

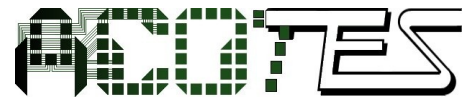
The ACOTES Project



Stream-computing extensions and tools
from the ACOTES project

Albert Cohen, Xavier Martorell et al.

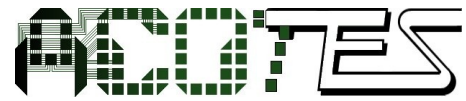
- Project goals & consortium
- ACOTES developments
- Results
- Future directions



The ACOTES Project



- Running June 1st, 2006 - May 31st, 2009
- Project Goals
 - Programming tools for high performance embedded architectures
 - Develop a methodology for specifying and implementing streaming data applications for such architectures

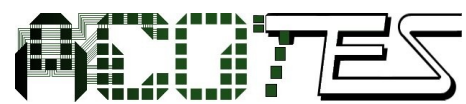


Project consortium



- ARMINES, France
- IBM Haifa Labs, Israel
- INRIA-Futurs, France
- NOKIA Research Center Bochum, Germany
 - until May 31, 2007
- NXP Semiconductors, The Netherlands
 - from October 1, 2006
- Philips Electronics/Semiconductors, The Netherlands
 - until September 30, 2006, all responsibilities were transferred to NXP Semiconductors

- Silicon Hive, The Netherlands
 - from June 1, 2007
- ST Microelectronics, Switzerland
 - until May 31, 2007, all responsibilities transferred to ST Microelectronics, Italy
- ST Microelectronics, Italy
 - from June 1, 2007
- UPC, Spain



Acotes developments



- Acotes programming model and supporting tools
 - Based on Mercurium and GCC4
- Automatic loop transformation and vectorization in GCC4
- Demonstration on FMradio, 802.11a, and a set of multimedia kernels
- Architectures
 - SMP, Cell/B.E., Ne-XVP, and xSTream

```
int main()
{
    char c;

    #pragma acotes taskgroup
    while (fread(&c, sizeof(c), 1, stdin)) {
```

```
        #pragma acotes task input(c) output(c:bypass) device (spu)
```

```
        if ('A' <= c && c <= 'Z') c= c - 'A' + 'a';
```

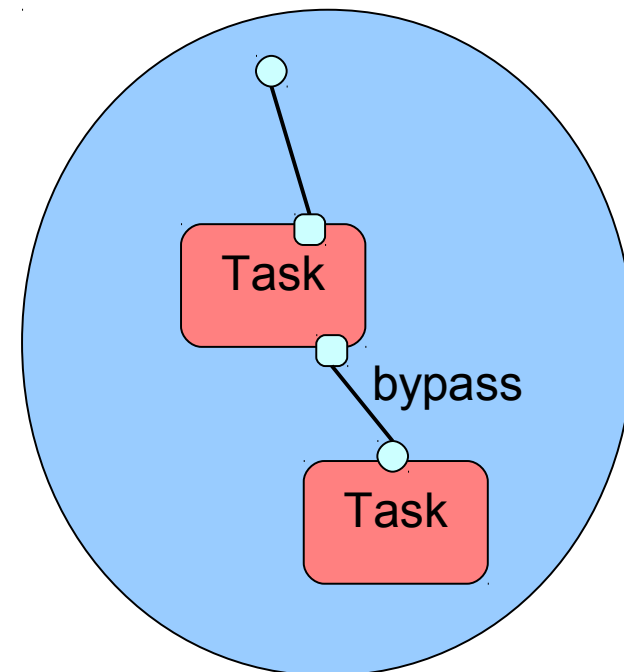
```
        #pragma acotes task input(c:bypass)
```

```
        fwrite(&c, sizeof(c), 1, stdout);
```

