

WORLD

METEOROLOGICAL

Data assimilation using FPGA: shallow water system

Sabrina B. M. Sambatti¹, Haroldo F. Campos Velho¹, Andrea S. Charão², Helaine C. M. Furtado²

¹ Instituto Nacional de Pesquisas Espaciais – INPE, São José dos Campos (SP), Brazil ² Universidade Federal de Santa Maria, Santa Maria (RS), Brazil ³ Universidade Federal do Oeste do Pará, Santarém (PS),

sabrinabms@gmail.com / haroldo.camposvelho@inpe.br / andrea@inf.ufsm.br / helaine.furtado@gmail.com







Introduction

Data assimilation

Results

Numerical prediction can be performed by Neural network time integration of mathematical equations. 1) Trained to emulate Kalman filter However, there are errors in the model. A data Automatic designed by MPCA 2) fusion method combining observations and the model background is applied: data assimilation. **MPCA:** Multi-Particle Collision Algorithm Artificial neural network is used to carry out the data assimilation on a shallow water system, Applied to solve optimization problem: optimal designed to represent the ocean dynamics. design to Multi-Layer Perceptron (supervised The neural network (NN) is trained to neural network)

Best neural network architecture designed by the MPCA:

- a) Hidden layer: 1 b) # neurons (hidden): 10
- Activation function: tanh(x) **C**)
- Learning rate: 0.94 **d**)

emulate the Kalman filter, and it is implemented using hybrid computer system: CPU + FPGA (co-processor). The shallow water system is codified using finite difference executed on the CPU. The analysis is calculated on the FPGA. Results show that neural network is effective scheme for data assimilation. Data assimilation by using hardware device has similar performance in comparison with software implementation. One remark to be mentioned: FPGA needs less power than CPU to execute the same processing.

Shallow Water (SW) for

$$F_{obj} = penalty * \frac{(\rho_1 * E_{trein} + \rho_2 * E_{gen})}{\rho_1 + \rho_2}$$

$$penality = \underbrace{\left(c_1 * \left(e^{\#neuron}\right)^2\right)}_{\text{complexity factor-1}} \times \underbrace{\left(c_2 * (\#epoch)\right)}_{\text{complexity factor-2}} + 1$$





Data assimilation: field q(x,y) at time-step 40.

Green: Neural network Blue: true **Red:** Kalman Filter

ocean circulation (2D)

 $\frac{\partial u}{\partial t} - fv + g\frac{\partial q}{\partial r} + r_u u = F_u$







Neural network: FPGA implemantation

- Cray XD1 system:
- 12 CPUs (AMD) quad-core
- 6 FPGA Virtex-II Xilinx



Time evolution: field q(7,7)



Table: execution time	Process	Time (us)
(FPGA demands lower power the CPU)	Software	121709
	CPU to FPGA	181635
	FPGA proc.	2
	FPGA to CPU	9455
	FPGA (total)	209187

Acknowledgements

Authors thanks the Brazilian research agencies: (*) FAPESP (*) CNPq (*) CAPES



