Pros and Cons of Running Comsol Touch-Sensor Simulations on Amazon Web Services

Author: A. Gourevitch,
198 Champion Ct. San Jose CA 95134, alxg@cypress.com

Abstract: We report an implementation of parallel computing on Amazon Web Services (AWS) for touch-sensor modeling. Comsol Multiphysics was used to simulate an electromagnetic field distribution in a capacitive sensor assembly. Multiple Comsol jobs were deployed on separate AWS instances and were executed in parallel. The simulation results indicate that implementation of parallel computing for Comsol simulations can significantly reduce the computational time required for optimization of capacitive touch sensor patterns.

Keywords: Touch-sensors, Amazon Web Services.

1. Introduction

Capacitive sensing as a human-device interface is becoming increasingly popular in many consumer products such as laptop track pads, touch-screen tablets, computer monitors and smart phones, but it is certainly not limited to these applications. A human finger motion is detected by a capacitive touch-screen. Touch-controllers interface touch-screens analog and digital device circuitry and process the detected signals using firmware algorithms. Cypress Semiconductor TrueTouch microcontrollers can detect the motion of multiple fingers in the presence of up to 40Vpp AC noise [1-3].

Touch-screens consist of one or two transparent conductive layers patterned to produce horizontally and vertically connected electrodes that form a grid structure. A touch object is sensed at the grid intersection.

The touch-screen patterns can be laid out in indium tin oxide (ITO), silver nanowire or metal mesh layers deposited or laminated on glass or polyethylene terephthalate (PET) substrates. These layers are embedded in the sensor stack-up. A capacitance measurement circuit is coupled to the sensors to read the output signal. Generally the touch-screen controller is mounted on a flexible printed circuit (FPC) that contains all necessary external components and interconnecting circuitry for the touch-screen system.

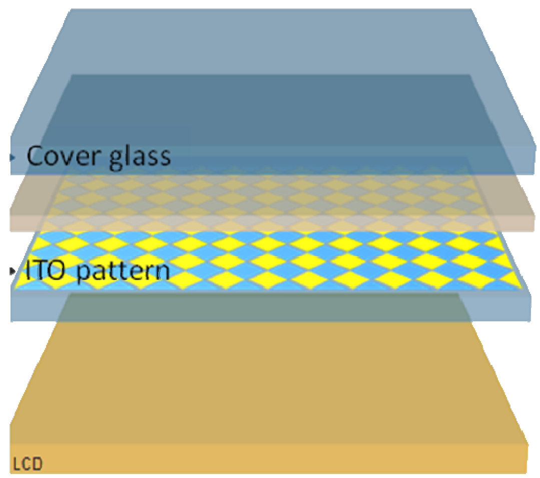

Figure 1. Single-layer touch-sensor stack-up

A typical touch-sensor stack-up is shown on Figure 1. The patterned ITO layer is sandwiched between a liquid crystal display (LCD) and a cover glass. Yellow and blue square electrodes connect together producing horizontal and vertical routes. When a human finger is touching the cover glass surface it distorts the electromagnetic field in the sensor assembly. This touch is measurable as a change in a mutual capacitance between horizontal and vertical electrodes [4].

2. Use of COMSOL Multiphysics for optimization of touch-sensor design

Comsol Multiphysics was used to simulate an electrostatic field distribution inside of the touch-sensor assembly. The 3D touch-sensor model was built using Comsol internal geometry editor. The model includes sensor electrodes and stack-up layers. The Comsol AC/DC module was used to simulate electromagnetic field distribution in the 3D assembly and estimate the coupling capacitance between the selected electrodes, see

Figure 2. The Comsol model was parameterized to enable sensor optimization without editing the model geometry.

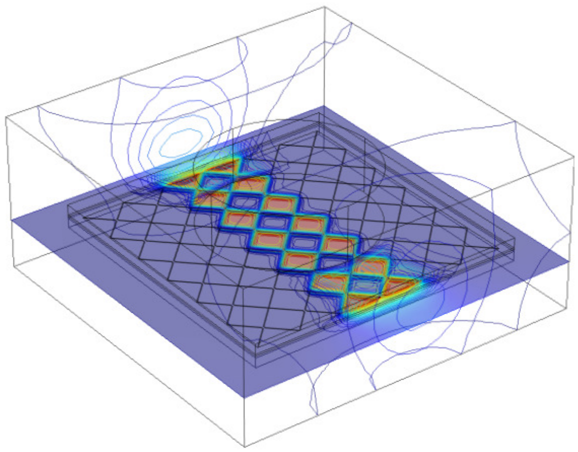

Figure 2. 3-D Comsol touch-sensor model

Sensor design optimization requires running numerous independent simulations with different pattern element shapes and dimensions. These simulations can be considered as a sequence of jobs independently varying one or more model parameters. The jobs can be sequentially executed on a single computing node or can be deployed as an array of jobs that share the same executable and resource requirements, but have different input parameters, see Figure 3. By running the jobs in parallel, the Comsol tool generates an accumulated simulation data set in one shot without waiting for a sequential job execution. This computational procedure was realized on AWS cloud.

