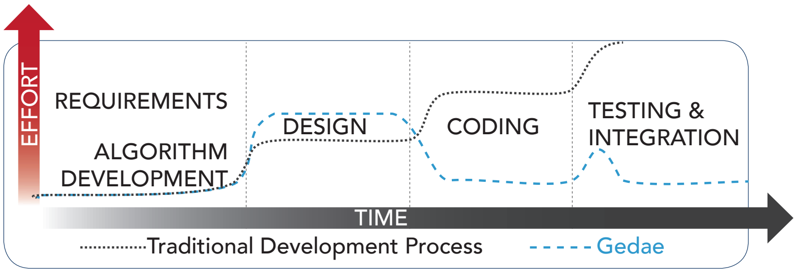
Issue No. 2, November 2012

GEDAICA

***pl n 1.*of or relating to the culture, artifacts, customs of Gedae**

DOING MORE WITH LESS?



Whether you are a big company or a small company -- striving to grow or downsizing, the phrase "More with Less" has significant meaning to you today. The challenging global economy and competitive pressures have placed this phrase on the lips of every leader in the industry. How can we become more effective and efficient with fewer resources?

From interviews with customers and prospects over the last 60 days we have discovered that the time between algorithm development to final integration and test is the largest cost for most programs -- new or legacy. Gedae's announcement of the !dea Development Environment has been well received as an answer to this question. Its suite of powerful software tools addresses the complexity of programming multi-core and multi-processor systems with complex memory architectures. The !dea language, automated compiler technology and analysis tools enable efficient and effective development of distributed heterogeneous processing for new and existing systems.

GEDAE IS GROWING:

CSP, Inc.

Today Gedae, Inc. announced support for it's !dea™ Development Environment on CSPI's TeraXP™ OpenVPX Rugged Embedded Servers and ATCA High Performance Embedded Computing Platforms enabling faster application development.

"We are pleased to offer our customers a software tool that can significantly reduce development time and generate application code optimized for the robust parallel processing capabilities of our latest generation of multicore platforms," said Paul Martino, CSPI's Director of U.S. Sales.

CURTISS-WRIGHT CONTROLS

Curtiss-Wright Controls Announces Gedae Software Development Support for its 3rd Generation Intel® Core™i7 CHAMP-AV8 DSP Engine.

"We are excited to bring support for Gedae's software development tools to our high performance multiprocessor CHAMP-AV8 DSP engine," said Lynn Bamford, vice president and general manager, Curtiss-Wright Controls Defense Solutions. "The combination of this advanced board and Gedae's tools provides a high-performance, scalable and portable application development environment for systems integrators developing demanding DSP applications. The Curtiss-Wright – Gedae partnership has migrated forward across several generations of DSP boards offering a seamless migration path for our mutual customers."

THE WATER COOLER

Submit questions for THE WATER COOLER via email at thewatercooler@gedae.com.

How easy is it to convert MATLAB to Idea code?

It is not very difficult to convert MATLAB into Idea code. We had one customer that needed both a MATLAB and Idea version of their deployed application. The MATLAB version had 974 semicolons and the Idea version had 977 semicolons. The almost line-for-line conversion indicates that MATLAB and Idea are at the same level of abstraction. There is some conversion required.

The next step is to add some data reorg code to the new Idea code and to invoke the compiler to build the parallel version. Depending on the complexity of the parallelization strategy this can take a few minutes to a few hours.

Why is conversion to Idea Code necessary to garner the benefits of the Idea Compiler?

Idea uses array algebra and MATLAB uses matrix algebra. Array algebra is a little more syntax but is more general purpose that matrix algebra. For example, matrix algebra cannot express morphological operations easily so MATLAB uses a toolkit library for those functions. Those functions shown below, illustrating dilation, are easily expressed in array algebra.

MATLAB: *`out = ordfilt2(in,9,true(3,3));`*

RESEARCH JOURNAL : $out_{r,c} = \underset{rk,ck}{MAX}(in_{r+rk,c+ck})$

IDEA: *`out[r][c]  >?= in[r+rk][c+ck];`*

DOLLARS AND SENSE

REMINDER: Limited time offer expires 31-Dec-12. Gedae is pleased to offer a 20% discount on new or reactivated Gedae licenses to Gedae Niagara. Please call [(856) 231.4458] or email [sales@gedae.com] if you are interested in these special upgrade offers.