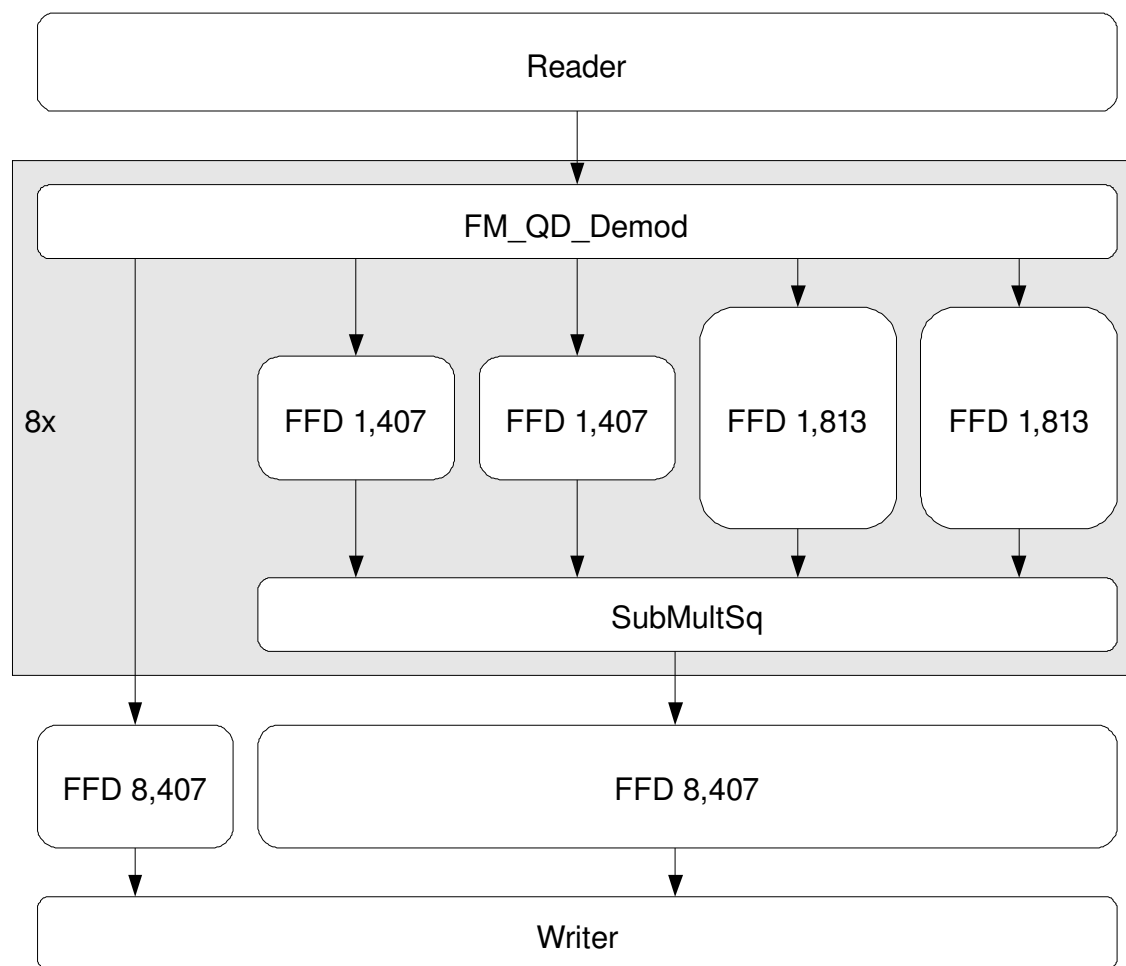
```
    }
```

```
    return 0;
```

```
}
```



- Task diagram
 - Reader/Writer in full processor
 - Filter tasks on accelerators



FMradio source code (Acotes)



```
# pragma acotes taskgroup
{
  while (8 == fread (read_buffer, sizeof (pair_t), 8, input_file)) {
    iterations++;
# pragma acotes port output (read_buffer[8]:rb_bp)
    for (i=0; i < 8; i++) {
      pair = read_buffer[i];
# pragma acotes task copyinstate (fm_qd_conf) input (pair:rb_bp) output (fm_qd_value) output (fm_qd_value:fm_qd_bp)
      fm_quad_demod (&fm_qd_conf, pair.first, pair.second, &fm_qd_value);
# pragma acotes task copyinstate (lp_11_conf) input (fm_qd_value) output (band_11)
      ntaps_filter_ffd (&lp_11_conf, 1, &fm_qd_value, &band_11);
# pragma acotes task copyinstate (lp_12_conf) input (fm_qd_value) output (band_12)
      ntaps_filter_ffd (&lp_12_conf, 1, &fm_qd_value, &band_12);
# pragma acotes task copyinstate (lp_21_conf) input (fm_qd_value) output (band_21)
      ntaps_filter_ffd (&lp_21_conf, 1, &fm_qd_value, &band_21);
# pragma acotes task copyinstate (lp_22_conf) input (fm_qd_value) output (band_22)
      ntaps_filter_ffd (&lp_22_conf, 1, &fm_qd_value, &band_22);
# pragma acotes task input (band_11,band_12) output (resume_1)
      subtract (band_11, band_12, &resume_1);
# pragma acotes task input (band_21,band_22) output (resume_2)
      subtract (band_21, band_22, &resume_2);
# pragma acotes task input (resume_1,resume_2) output (ffd_value:ffd_bp)
      multiply_square (resume_1, resume_2, &ffd_value);

      fm_qd_buffer[i]= fm_qd_value; ffd_buffer[i]= ffd_value;
    }
# pragma acotes task copyinstate (lp_2_conf) input (fm_qd_buffer[8]:fm_qd_bp) output (band_2)
      ntaps_filter_ffd (&lp_2_conf, 8, fm_qd_buffer, &band_2);
# pragma acotes task copyinstate (lp_3_conf) input (ffd_buffer[8]:ffd_bp) output (band_3)
      ntaps_filter_ffd (&lp_3_conf, 8, ffd_buffer, &band_3);
# pragma acotes task input (band_2,band_3) state (output[2], output_short[2]) copyinstate (output_file,text_file)
      {
        stereo_sum (band_2, band_3, &output[0], &output[1]);
        output_short[0]= dac_cast_trunc_and_normalize_to_short (output[0]);
        output_short[1]= dac_cast_trunc_and_normalize_to_short (output[1]);
      }
  }
}
```

