International Journal of Computer and Communication Technology

Volume 2 | Issue 1

Article 2

January 2011

“Next Gen Smart and Connected Systems”

Srinath Sriram

Associate Vice President HCL Technologies Ltd, srinathsriram@gmail.com

Follow this and additional works at: <https://www.interscience.in/ijcct>

Recommended Citation

Sriram, Srinath (2011) “Next Gen Smart and Connected Systems”, *International Journal of Computer and Communication Technology*: Vol. 2 : Iss. 1 , Article 2.

DOI: 10.47893/IJCCT.2011.1062

Available at: <https://www.interscience.in/ijcct/vol2/iss1/2>

This Article is brought to you for free and open access by the Interscience Journals at Interscience Research Network. It has been accepted for inclusion in International Journal of Computer and Communication Technology by an authorized editor of Interscience Research Network. For more information, please contact sritampatnaik@gmail.com.

“Next Gen Smart and Connected Systems”

Srinath Sriram
Associate Vice President
HCL Technologies Ltd

Abstract—With the kind of technological advancements such as cloud computing, virtualization, computing grids, etc, that we are seeing today, soon the hardware and software producers will be under severe pressure to deliver cheaper and cost effective solutions and to combine them with innovate commercial models such as “pay as you use” or “pay per use” etc.

Imagine that your lap top contains only a browser like software, which will enable you to connect to clouds around your area and then dynamically fetch the required computing resources of h/w and s/w. It will be even better; if you could get software (even a DBMS) that you want at that point in time to execute an application, for a pepper corn price subscription.

In order to be made affordable of such massive computing hardware and software systems, the vendors have to depend on exponential volume growth of users and also cost reduction of the computing infrastructure by taking a complete re-look at the architecture of the h/w and s/w systems to make them lighter and cheaper to manufacture.

The industrial revolution taught us job specialization as one of the major means to improve productivity and to cut manual labor costs. We now need such a specialization in the computing arena as well.

I am starting with a fundamental premise that all computing systems available today are heavily biased towards data processing with very limited specialization. The systems are produced to contain an overload of computer parts of h/w and s/w code, more than what is required for most situations. Therefore, whether you like it or not, you end up paying for the overloaded complexity that you may very rarely require.

The main theme of this submission is to relook at creating optimally functioning specialized systems coupled with the elimination of massive overload of unwanted processing needs could make them less complex, easy to use and cheap.

I. Introduction

In order for a system to be called smart and connected, we want the system to be very specialized in certain things that it can do best and connect to other systems that can perform the functions that it can't do by itself.

We no longer have to depend on owning a single large and powerful system at hand for ourselves and instead use several small systems together on demand in a coordinated way, to complete the job even faster than before.

Most companies currently invest in massive computing infrastructure and do not use it to full extent, as they do not hit the full peak load anticipated every time they use. They hit it once in a while, may be few days in a year and yet have to invest and maintain the infrastructure, which remains largely unused for most days of their operations. Therefore the capital expenditure spent up front lies unproductive.

In the following sections, we will look at designing some of the components that make up these smart and connected systems.

II. Solution Components

Let us assume that we are in the year “2020” and that we have the “Next Gen Metro” called The metro relies on multiple water resources from where the water is collected, treated and then supplied to millions of households 24/7. You not only get water supply any time you open a tap but you are also guaranteed a “safe for drinking purposes” as well, by having the chemicals in the water precisely controlled to the agreed and published safety levels.

Let us also assume that the entire water resources management, treatment and supply lines are managed by our smart and connected systems.

Such a system would have been built on an architecture envisaged below.

Smart and connected system’s architecture has several systems or subsystems:-

1. Data acquisition system
2. Data presentation system
3. Data collation, ordering and persistence system
4. Decision making or data processing system
5. Process or work flow management system
6. Infrastructure management system

a. Data Acquisition System

In order to do water quality management at the water sources and at the outlets from the treatment plant and on the supply pipes to the households in our example, we need to have “smart water sampling systems”. These can analyze the water sample, send data about the

“Nemo” in India. Don’t you see the similarity with the name “Nano”? Yes, it was intentional!

chemical contents to other systems that can be connected on demand.

As far as the specialization of this water sampling data acquisition system goes, all it needs to have is the ability to collect the right volume of water sample, do analysis, derive the required data, connect to configured systems, send the data and wait for acknowledgement.

It does not need a massive OS, CPU with large instruction set that does all possible computations from a simple addition of 2 numbers to sending a Mars Lander to Mars.

Additionally, it can also send the data to the presentation devices (to mobiles of the monitoring agents or a control room CCTV or even to a dashboard installed at the site or to even your household laptops). So what to connect and what data to be sent could be configurable and no need for great deal of programming or coding once you buy a smart system off the shelf for water sampling and analysis.

You can count by hand; the high level instruction set that the combination of hardware and software that the system needs to perform.

1. Pick water / Pour water
2. Analyze water
3. Derive data
4. Store/Retrieve data in temp storage within itself
5. Connect (to) Smart / Disconnect (should be able to connect to any device or any scheme form internet to household devices to mobiles or to other smart systems etc)

6. Send/Receive/Acknowledge data
7. Configure connect parameters / systems / data to be sent or received and the workflow or process required to be accomplished.
- 8.

