Spiral: SMP Example

Franz Franchetti

$$\underbrace{AB}_{\text{par}(p)} \rightarrow \underbrace{A}_{\text{par}(p)} \underbrace{B}_{\text{par}(p)} \quad (1)$$

$$\underbrace{A_m \otimes \mathrm{I}_n}_{\text{par}(p)} \rightarrow \left(\mathrm{I}_p \otimes_{\|} (A_m \otimes \mathrm{I}_{n/p})\right)^{\left(\mathrm{L}_p^{mp} \otimes \mathrm{I}_{n/p}\right)} \quad (2)$$

$$\underbrace{\mathrm{I}_m \otimes A_n}_{\text{par}(p)} \rightarrow \mathrm{I}_p \otimes_{\|} \left(\mathrm{I}_{m/p} \otimes A_n\right) \quad (3)$$

$$\underbrace{D}_{\text{par}(p)} \rightarrow D, \ D \text{ diagonal} \quad (4)$$

$$\underbrace{P}_{\text{par}(p)} \rightarrow P, \ P \text{ permutation} \quad (5)$$

Table 1. Target architecture rules.

$$\underbrace{\mathrm{DFT}_{mn}}_{\mathrm{par}(p)} \rightarrow \underbrace{\left(\mathrm{DFT}_m \otimes \mathrm{I}_n\right)}_{\mathrm{par}(p)} \mathrm{T}_n^{mn} \underbrace{\left(\mathrm{I}_m \otimes \mathrm{DFT}_n\right)}_{\mathrm{par}(p)} \mathrm{L}_m^{nm}$$

$$\begin{aligned}
\underbrace{\left(\mathrm{DFT}_m \otimes \mathrm{I}_n\right)}_{\mathrm{par}(p)} &\rightarrow \left(\mathrm{L}_m^{mp} \otimes \mathrm{I}_{n/p}\right)\left(\mathrm{I}_p \otimes_{\parallel} (\mathrm{DFT}_m \otimes \mathrm{I}_{n/p})\right)\left(\mathrm{L}_p^{mp} \otimes \mathrm{I}_{n/p}\right) \\
&\rightarrow \mathrm{perm}\left(\ell_m^{mp} \otimes \imath_{n/p}\right) \left(\mathop{\textstyle\sum_{\parallel}}_{j=0}^{p-1} \mathrm{S}_{(j)_p \otimes \imath_{mn/p}} \left(\sum_{k=0}^{\frac{n}{p}-1} \mathrm{S}_{\imath_m \otimes (k)_{n/p}} \mathrm{DFT}_m \, \mathrm{G}_{\imath_m \otimes (k)_{n/p}} \right) \mathrm{G}_{(j)_p \otimes \imath_{mn/p}} \right) \mathrm{perm}\left(\ell_p^{mp} \otimes \imath_{n/p}\right) \\
&\rightarrow \mathop{\textstyle\sum_{\parallel}}_{j=0}^{p-1} \sum_{k=0}^{\frac{n}{p}-1} \mathrm{S}_{(\ell_p^{mp} \otimes \imath_{n/p}) \circ \left((j)_p \otimes \imath_m \otimes (k)_{n/p}\right)} \mathrm{DFT}_m \, \mathrm{G}_{(\ell_p^{mp} \otimes \imath_{n/p}) \circ \left((j)_p \otimes \imath_m \otimes (k)_{n/p}\right)} \\
&\rightarrow \mathop{\textstyle\sum_{\parallel}}_{j=0}^{p-1} \sum_{k=0}^{\frac{n}{p}-1} \mathrm{S}_{\imath_m \otimes (j)_p \otimes (k)_{n/p}} \mathrm{DFT}_m \, \mathrm{G}_{\imath_m \otimes (j)_p \otimes (k)_{n/p}}
\end{aligned}$$