ADOTES FMradio source code (OpenMP with streaming)

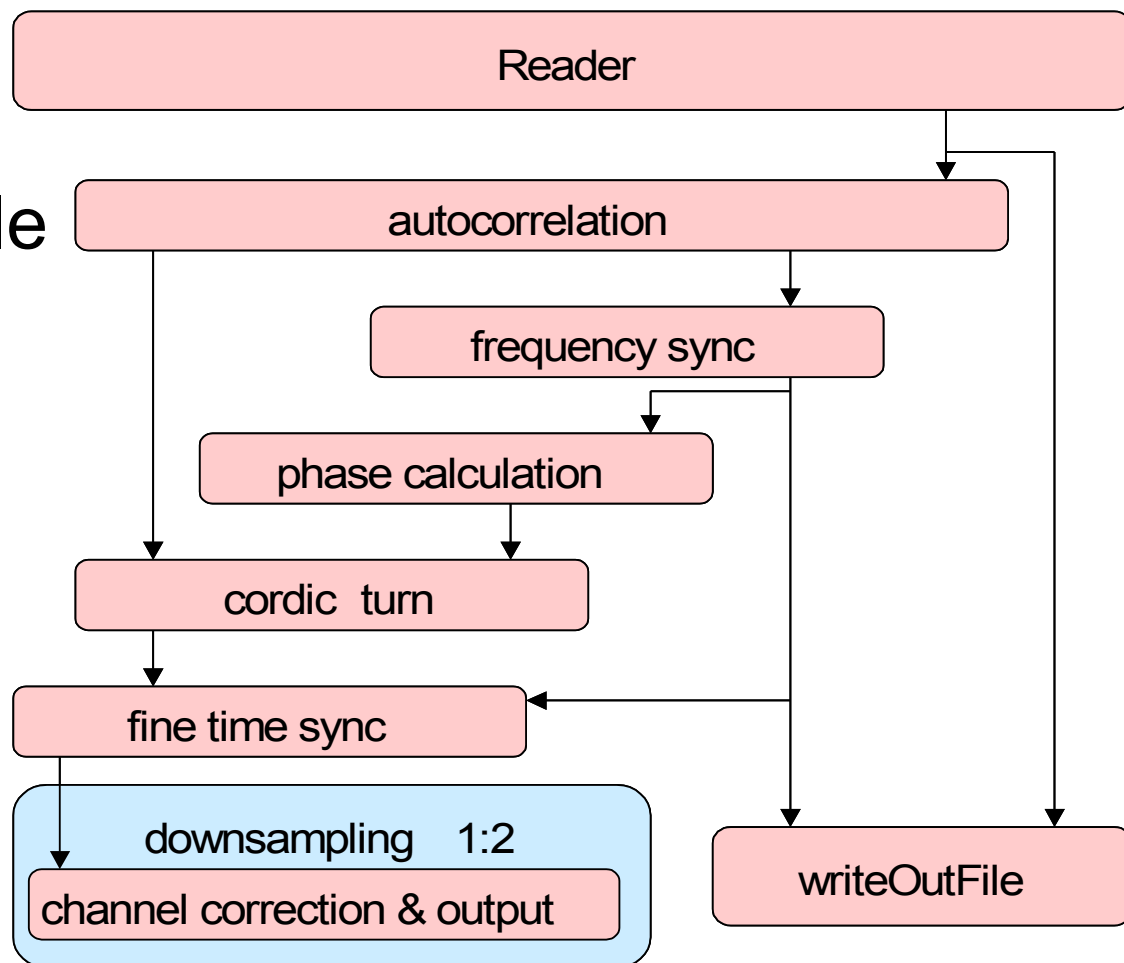


```
#pragma omp parallel num_threads (16) default (none) \
    firstprivate (input_file, output_file, lp_2_conf, lp_11_conf, lp_12_conf,
                lp_21_conf, lp_22_conf, lp_3_conf, fm_qd_conf) \
    private (read_buffer, fm_qd_buffer, ffd_buffer, band_2, band_3, \
            band_1, band_12, resume_1, band_21, band_22, resume_2, \
            output1, output2, output_short, i, j, count)
{
#pragma omp single
{
    while (16 == fread (read_buffer, sizeof(float), 16, input_file))
    {
#pragma omp task input (read_buffer) output (fm_qd_buffer) firstprivate (fm_qd_conf)
        {
            for (i = 0; i < 8; i++)
                fm_quad_demod (&fm_qd_conf, p1, p2, &fm_qd_buffer[i]);

#pragma omp task input (fm_qd_buffer) output (band_11) firstprivate (lp_11_conf)
                for (i = 0; i < 8; i++)
                    ntaps_filter_ffd (&lp_11_conf, 1, &fm_qd_buffer[i], &band_11[i]);

#pragma omp task input (fm_qd_buffer) output (band_12) firstprivate (lp_12_conf)
                for (i = 0; i < 8; i++)
                    ntaps_filter_ffd (&lp_12_conf, 1, &fm_qd_buffer[i], &band_12[i]);
        }
    }
}
...
}
```

