



Hipeac2 Programming Models and Operating Systems



Xavier Martorell, Roger Ferrer
Daniel Jiménez, Daniel Cabrera

BSC-UPC



Hipeac2 cluster meetings,
Barcelona, June 1st-2nd, 2008





OpenMP

- Standard
 - Official release: 3.0
 - Released two weeks ago
- Extensions
 - **Heterogeneous multi-core environments**

OpenMP for Cell

- Heterogeneous tasking

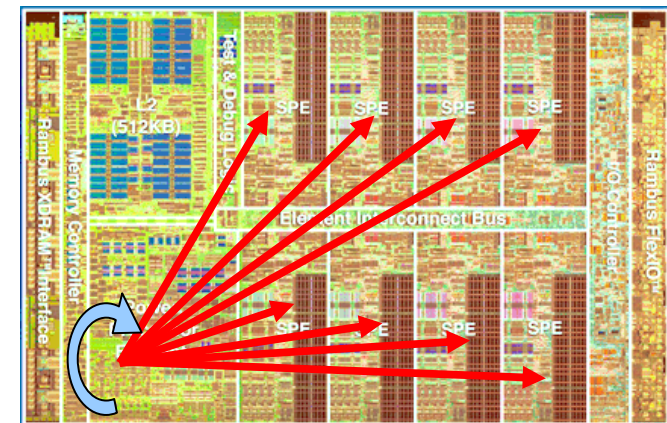
```

for (i=0; i<N; i++) {
  for (j=0; j<N; j++) {

#pragma spu task
  {
    int k = 0;
#pragma spu data input(C[i][j])
    for (k = 0; k < N; k++) {
#pragma spu data input(A[i][k],B[k][j])
#pragma spu block nest(3) factors(16,16,16)
      {
        C[i][j] += A[i][k] * B[k][j];
      }
    }
#pragma spu data output(C[i][j])
  }
}
}

```

- n-depth blocking
- DMA generation



Task spawning

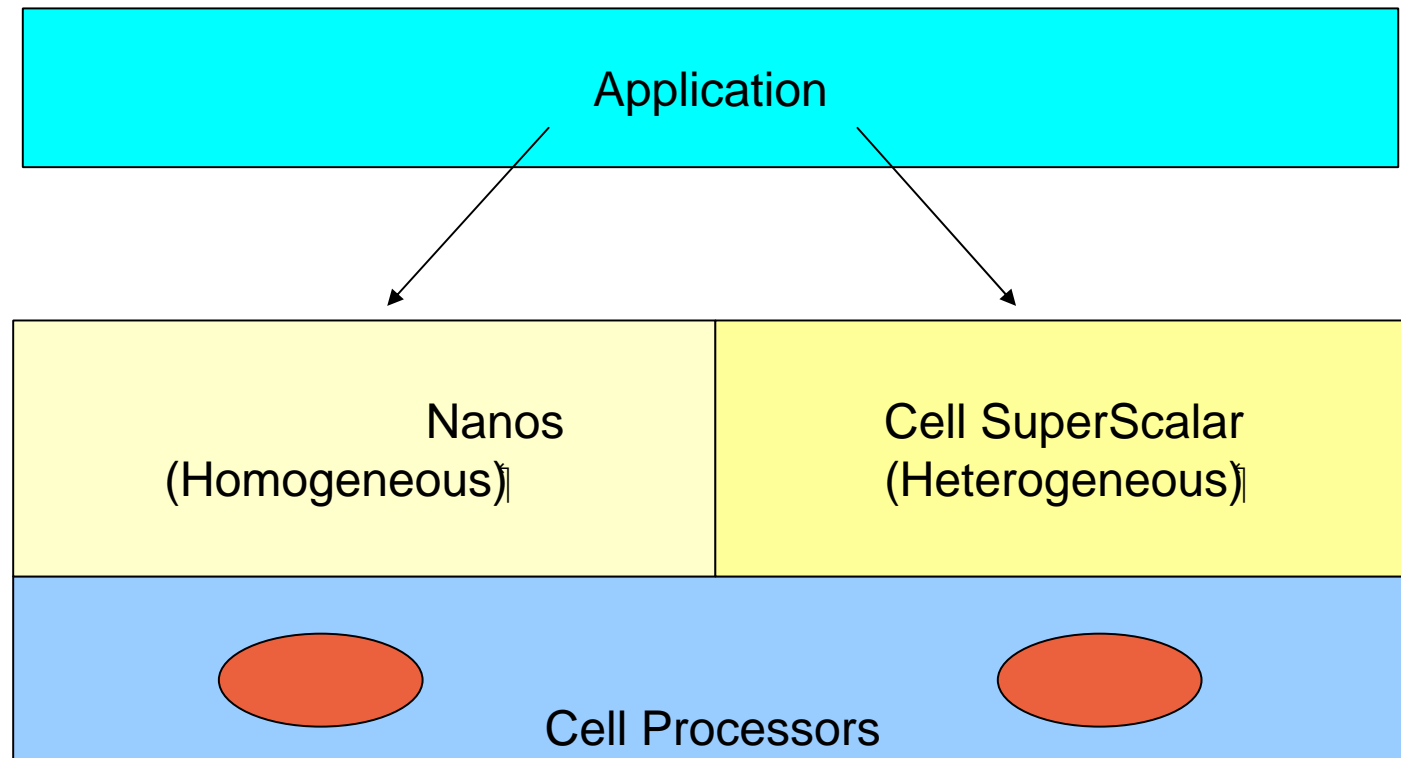
OpenMP for Cell

- Can use the SDK block

```
    for (i = 0; i < DIM_SIZE; i += 64)
    {
        for (j = 0; j < DIM_SIZE; j += 64)
        {
#pragma spu task
            {
#pragma spu data input(c[i:i+63][j:j+63])
                int ii, jj, kk;
                for (k = 0; k < DIM_SIZE; k += 64)
                {
#pragma spu data input(a[i:i+63][k:k+63], b[k:k+63][j:j+63])
                    MatMult_MxM (&c[i][j], &a[i][k], &b[k][j]);
#pragma spu data output(c[i:i+63][j:j+63])
                }
            }
        }
    }
```