Obviously, commercial data processing applications may not rely so much on such special data acquisition systems and will require screen based data presentation and capture which is dealt in the next system.

b. Presentation System

This specialization revolves around the ability to present data in the configured device given a layout and the data. The layout is configurable and stored in the device itself or received as part of the request for presenting something. For example, your mobile phone is to be sent an alert that the “water supplied to your pipe is contaminated or not within permissible chemical levels” and that the “supply is cut-off for the next 3 minutes until a good quality water is restored”.

The data could be even complicated with many different layouts with its own pages of data (the most I have seen is in a property re-insurance, a screen collecting up to 255 fields in a horizontal scroll and had 15 different screens full of data – about the risk being re-insured).

The presentation smart system will be able to interpret the layouts and the data and render it in a rich presentation.

Obviously, in a typical data processing application, presentation system will also have the ability to collect data from the users (new

data entered, edits etc) in a predefined format just like the smart data acquisition system. So the choice here is to have data acquisition and data presentation abilities merged in to a single smart system or have 2 individual components attached in a system and working in tandem. That is, taking specialization to the extreme, so to say.

c. Data collation, ordering and persistence system

This smart system will implement specially designed data related functions to receive large volumes of data from millions of data acquisition systems. Some of the data related functions are sorting, collating, storing and retrieving in a data management system of your choice. It is another matter that we have to even redesign and produce a new data base management system that are specialized for storing and retrieving purposes only – rather than doing everything such as application development, network and near OS level functions that are duplicated with other paraphernalia in the present day data base management systems. It should be able to summarize, cube, dimension, calculate, total etc at lightning speed on a very large volume of data (remember that we have millions of data acquisition systems collecting water samples at vantage points all over the metro) mostly in memory and if required to be persisted.

The logic for performing the data functions are application specific and will be supplied by the application provider or even the end user at run time dynamically and supplied along with the data.

The data systems must have also the ability to interpret the commands for performing data functions and a standardized instruction set that are limited to data functions such as Data

definition, Derive/Send/Receive/Acknowledge data, Store/retrieve, Data functions (such as sort, sum, average, summarize, pivot, calc on dimension etc) and Data configuration.

d. Decision making or data processing system

This specialized system deals with the actual work that will be performed in any application. It contains the business logic once you have the data required is delivered or guaranteed to be delivered when the data access logic is performed. This will retain most part of the syntax (instructions) and interpretation of it, minus the paraphernalia. It could be as simple as a basic language mainly dealing with fetching the application logic, interpretation, receiving/sending and setting the required data, instruction execution control functions (such looping, jumping etc, conditional, co-execution etc) and conditional or comparative operators.

Most modern day applications are fat and complex in this area due to the vagaries of the development language, development environment, deployment environment etc, rather than the application logic itself. Where the application logic itself is very complex, the total complexity gets even more compounded, leaving the hapless users even more confounded.

e. Process or Work flow management system

This smart system simply deals with the ability to configure a sequence of steps that the user wishes to perform in order to construct a transaction required. An example of the workflow steps or sequence in our water sampling process will be to first collect the samples,

collate/sort/calculate, present data about analysis results, decision making (for stopping the water from flowing to the treatment plant itself or from treatment plant to the specific contaminated pipes).

Each step in the constructed transaction can be configuration enabled to define who are the preferred application logic providers for that step, where or how to connect to such smart systems and what resources to be acquired to perform that step.

f. Infrastructure Management System

Bulk of the dynamic fetching and releasing of the computing resources and application logic will be managed by this specialized system. It will also deal with capacity management, load balancing and parallel execution or setting up a grid for parallel execution etc. The key piece of the solution for the whole puzzle lies in here where it would understand the computing resource needs of the connected systems and the application logic to be served by them and making the reserved/available infrastructure for the logic to be executed in a cheap and faster way.

III. Conclusions

In the submission above, I have tried to make the point that the present day systems are built towards large generic data processing solution and are built to perform many functions that are redundant or irrelevant to many computing situations. In order to make it available in a cost effective way to the end users and on a utility based ownership model (where one pays for the use of electricity but not own or pay for major pieces of the supply systems), it is imperative that most operating systems, database systems,

office productivity suite, application systems and many more software and hardware systems are created with less complexity, size, volume and cost. Most vendors will end up redesigning lighter and specialized systems to order to cut cost and complexity. One such attempt was this article, wherein, the argument was about redesigning and if redesigned what could be the high level architecture.

It will be surprising to know that only two decades ago companies like HCL and Wipro built h/w, operating systems, language compilers, database systems and applications, all indigenously without much of exposure to western design and products to overcome the import duty and restrictions regime. But somewhere down the line, when we started tasting the success of exporting IT manpower for easy and quick money rather than the toil associated with the original research and innovation, we have silently sealed off R&D units and projects that could have by now produced such next gen ideas, concepts or even products.

It is also my desire that the enormously successful Indian IT companies refocus their investments to take up the original research, design and development again to produce such next generation smart and connected systems not necessarily using the same architecture presented here.