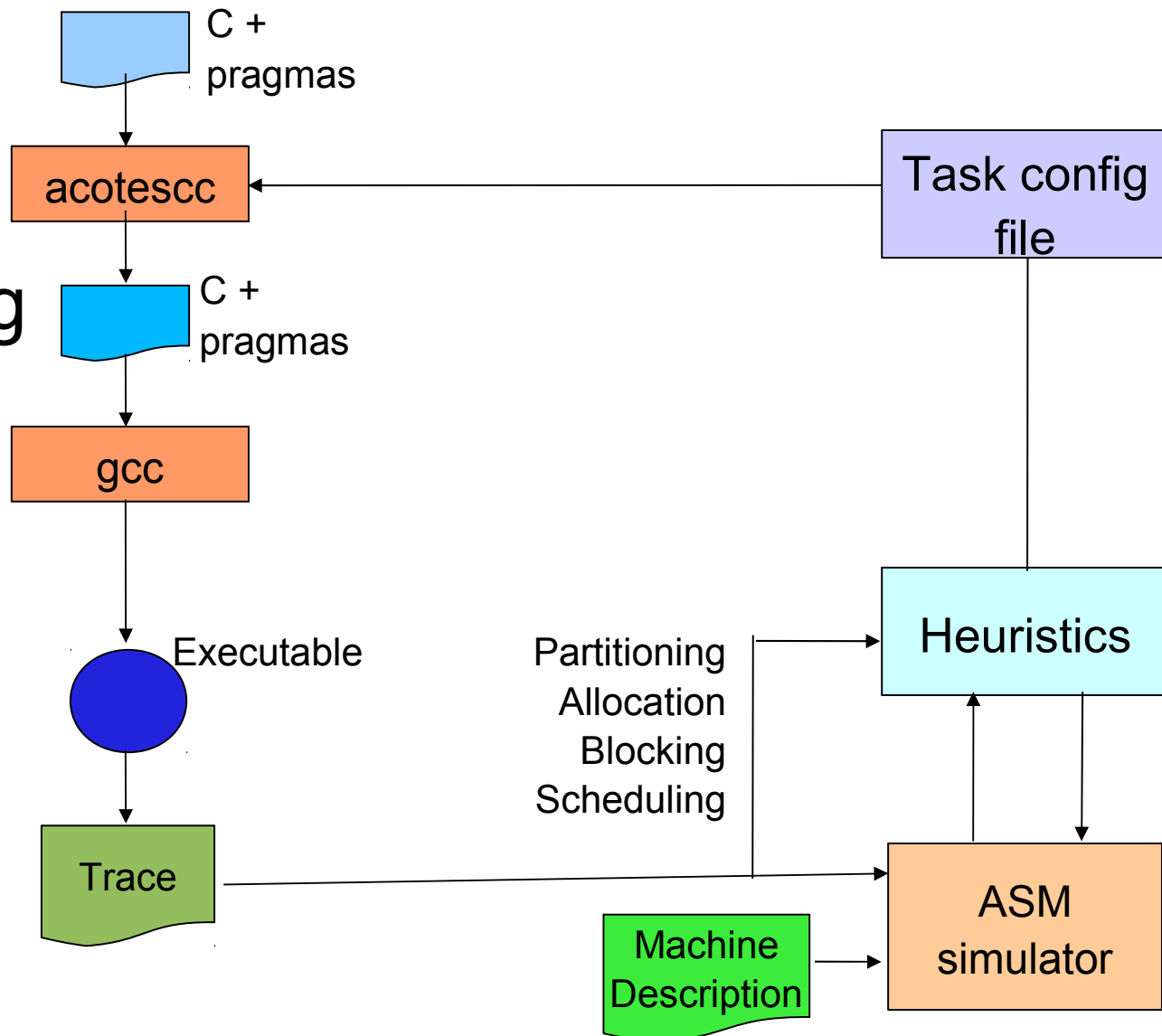
- Task diagram
 - Reader/WriteOutFile in main cores



