THE UNIVERSITY OF
SYDNEY

Honours Research Proposal

Email this proposal to Josiah and Katie in Monday of Week2

Name:	Nicholas Hollingum	**SID:**	308193415
Supervisor:	Dr Bernhard Scholz	**Desk #**	3 East - 42
Tentative Project Title	Obtaining Fault Tolerance in the Synchronous Dataflow Model		

Abstract

The idea of stream processing is not new, with research dating back to the early 60s on Digital Signal Processing and Block Schematic programs. It has gained new interest with the advent of multi-core/multi-processor machines and cloud computing, as little research has been done on stream processing in this context. In this work we examine the stream processing paradigm restricted to the Synchronous Data Flow model (SDF). A program is represented as a set of blocks (actors) and arcs (channels) in a computational graph, where data (tokens) flowing between actors are produced/consumed at a constant rate per actor invocation. Actors do not share a common state, though they may or may not have their own. This work consists of two components:

- A framework for SDF program execution is implemented in the cloud to study the runtime behaviour of the cloud
- A theoretical study is undertaken to formalise the trade-offs between makespan, fault tolerance and communication costs of a stream program in the cloud and construct a solver that computes optimal placement of actors for the cloud.

5 (Five) References

- S. S. Battacharyya, E.A. Lee and P.K. Murthy, “Software Synthesis from Dataflow Graphs”, Kluwer Academic Publishers; 1996
- E. A. Lee, D. G. Messerschmitt, “Static scheduling of synchronous Data Flow Programs for Digital Signal Processing”, IEEE TOC V.36-1; Jan 1987
- M. Kudlur, S. Mahlke “Orchestrating the Execution of Stream Programs on Multicore Platforms”, PLDI'08; Jun 2008
- S. Amarasinghe, M. Karczmarek, W. Theis “StreamIt: A Language for Streaming Applications”, Massachusetts Institute of Technology; 2002
- S. Amarasinghe, W. Thies “An Empirical Characterization of Stream Programs and its Implications for Language and Compiler Design”, PACT10, Sep 2010