**Parallel and Continuous Join Processing for Data Stream**

Hello everyone, my name is Song Ge, I am a Ph.D. candidate from MICS library of CentraleSupelec University of Paris-Saclay. Thank you all for coming to my Ph.D. oral defense. I hope that you will enjoy my presentation.

My dissertation title is Parallel and Continuous Join Processing for Data Stream. And this is a large topic, so I decided to divide it into two parts. The first part is Data Driven Stream Join which we take the algorithm of finding the nearest neighbors as our use case. And the second part is Query Driven Stream Join, for this one we take Semantic Web as an example. And I will have a short introduction about the background of big data and data stream at the beginning. And a conclusion and future research directions at the end.

We are all familiar with the words “Big Data”, because this is one of the hottest topics in the IT industry for the past 5 years. But how big is the data? Let’s have a short review. Google processes 24 PB of data every day. Facebook has 10 million photos and 3 billion “likes” each data. YouTube has 800 million visitors every month. And Twitter is doubling its size every year. So the most significant issue for processing these data comes from the size of data. And the flip side of size is speed. Normally, these data can not be processed in a single machine, so it often requires a parallel and distributed solution. But the network communication in the parallel and distributed processing system may cause a bottleneck of the system. The last issue is the dynamic of data. Data is changing all the time, and usually the newest ones have the most useful values. So the best way of describing data is data streams.

Different from a persistent static relation which requires a batch oriented data processing, the transient dynamic data streams require a real-time stream processing. In order to provide a stream processing, we need to be able to add or remove computational nodes based on the current load in the architecture level; and I also want to be able to withdraw old results and take new coming data into account in the application level.

The purpose of this dissertation is to study about the process for join operation in a parallel and continuous way. Join is a very popular and often used operation in the big data area. And when the data becomes large, we usually require a parallel join. According to the two different types of parallelism, we divide the joins studied in this dissertation into two different types, the first one is Data Driven join based on Data Parallelism and the second one is Query Driven join based on Task Parallelism.