OpenMP

- Infrastructure



OpenMP

- Specification of the directives - `spu task`
 - Creates a task to run on an available SPU
 - The task can execute any function available in the SPU execution environment
 - Emulated system calls (`printf...`)
 - Intrinsic functions, including SIMD

```
#pragma spu task  
{  
    // code  
}
```

OpenMP

- Specification of the directives - spu data
 - Indicates data transfers
 - Input and output
 - Can be specified at any time in the task
 - Include range specifiers

```
#pragma spu data input (A[i][k], B[k-1:k+1][j-1:j+1])
```

OpenMP

- Specification of the directives - `spu block`
 - Blocking of loop nests (2/3 levels)
 - Blocking factors indicated in the directive

```
#pragma spu block nest(3) factors (16, 32, 16)
```

OpenMP for FPGAs

```
/* FPGA code */
```

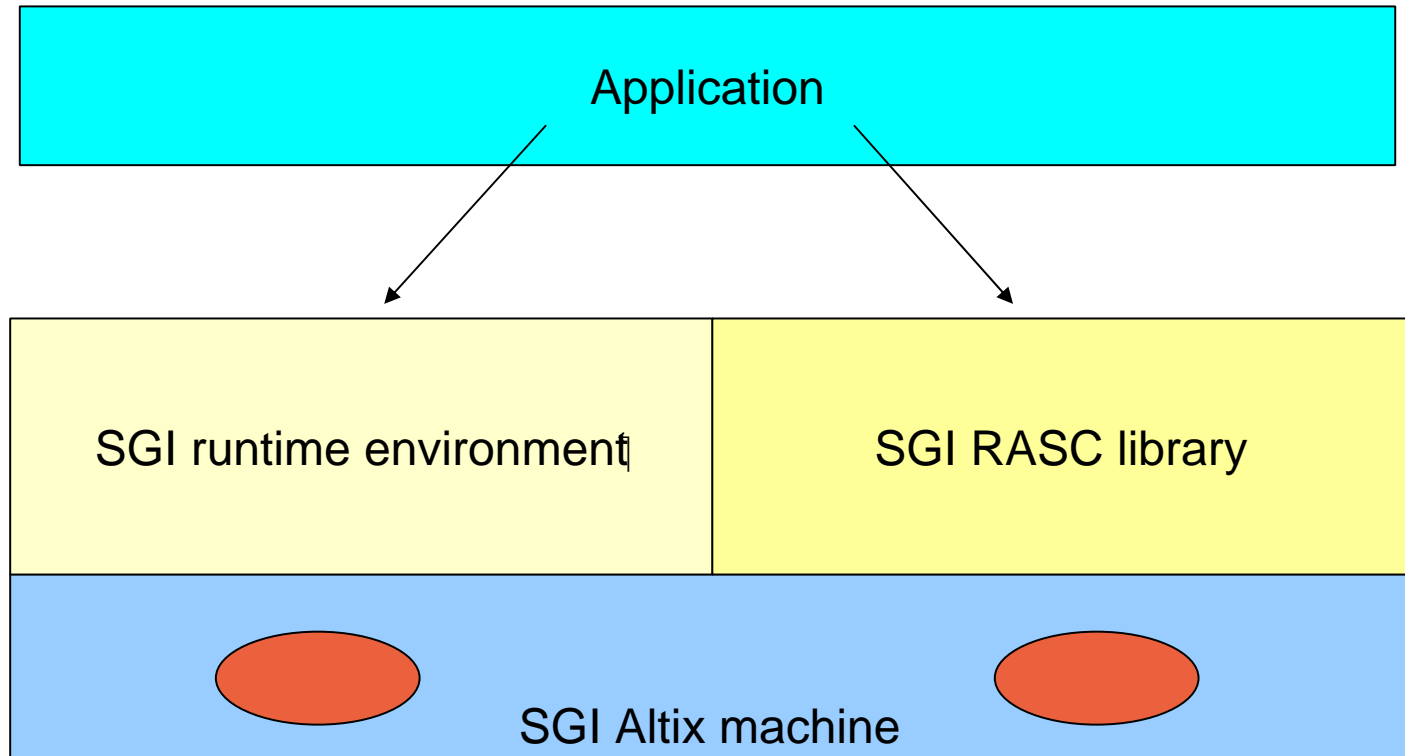
```
#pragma fpga task input(data_in, match_string) \  
  output (match_list, sort_output, pop_count) \  
  reset("multi_iter_rst", multi_iter_rst_on, multi_iter_rst_off) \  
  program_type("pattern_matching")  
  {  
  pattern_matching(data_in, match_list, sort_output, SIZE, match_string, &pop_count);  
  }
```

```
#pragma fpga taskwait
```

“pattern_matching” bitstream is taken from
an SGI RASC library example (alg_data_flow)

OpenMP

- Infrastructure



Q&A

- Thanks for your attention