$$\begin{aligned}
\underbrace{\left(\mathrm{DFT}_m \otimes \mathrm{I}_n\right)}_{\mathrm{par}(p)} \mathrm{T}_n^{mn} &\rightarrow \left(\mathop{\textstyle\sum_{\parallel}}_{j=0}^{p-1} \sum_{k=0}^{\frac{n}{p}-1} \mathrm{S}_{\imath_m \otimes (j)_p \otimes (k)_{n/p}} \mathrm{DFT}_m \, \mathrm{G}_{\imath_m \otimes (j)_p \otimes (k)_{n/p}} \right) \mathrm{T}_n^{mn} \\
&\rightarrow \mathop{\textstyle\sum_{\parallel}}_{j=0}^{p-1} \sum_{k=0}^{\frac{n}{p}-1} \mathrm{S}_{\imath_m \otimes (j)_p \otimes (k)_{n/p}} \mathrm{DFT}_m \, D_{j,k} \, \mathrm{G}_{\imath_m \otimes (j)_p \otimes (k)_{n/p}}
\end{aligned}$$

$$\begin{aligned}
\underbrace{\left(\mathrm{I}_m \otimes \mathrm{DFT}_n\right)}_{\mathrm{par}(p)} \mathrm{L}_m^{nm} &\rightarrow \mathrm{I}_p \otimes_{\parallel} \left(\mathrm{I}_{m/p} \otimes \mathrm{DFT}_n\right) \mathrm{L}_m^{mn} \\
&\rightarrow \left(\mathop{\textstyle\sum_{\parallel}}_{j=0}^{p-1} \mathrm{S}_{(j)_p \otimes \imath_{mn/p}} \left(\sum_{k=0}^{\frac{m}{p}-1} \mathrm{S}_{(k)_{m/p} \otimes \imath_n} \mathrm{DFT}_n \, \mathrm{G}_{(k)_{m/p} \otimes \imath_n} \right) \mathrm{G}_{(j)_p \otimes \imath_{mn/p}} \right) \mathrm{perm}\left(\ell_m^{mn}\right) \\
&\rightarrow \mathop{\textstyle\sum_{\parallel}}_{j=0}^{p-1} \sum_{k=0}^{\frac{m}{p}-1} \mathrm{S}_{(j)_p \otimes (k)_{m/p} \otimes \imath_n} \mathrm{DFT}_n \, \mathrm{G}_{\ell_m^{mn} \circ \left((j)_p \otimes (k)_{m/p} \otimes \imath_n\right)} \\
&\rightarrow \mathop{\textstyle\sum_{\parallel}}_{j=0}^{p-1} \sum_{k=0}^{\frac{m}{p}-1} \mathrm{S}_{(j)_p \otimes (k)_{m/p} \otimes \imath_n} \mathrm{DFT}_n \, \mathrm{G}_{\imath_n \otimes (j)_p \otimes (k)_{m/p}}
\end{aligned}$$

$$\underbrace{\mathrm{DFT}_{mn}}_{\mathrm{par}(p)} \rightarrow \mathop{\textstyle\sum_{\parallel}}_{j=0}^{p-1} \sum_{k=0}^{\frac{n}{p}-1} \mathrm{S}_{\imath_m \otimes (j)_p \otimes (k)_{n/p}} \mathrm{DFT}_m \, D_{j,k} \, \mathrm{G}_{\imath_m \otimes (j)_p \otimes (k)_{n/p}} \mathop{\textstyle\sum_{\parallel}}_{j=0}^{p-1} \sum_{k=0}^{\frac{m}{p}-1} \mathrm{S}_{(j)_p \otimes (k)_{m/p} \otimes \imath_n} \mathrm{DFT}_n \, \mathrm{G}_{\imath_n \otimes (j)_p \otimes (k)_{m/p}}$$

