Using the PlayStation®3 for speeding up metaheuristic optimization

S. Van Volsem & S. Neirynck
Ghent University, Department of Industrial Management
sofie.vanvolsem@ugent.be

1 Scientific computing with the PlayStation®3

Traditional computer software is written for serial computation. To solve an optimization problem, an algorithm or metaheuristic is constructed and implemented as a serial stream of instructions. These instructions are executed on a central processing unit (CPU) on one computer. Parallel computing uses multiple processing elements simultaneously to solve a problem. This is accomplished by breaking the problem into independent parts so that each processing element can execute its part of the algorithm simultaneously with the others. The processing elements can be diverse and include resources such as a single computer with multiple processors, several networked computers, specialized hardware, or any combination of the above. Today most commodity CPU designs include single instructions for some vector processing on multiple (vectorized) data sets, typically known as SIMD (Single Instruction, Multiple Data). Modern video game consoles and consumer computer-graphics hardware rely heavily on vector processing in their architecture. In 2000, IBM, Toshiba and Sony collaborated to create the Cell Broadband Engine (Cell BE), consisting of one traditional microprocessor (called the Power Processing Element or PPE) and eight SIMD co-processing units, or the so-called Synergistic Processor Elements (SPEs), which found use in the Sony PlayStation®3 among other applications. The computational power of the Cell BE or PlayStation®3 can also be used for scientific computing. Examples and applications have been reported in e.g. Kurzak et al. (2008), Bader et al. (2008), Olivier et al. (2007), Petrini et al. (2007).

2 Example - Metaheuristic optimization

In this work, the potential of using the PlayStation®3 for speeding up metaheuristic optimization is investigated. More specifically, we propose an adaptation of an evolutionary algorithm with embedded simulation for inspection optimization, developed in Van Volsem et al. (2007), Van Volsem (2009a) and Van Volsem (2009b). Thereto, two issues needed to be addressed :

1. the metaheuristic itself needed to be adapted to make it suitable for parallel computation and vector processing, and

2. the random number generation required by the metaheuristic had to be adapted for parallel computing as well.

We thus adapted the algorithm to eliminate branches and optimized the code using standard techniques such as loop unrolling and vectorization. We adapted the random number generation process : we developed and implemented a Box-Müller transform on uniform random numbers generated with an 128-bit Mersenne twister.

Algorithm 1 original EA

```
Create initial sorted population
for generation = 1 to number of generations do
  Create offspring
  for solution = 1 to M do
    for stage = 1 to n do
      Calculate process values
      Calculate inspection cost
    end for
    Calculate TIC
  end for
  Sort population
end for
Take winner
```

Algorithm 2 adapted version of the EA for computation on the PS3

```
Create initial sorted population
for generation = 1 to number of generations do
  for solution = 1 to M do
    Create offspring
    Calculate TIC on SPE (IN PARALLEL)
  end for
  Sort population
end for
Take winner
```

3 Summary & Conclusion

We designed an optimized parallel implementation of an evolutionary algorithm and simulation for optimizing inspection strategies for multi-stage processes on the PlayStation®3. We adapted the algorithm itself as well as the embedded random number generation process. The original algorithm calculation time was >1 hour for 200 generations ; re-writing the code with SIMDizing the RNG led to a calculation time of 5'59" on a single core of a 2.5GHz AMD Phenom processor. The further porting and optimizing of the code to make it suitable for running on the cell broadband engine led to a calculation time of 14" on a PlayStation®3. We thus realized a speedup factor of more then 256 in comparison with the original algorithm in Van Volsem et al. (2007), and showed the PlayStation®3 suitable for scientific computing.