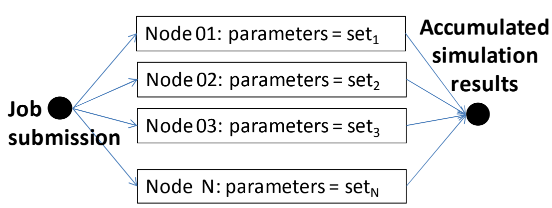

Figure 3. Running EM simulations as parallel jobs

3. Comsol simulations on AWS

The Comsol computational jobs were deployed on AWS and performance improvement was benchmarked with a premise DELL T5500 desktop.

AWS provides a collection of remote computing services that together make up a cloud computing platform [5]. The main advantage of AWS is vertically (instance type) and horizontally (cluster) computational resource scalability. Comsol simulations were tested on a single AWS instance and on a cluster. Table 1 summarizes resources of cc2.8xlarge and cr1.8xlarge instances used in this work. The cluster solution architecture is shown in Figure 4.

Instance type	vCPU	RAM (GiB)	Network
cc2.8xlarge	32	60.5	10 Gb
cr1.8xlarge	32	244	10 Gb

Table 1: Resources of the AWS instances

CC2 and CR1 Instances are backed by 2 x Intel Xeon E5-2670 processors, eight-cores with hyper threading. Instances launched into the same cluster placement group are placed into a non-blocking 10 Gigabit Ethernet network.

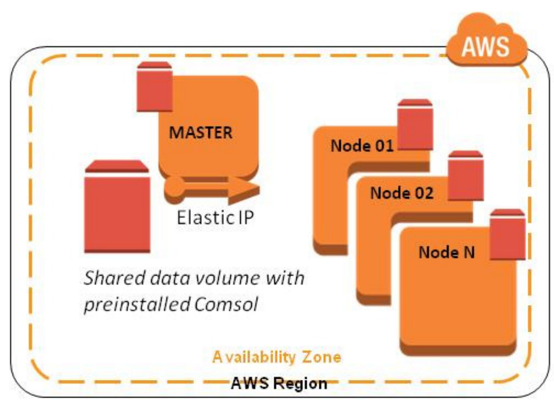

Figure 4. AWS cluster architecture

Several touch-sensor models were run on AWS instances and on a DELL T5500 desktop. The T5500 machine has 48Gb RAM, Intel® Xeon® CPU X5680 @3.33 GHz (2 processors) total 12 CPU, 64-bit operating system. Significant improvement was observed for a large simulation project with more than 30 million mesh nodes. We were not able to complete this job on the T5500 machine within 2 hours of run time (only several percent of the job was completed) vs 30 minutes required to fully complete the job on AWS cr1.8xlarge instance.

We also tested single job performance running on distributed resources. The Comsol touch-sensor model was deployed on a 3-node cluster and on a single instance and simulation time was compared. The cc2.8xlarge instance was used in the benchmarking experiment. We observed almost 3X simulation time increase for the distributed job vs stand alone one. The simulation time increases due to intercommunications between the shared nodes.

4. Conclusions

We report an implementation of Comsol touch-sensor simulations on AWS. Multiple Comsol jobs were deployed on separate AWS instances and were executed in parallel. The simulation results indicate that implementation of the parallel computing for Comsol simulations can significantly reduce computational time required for optimization of capacitive touch sensor patterns.

5. References

1. http://www.cypress.com/
2. M. Ribeiro and J. Carey, 5th Generation Touchscreen Controller for Mobile Phones and Tablets, Hot Chips Symposium 2013
3. J. Carey, EE Times, Noise wars: Projected capacitance strikes back against internal noise (2011)
4. L.K. Baxter, *Capacitive Sensors: Design and Applications,* John Wiley and Sons (1996).
5. http://aws.amazon.com/

6. Acknowledgements

We would like to acknowledge AWS HPC team and especially Dougal Ballantyne, a Solutions Architect with Amazon Web Services LLC assisted with the cloud formation architecture.