Table 2. Formal parallelization of DFT_{mn} for p processors using tSPL and Σ-SPL rewriting.

```
#include <omp.h>

static _Complex double D[8] = {
  1, 1, 1,
  0.70710678118654+__I__*0.70710678118654,
  1, 1, 1,
  -0.70710678118654+__I__*0.70710678118654
};

void DFT_8(_Complex double *Y,
           _Complex double *X) {
  int i1, i3;
  static _Complex double T[8];
  #pragma omp parallel for \
   schedule(static) shared(T, X, Y)
  for(i1 = 0; i1 <= 1; i1++) {
    _Complex double s1, s2;
    int i2;
    for(i2 = 0; i2 <= 1; i2++) {
      s1 = X[2*i1 + i2];
      s2 = X[4 + 2*i1 + i2];
      T[4*i1 + 2*i2] = s1 + s2;
      T[4*i1 + 2*i2 + 1] = s1 - s2;
    }
  }
  #pragma omp parallel for \
   schedule(static) shared(T, X, Y)
  for(i3 = 0; i3 <= 1; i3++) {
    _Complex double s3, s4, s5, s6,
      s7, s8, s9, s10;
    s10 = D[i3]*T[i3];
    s9 = D[4 + i3]*T[4 + i3];
    s8 = s10 + s9;
    s4 = D[6 + i3]*T[6 + i3];
    s7 = D[2 + i3]*T[2 + i3];
    s6 = s7 + s4;
    s5 = s10 - s9;
    s3 = __I__*(s7 - s4);
    Y[i3] = s8 + s6;
    Y[4 + i3] = s8 - s6;
    Y[2 + i3] = s5 + s3;
    Y[6 + i3] = s5 - s3;
  }
}
```

Figure 1. Multithreaded C99 OpenMP program for $y = \mathrm{DFT}_8\ x$ running on 2 processors. `DFT_8(y,x)` **is called by a sequential program.**

```
volatile long barrier_val[]={0,0};
__declspec(thread) int current_barrier = 0;

static __forceinline void barrier(int thread_id) {
  int b_id = current_barrier;
  current_barrier ^= 1;
  _InterlockedIncrement(&barrier_val[b_id]);
  if (!thread_id) {
    while (barrier_val[b_id] < 2);
      barrier_val[b_id] = 0;
  }
  else
    while (barrier_val[b_id]);
}

static _Complex double D[8] = {
  1, 1, 1,
  0.70710678118654+__I__*0.70710678118654,
  1, 1, 1,
  -0.70710678118654+__I__*0.70710678118654
};

void DFT_8_thread(_Complex double *Y,
                  _Complex double *X,
                  int thread_id) {
  static __declspec(align(32)) double T[8];
  int i1, i2, i3;
  barrier(thread_id);
  // for(i1 = 0; i1 <= 1; i1++) {
  i1 = thread_id; {
    for(i2 = 0; i2 <= 1; i2++) {
      _Complex double s1, s2;
        s1 = X[2*i1 + i2];
      s2 = X[4 + 2*i1 + i2];
      T[4*i1 + 2*i2] = s1 + s2;
      T[4*i1 + 2*i2 + 1] = s1 - s2;
    }
  }
  barrier(thread_id);
  // for(i3 = 0; i3 <= 1; i3++) {
  i3 = thread_id; {
    _Complex double s3, s4, s5, s6,
      s7, s8, s9, s10;
    s10 = D[i3]*T[i3];
    s9 = D[4 + i3]*T[4 + i3];
    s8 = s10 + s9;
    s4 = D[6 + i3]*T[6 + i3];
    s7 = D[2 + i3]*T[2 + i3];
    s6 = s7 + s4;
    s5 = s10 - s9;
    s3 = __I__*(s7 - s4);
    Y[i3] = s8 + s6;
    Y[4 + i3] = s8 - s6;
    Y[2 + i3] = s5 + s3;
    Y[6 + i3] = s5 - s3;
  }
  barrier(thread_id);
}
```

Figure 2. Multithreaded C99 program for $y = \mathrm{DFT}_8\, x$ running on 2 processors using lightweight synchronization. Two threads are already running and `DFT_8_thread(y,x,thread_id)` is called by both threads simultaneously.