CODE, TOOLS & TIPS Video Processing Example

Is there an example you'd like to see in CODE, TOOLS & TIPS? – We're looking for industry parallel processing applications to implement in Idea Language and publish results. Submit yours today via cc@gedae.com.

The MATLAB algorithm in *Figure 1* was derived from an article by Andrew Kirillov[1]. The same application, written in Idea is shown in *Figure 2*. The algorithm detects motion in a video. *Figure 3* shows the result of processing a video of satellite weather imagery.

```
function out = MotionFilters(in,ref,Thres)
  diff = abs(in-ref);
  c = zeros(size(in));
  ind = diff>Thres;
  c(ind) = in(ind);
  er = ordfilt2(c,1,true(3,3));
  di = ordfilt2(er,9,true(3,3));
  gau = fspecial('gaussian',[5,5],4);
  smo = conv2(di,gau,'same');
  out = edge(smo,'sobel');
end
```

Figure 1: MATLAB (Octave) Code

```
out MotionFiltersI(float in[i][j],
                   float ref[i][j], float Thres) {
  diff[r][c]  = abs(in[r][c] - ref[r][c]);
  in2[r][c]   = diff[r][c] > Thres ? in[r][c] : 0;
  in3[i2][j2] = Erosion(in2,3,3);
  in4[i2][j2] = Dilation(in3,3,3);
  ker[ik][jk] = GaussianKernel(5,5,4);
  in5[i3][j3] = Filter(in4,ker);
  out[i][j]   = Sobel(in5);
}
```

Figure 2: Idea Code

This algorithm is a good demonstration of Gedae's Idea™ Development Environment (Idea DE) because it can show 1) the ease of converting MATLAB to Idea, 2) the clarity of the Idea code and 3) the efficiency of applications built by the Idea™ compiler. The reason is that each of the operations shown are 1, 2 or 3 lines of Idea code as shown at the bottom of the figure. The indices shown are C like indices.

The use of [operator]= is called a collapsing operator. For example, the += used to convolve the kernel with the processed image in the Filter function is:

out[r1][c1] += MK[rk][ck]*in[r1+rk][c1+ck];

This is equivalent to the equation as you might see it in a paper:

$$out_{r1,c1} = \sum_{rk,ck} MK_{rk,ck} * in_{r1+rk,c1+ck}$$

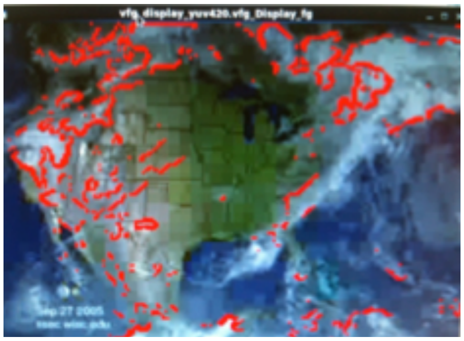


Figure 3: Results of Processing

The code required to specify the parallelization strategy is show in *Figure 4*. The function kernelzpartf extracts overlapping tiles to be processed in parallel on the 4 core processor referenced in this work. The tiles are overlapping to accommodate the kernel operations in the processing such as the Erosion & Dilation. The indices on the left hand side of the variable are family indices and indicate there are #r * #c tiles each with #i1 rows and #j1 columns.

```
[r][c]in1[i1][j1]  = kernelzpartf(in,Ksz_total,Ksz_total);
[r][c]in2[i][j]    = MotionFiltersI([r][c]in1,[r][c]ref1,...);
out[i1][j1]        = rcconcat_fam([r][c]in2);
```

Figure 4: Idea code for parallelization specification

Figure 5 shows the performance numbers for the algorithm on the same platform from Octave to optimized 4 core processing built using the Idea DE. The speed up from 5 to 500 frames is indicative of the quality and efficiency of the code built by the Idea compiler.

The total implementation of this algorithm including conversion and implementation on a four core processor took about 2 hours. Details can be found in the full whitepaper on our website: www.gedae.com.

Software	mSec/f	Speed Up	f/s*
MATLAB (Octave)	190	-	5.3
Idea DE	75	2.53 x	11.3
Idea Optimized 1 Core	3.5	63 x	253
Idea Optimized 4 Cores †	1.1	173 x	502

* Idea version includes display ~33% of load on 4 cores
† VMware with other apps running

Figure 5: Results Table

[1] Motion Detection Algorithms, http://www.codeproject.com/Articles/10248/Motion-Detection-Algorithms, 27 Mar 2007, Andrew Kirillov, Software Developer (Senior), United Kingdom