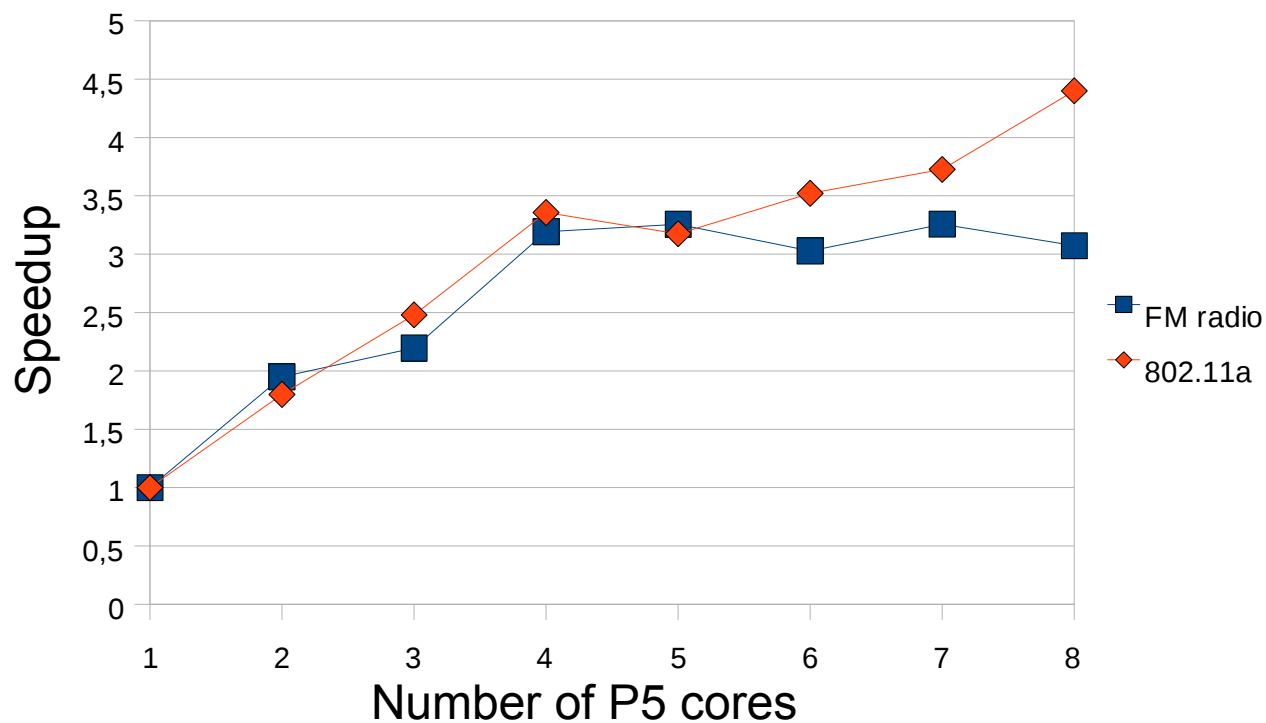
Compilation and optimization



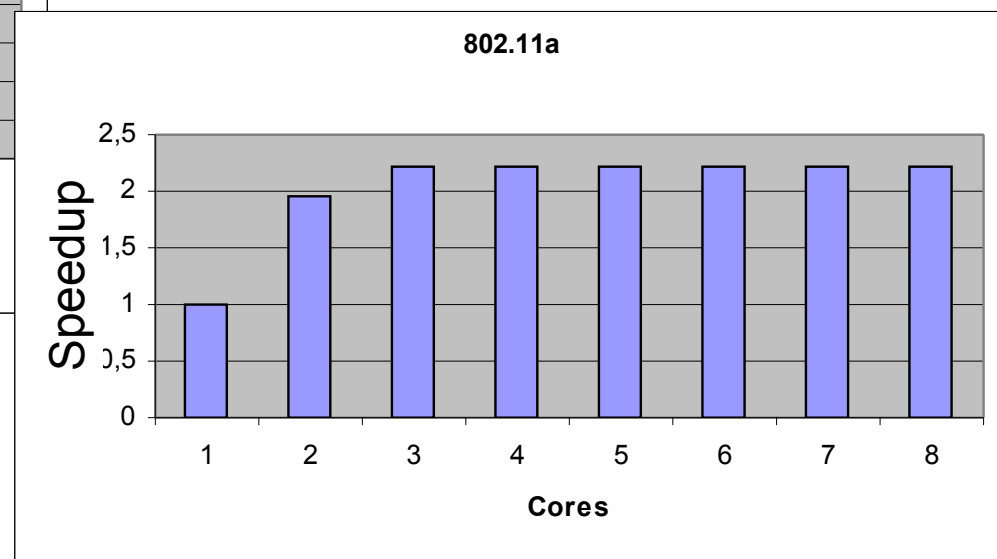
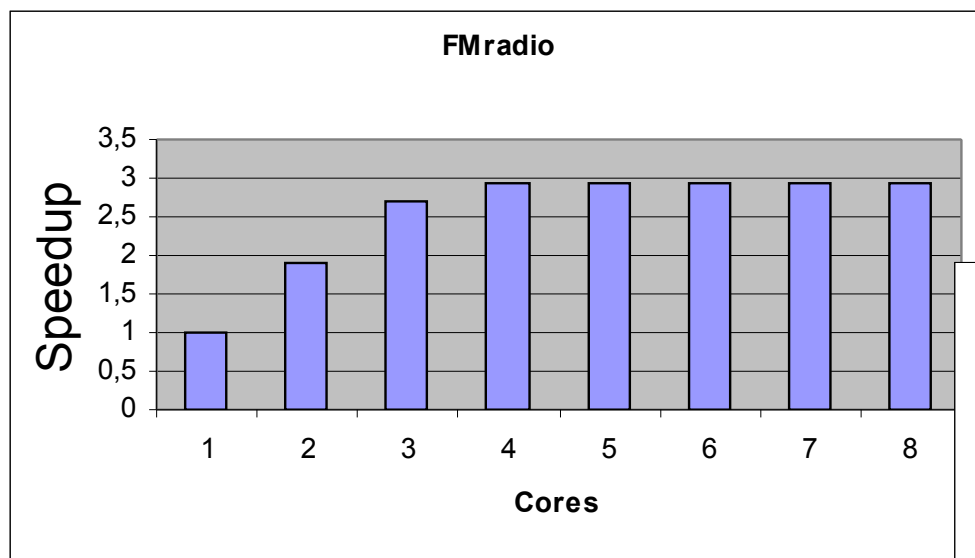
- Acotescsc prototype S2S
- GCC4 streaming pragmas
