our power grids and as consumers demand freedom to supply and to sell, low voltage capillary interconnectivity will be necessary so that neighbors can engage in energy trading and implicitly transform neighborhoods into energy neutral areas. Some recent studies of ours suggest that the new structure of such grid will be similar to that of social networks, that is, take the form of *small world graph*. But why am I talking about the energy grid while reasoning about distributed systems?

The transition that the power grid is facing towards being an open energy infrastructure, in much the same way the Internet is an open network, can only be accomplished with coupling it with a strong information and telecommunication technology infrastructures. If electrons start traveling in a peer to peer fashion on the power grid, information bits have to accompany them in order to allow accounting of the energy flows. And if that was not enough, trust and security will become a greater concern that can only be addressed by having secure information systems control the power grid.

A term gaining popularity concerning the trend towards informatization of the electricity system is *"Smart Grid."* In the current initial phases of this trend, there is not much smartness in the grid. We can say that we have more information available, that we have digital meters, that we have digital control over portions of the grid. But the potential of having such information available and building applications on top of it is all to be explored and is enormous. Large companies can predict loads and control precisely the generation and distribution of energy. Renewable sources---which are typically irregular in their generation patterns---can be more easily included in the grid. Finally, the end users can access an open infrastructure where they can trade energy freely with any other party.

Having had the privilege to enjoy the Internet revolution during my lifetime, I hope I will be able to witness a second one: the imminent energy revolution.

---*I have spoken.*

Appendix. Spatial Distribution in The Netherlands

Having argued on the relevance of distributed information systems in our society and in computer science, I would also like to reflect on why would an individual decide to perform such research in The Netherlands.

The first natural answer is that it doesn't matter where the research takes place. If information is distributed and pervasively accessible, creating infrastructure for managing that information should be transparent to the location too. Such an answer, in as much as it is true, is though missing on some important historical and cultural aspects. The Netherlands is among the first countries initiating the debate over the qualities of the more modern peer-to-peer models versus the old fashioned centralized client-server ones. Today we accept as a fact that peer-to-peer architectures of distributed computation are by far more efficient in addressing issues of scalability, availability and resilience to a large variety of faults and attacks, but dogmatically for years we considered the centralized client-server model of computation as the only possible one. As I understand it, the debate started as early as in the XVI century with a very specific topic, which concerned the computation of actions to remove sin. Traditionally, such computation could only be done in a multiple client single server way. The centralized component would receive a request for indulgences and related payment, and then sins would be forgiven: "*As soon as the coin in the coffer rings, the soul from purgatory springs.*" The Netherlands was at the forefront of the movement that considered centralized payments related to sin-information to be a wrong model, while favoring decentralized ones.

If such foresight of the country was not enough, I must admit that I was also highly motivated to pursue research and live in the Netherlands after reading about the country in various classic texts. The first description I came across was the succinct depiction present in Voltaire's Candide, 1759. I was intrigued to meet the descendants of the people for which Voltaire would write:

"Good heavens, to what excess does religious zeal transport womankind!"

One of Schiller's first writings, The History of the Revolt of the Netherlands, 1788, has also contributed largely to foster my curiosity. He described a country

"where with the good cause unwonted powers were united, and the resources of resolute despair triumphed in unequal contest over the terrible arts of tyranny. [...] Not so, however, in the history before us. The people here presented to our notice were the most peaceful in our quarter of the globe, and less capable than their neighbors of that heroic spirit which stamps a lofty character even on the most insignificant actions. The pressure of circumstances with its peculiar influence surprised them and forced a transitory greatness upon them, which they never could have possessed and perhaps will never possess again. It is, indeed, exactly this want of heroic grandeur which renders this event peculiarly instructive; and while others aim at showing the superiority of genius over chance, I shall here paint a scene where necessity creates genius and accident makes heroes."

Finally, Domenico Alberto Azuni in the Maritime Law of Europe Volume I, 1805, provides an account of the country. Interestingly the text, which was commissioned by Napoleon as the foundation for maritime regulation, is becoming a reference today for looking at the Internet and how to regulate it.

"If, at the present day, a perfect democracy were possible, the sea alone could become the theatre of its existence. Every nation has an equal right to launch its fleets on that element; there, every man has a right to navigate, to transport the productions of his soil or the fruits of his industry, and to plough the surface of the deep, from pole to pole. Absolute democracy, or, to speak more correctly, a perfect equality of rights on the sea, is the natural state of maritime nations."

While referring to the Netherlands, the Sardinian jurist wrote:

"Holland is nothing more than a bank of band and mud, raised by accident, and which accident may, at any time, destroy. The situation of its sunken land, below the level of the sea, has obliged the inhabitants to oppose the violence of the waves, by dikes, to dig canals in every direction, large enough to facilitate internal communication, and draw off the stagnant waters which covered its surface. Familiar with water, the Hollanders have been compelled to seek their chief subsistence from the sea. Necessity, which first made them fishermen, soon taught them how to cure and preserve their fish, to make them an object of commerce, procure in exchange from neighbouring countries, what they wanted in their own, and thus augment the pleasures of life. The riches which this industry brought into the state, enabled them to pursue other branches of trade that opened new sources of wealth. The strength of Holland increased with her opulence, she dared to take arms to obtain her freedom, and she succeeded in the attempt. Commercial and industrious, she knew how to create, at the same time, territory and liberty, commerce and colonies, a useful navigation and a formidable marine. Emerging by the force of genius from the bosom of the ocean, she seemed to have more right than any other power to rule over its waves."

List of Names

- **John McCarthy** (Boston 1927 – Stanford 2011) founding father for Artificial Intelligence
- **Mario Aiello** (Sassari 1945 – Pisa 1976) my father, a computer scientist when the field was still for pioneers
- **Luigia Carlucci Aiello** (Cerreto d'Esi 1946 –) my mother, professor of Artificial Intelligence at La Sapienza University of Rome
- **Gordon Moore** (San Francisco 1929 –) co-founder and Chairman Emeritus of Intel
- **Italo Calvino** (Cuba 1923 – Siena 1985) journalist and writer
- **Leslie Lamport** (New York 1941 –) computer scientist, currently at Microsoft Research
- **Samuel Insull** (London 1859 – Paris 1938) public utilities magnate
- **Thomas Edison** (Milan, Ohio 1847 – West Orange 1931) inventor and businessman
- **François-Marie Arouet** a.k.a. **Voltaire** (Paris 1694 – Paris 1778) philosopher and historian
- **Johann Christoph Friedrich von Schiller** (Württemberg 1759 – Wiemar 1805) poet, philosopher and historian
- **Domenico Alberto Azuni** (Sassari 1749 – Cagliari 1827) jurist and judge

Acknowledgments

I could not have come thus far in my research and academic career without the influence, help, support and sympathy of many people. Besides the ones already mentioned in the text of the inaugural speech, and running the risk of forgetting someone, I must explicitly mention the following individuals: my PhD advisors Johan van Benthem and Arnold Smeulders; my senior colleagues: Nicolai Petkov, Schahram Dustdar, Mike Papazoglou, and Jim Spohrer; my colleagues and PhD students: Alexander Lazovik, Chang-ai Sun, Pavel Bulanov, Viktoriya Degeler, Ando Emerencia, Anna Frankova, Ilche Georgievski, Heerko Groefsema, Eirini Kaldeli, Faris Nizamic, Tuan Anh Nguyen, and Andrea Pagani; and the logistic support of Esmee Elshof. Finally, I owe my serenity and well being to Heike, Maurizio and Aurelia.

Contact aiellom@ieee.org

www.distributedsystems.nl