LA-UR-01-3028

Approved for public release; distribution is unlimited.

Title: Parameterized K-Means Clustering for Rapid Hardware Development to Accelerate Analysis of Satellite Data

Author(s): Miriam Leeser
Pavle Belanov
Michael Estlick
Maya Gokhale
John Szymanski
James Theiler

Submitted to: Fifth Annual Workshop on High Performance Embedded Computing (HPEC 2001)
MIT Lincoln Laboratory
September 25-27, 2001

Los Alamos
NATIONAL LABORATORY

Parameterized K-means Clustering for Rapid Hardware Development to Accelerate Analysis of Satellite Data

Miriam Leeser, Pavle Belanov, Michael Estlick
Department of Electrical and Computer Engineering
Northeastern University, Boston MA 02115

Maya Gokhale, John Szymanski, James Theiler
Space and Remote Sensing Sciences Group,
Los Alamos National Laboratory, Los Alamos, NM 87545

Reconfigurable hardware has successfully been used to obtain speed-up in the implementation of image processing algorithms over purely software based implementations. At HPEC 2000 [1], we described research we have done in applying reconfigurable hardware to satellite image data for remote sensing applications. We presented an FPGA implementation of K-means clustering that exhibited two orders of magnitude speedup over a software implementation.

The algorithm, k-means clustering, is an iterative approach for unsupervised clustering. The basic steps are:

1. Initialize class centers
2. Repeat until no more pixels change classes:
   2a. Assign each pixel to the class with the closest center
   2b. Recalculate centers (mean values) for each class

When complete, each pixel in the classified image is represented as a pointer to a class. The class is represented by the mean value of all the pixels that belong to it, thus summarizing the spectral information of the input.

The implementation presented at HPEC 2000 was optimized for one type of data set only, and could not easily be adapted to other data sets. Satellite images come in many shapes and sizes, and vary in terms of number of spectral channels and size of the image. In addition, different numbers of clusters may be needed for different applications. We have re-implemented k-means using a parameterized design style which allows us to adapt to many different data sets and clustering tasks. Our new design is parameterized by the size of the input image, number of spectral channels per pixel, number of bits per pixel and number of clusters. In our implementation, the number of clusters is given by the image analyst and fixed during clustering.

Specifically, the parameters to our implementation are defined as:

K -- number of clusters
C -- number of channels per pixel
B -- number of bits per channel
P -- log_2 (total pixels in image)
ie P=20 signifies 2^20 pixels (1024x1024)

Our design is implemented on a Wildstar board from Annapolis Microsystems. The design uses one Virtex 1000. The current implementation assumes that a single pixel (C * B bits) can be read in one clock cycle, which for our board constrains us to C*B <= 128. In addition, we have constrained the number of classes to being a power of two.

We have compared an 8 cluster, 10 channel, 12 bits per channel design using the RTL methodology to the same design generated using our new paramterized methodology. Here are the results of the 2 designs:

Design	RTL	Parameterized
	----	----------
Area in slices	9420 out of 12288 76%	8884 out of 12288 72%
Total gate equiv.	406,536	433,474
Max delay		
K_Clk	27.967ns	28.773ns
M_Clk	15.680ns	15.855ns
P_Clk	14.153ns	15.275ns
Max Frequency	63.78Mhz	63.07Mhz
Place-and-route time	3:32:56	1:04:49

The RTL design was implemented in VHDL as a set of processes and uses Synplicity design tools plus Xilinx place-and-route to generate a bit stream. The parameterized design also makes use of Synplicity design tools, but builds the design out of components built using the Xilinx CoreGen libraries. The design adapts to the number of bits of processing in the width of accumulators and the size of datapath components both as a direct result of the parameters and indirectly due to growth of operands during the computation.

The two designs generated with these competing design styles have comparable area and speed. The parameterized design has more gate equivalents packed into fewer Virtex slices; we believe this is due to the efficient use of resources by CoreGen.

The big difference between the two designs is the ease of generating new implementations using the parameterized methodology, and the fact that the parameterized design has a place-and-route time more than 3 times faster than the RTL design without a loss of quality in the results.

We have implemented k-means clustering on the Wildstar board for the following combinations of parameters using the parameterized design style:

K	C	B	P
16	5	10	20
8	14	8	20
8	20	6	20
8	10	12	20

Each new design took a few minutes of setting the parameters plus approximately an hour to place-and-route. Changing the RTL design to generate a new design takes at least a day of changes as well as 3 and a half hours of place-and-route time. In addition, it is much easier to introduce errors when changing the RTL code vs. the parameterized code.

While these results reflect the use of one Virtex 1000 chip on a Wildstar board, in reality we parallelize the implementation by using 2 Virtex chips and having each process half the image. This change effects the host code only, and the speedup is the same for both the RTL code and the parameterized code.

Our current approach is constrained by the memory bandwidth to the Virtex chip. We process one pixel each clock cycle, and fully pipeline the processing. In the future, we plan to investigate adding pipelining to our parameterized implementation. In other words, the structure of the pipeline would change depending on the number of clock cycles required to fetch the input data. In addition, we are investigating applying this parameterized approach in other domains, including parameterized floating point modules for reconfigurable hardware.

[1] Miriam Leser, Michael Estlick Natasha Kitaryeva, James Theiler and John Szymanski, "Applying Reconfigurable Hardware to Segmentation for Multispectral Imagery" in High Performance Embedded Computing (HPEC 2000), September 2000.