- Abstract streaming machine for task/loop optimizations



- FMradio and 802.11a applications
- On Power5-based SMP node
 - 4 cores with 2 SMT threads



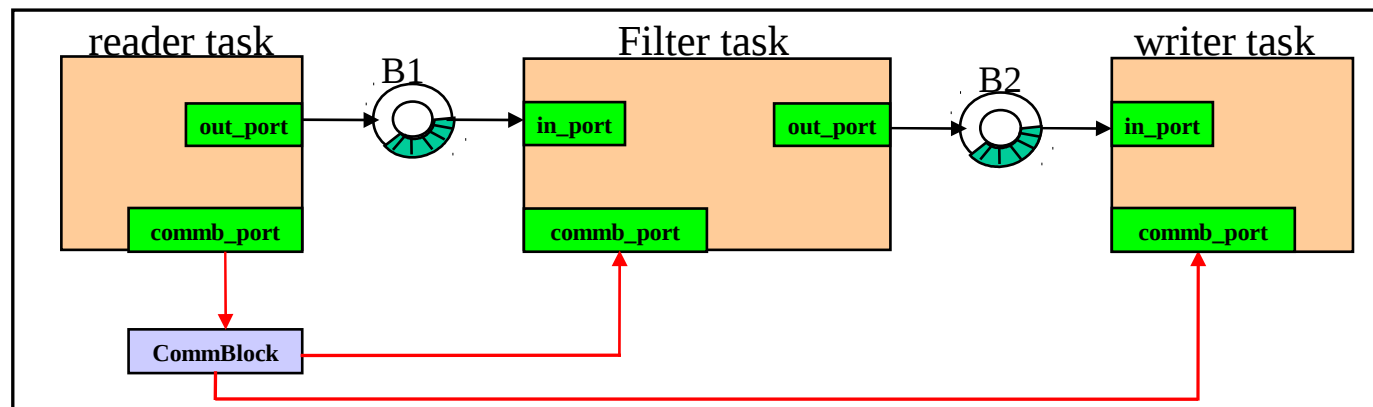
- Experiments on Ne-XVP simulation environment



MSF Execution Environment



Application level



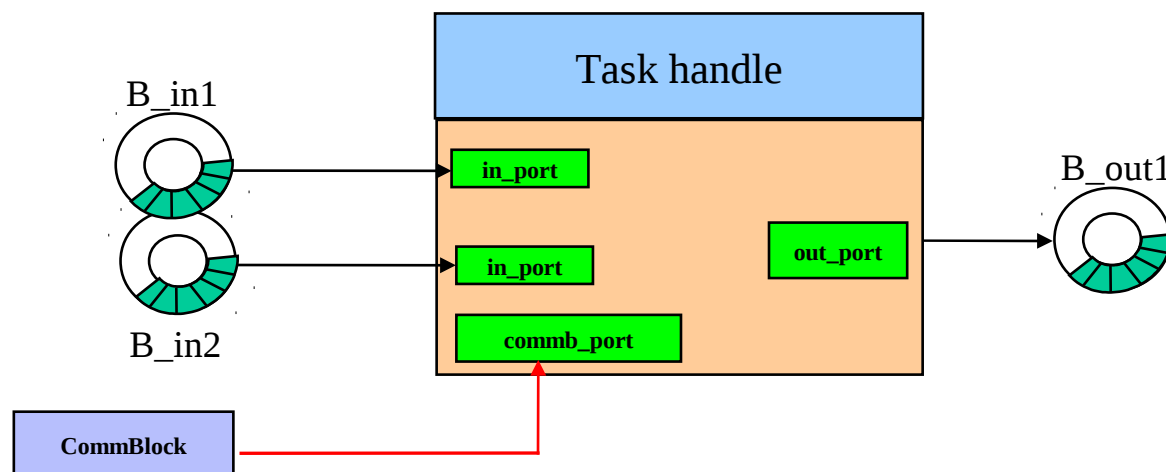
libmsf-PPU.so

libmsf-SPU.a

libmsf-PPU.so

A task

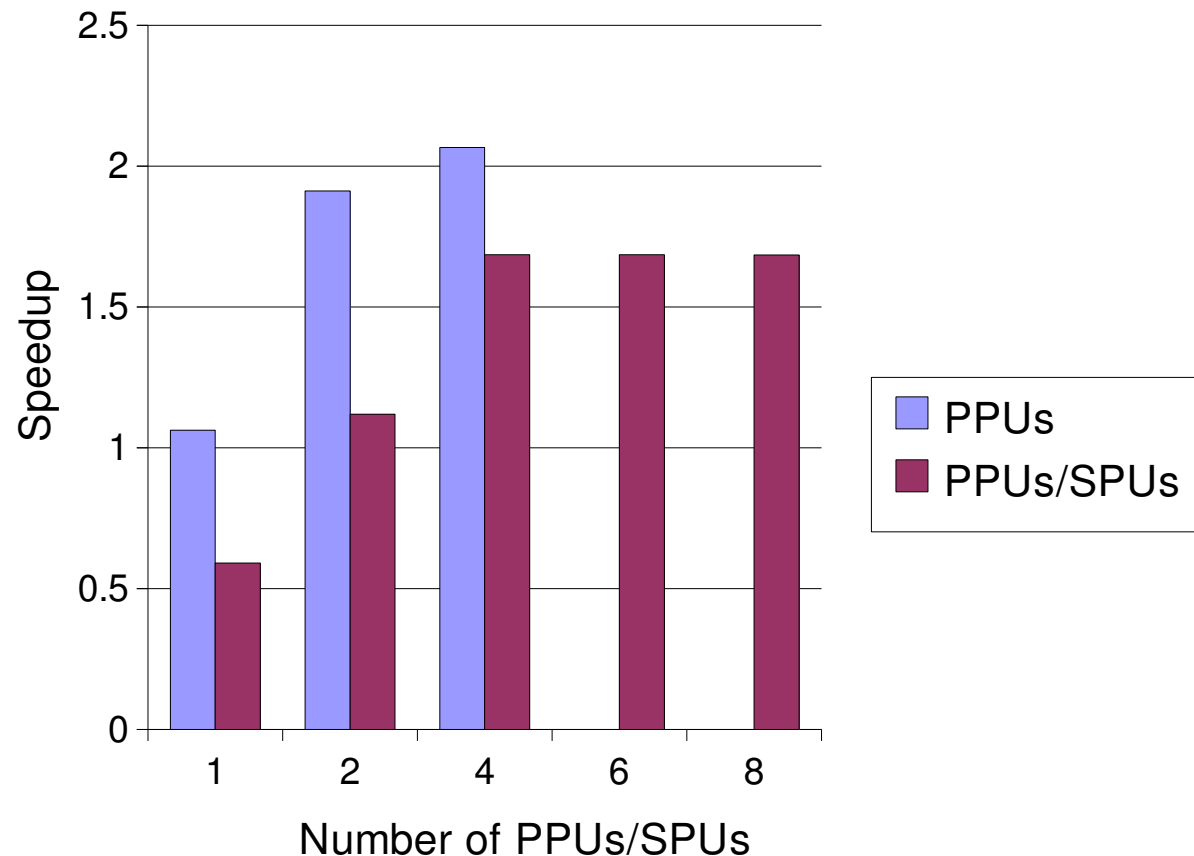
Task level



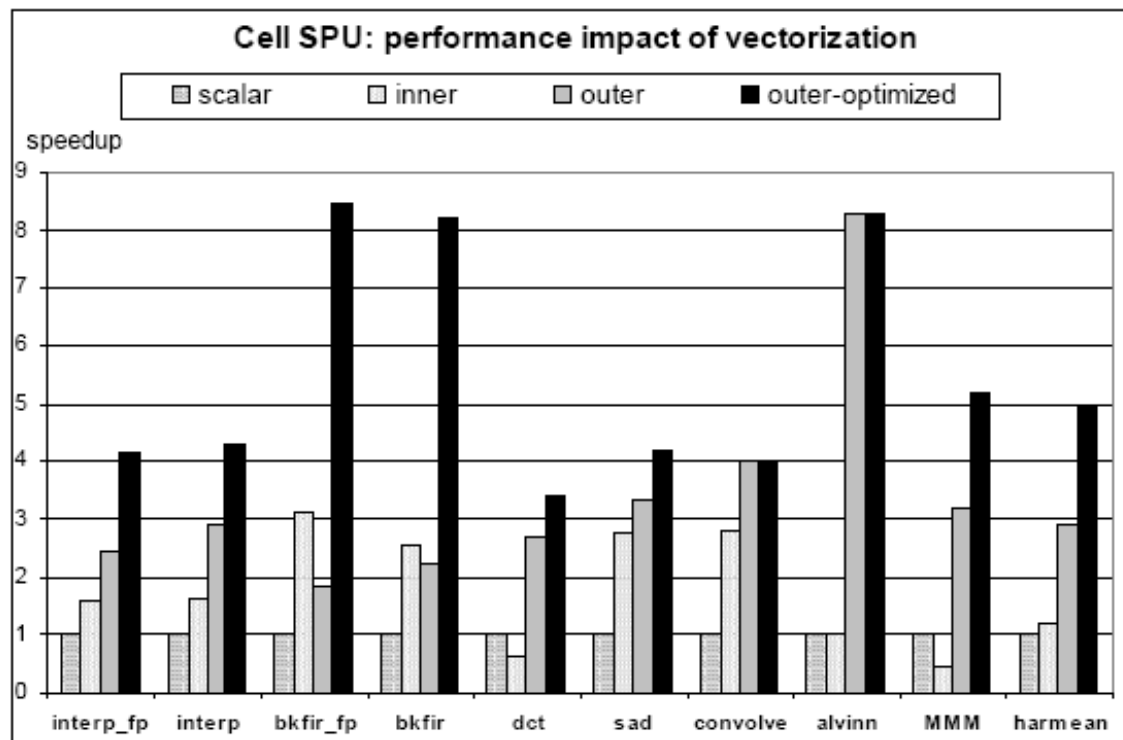
Performance of FMradio



- PPU vs PPUs/SPUs versions



- Impact on FMradio
 - 1.4x speedup with respect the streamized version
- Impact on multimedia kernels



- Erbium: intermediate language and lock-free run-time

