ECE 565 homework 3

Js895 **Junqi Sun**

Problem1 :

For I loop only:

Read only:
N, data_array

Read/write non-conflicting:
Data_gridX, data_gridY, measurement

Read-write conflicting:
I, j, product, sum

For j loop only:

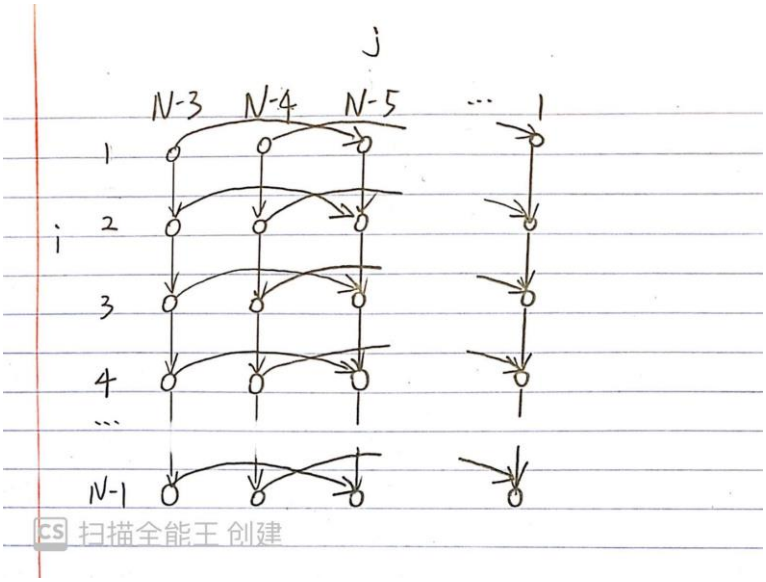
Read only:
I, N, sum

Read/write non-conflicting:
Data_gridX, data_gridY, measurement

Read-write conflicting:
j, product

Problem2 :

LDG:



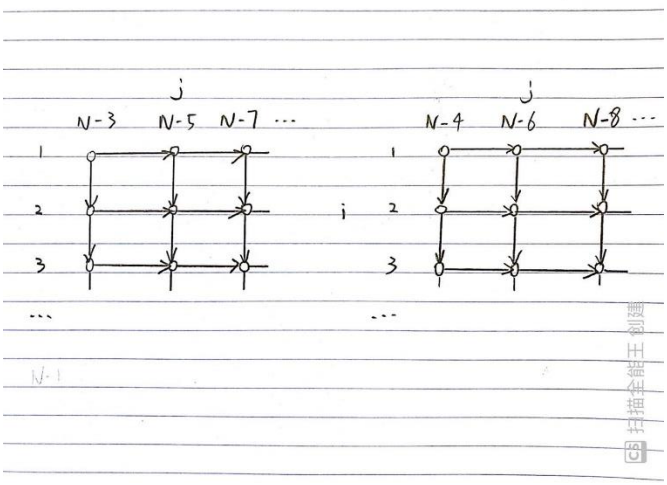
(a)

No, for I loop is not independent parallel task, according to the LDG above
(eg: dependence from 1 to 2)

(b)

No, for j loop is not independent parallel task, according to the LDG above
(eg: dependence from N-3 to N-5)

(c)



When I divide the LDG above into two, each of them could be optimized.
For diagonal traverse, it's not independent parallel task, since there are still dependences that

could not ignore if we traverse nodes by diagonals.

For anti-diagonal traverse, it is independent parallel task, since each nodes on anti-diagonals are independent to each other, there's no other dependence remains when we execute the nodes on current anti-diagonals.

(d)

Except for parallel tasks above, it can also be DOPIPE parallelism.

Since we have Loop-independent dependence:
S1[i][j] -> A S2[i][j]

And loop-carried dependence:
S1[i][j] -> T S1[i][j-2]
S2[i][j] -> T S2[i+1][j]
S2[i][j] -> A S2[i+1][j]

So we could make S2 execute after S1.

```
For(i=1; i<N; i++) {
    For(j=N-3; j>0; j--) {
        S1;
        Post(j);
    }
}
For(i=1; i<N; i++) {
    For(j=N-3; j>0; j--) {
        Wait(j);
        S2;
    }
}
```

Problem3 :

(a)

With -fno-inline:

```
File Edit Options Buffers Tools Text Help
Flat profile:

Each sample counts as 0.01 seconds.
  %   cumulative   self              self     total
 time   seconds   seconds    calls   s/call   s/call  name
 32.57      3.97     3.97      201     0.02     0.02  miniFE::matvec_std<miniFE::CSRMatrix<double, int, int>, miniFE::Vector<double, i\
nt, int> >::operator()(miniFE::CSRMatrix<double, int, int>&, miniFE::Vector<double, int, int>&, miniFE::Vector<double, int, int>&)
 13.54      5.62     1.65 1622833938     0.00     0.00  frame_dummy
  5.66      6.31     0.69 57598102     0.00     0.00  std::_Rb_tree<int, int, std::_Identity<int>, std::less<int>, std::allocator<int>\
 >::_S_key(std::_Rb_tree_node<int> const*)
  4.43      6.85     0.54 435792686     0.00     0.00  std::_Rb_tree<int, int, std::_Identity<int>, std::less<int>, std::allocator<int\
> >::_S_value(std::_Rb_tree_node<int> const*)
  4.10      7.35     0.50 435928532     0.00     0.00  std::_Rb_tree_node<int>::_M_valptr()
  3.94      7.83     0.48   512000     0.00     0.00  void miniFE::Hex8::diffusionMatrix_symm<double>(double const*, double const*, do\
uble*)
  3.69      8.28     0.45 510695430     0.00     0.00  std::pair<int const, int>* std::__addressof<std::pair<int const, int> >(std::pa\
ir<int const, int>&)
  2.63      8.60     0.32 32768000     0.00     0.00  int* std::lower_bound<int*, unsigned long>(int*, int*, unsigned long const&)
```

Function name:

Function name	Num of calls	% of exe time for all call
miniFE::matvec_std<miniFE::CSRMatrix<double, int, int>, miniFE::Vector<double, i\ nt, int> >::operator()(miniFE::CSRMatrix<double, int, int>&, miniFE::Vector<double, int, int>&, miniFE::Vector<double, int, int>&)	201	32.57
frame_dummy	1622833938	13.54
std::_Rb_tree<int, int, std::_Identity<int>, std::less<int>, std::allocator<int>\ >::_S_key(std::_Rb_tree_node<int> const*)	57598102	5.66
std::_Rb_tree<int, int, std::_Identity<int>, std::less<int>, std::allocator<int\ > >::_S_value(std::_Rb_tree_node<int> const*)	435792686	4.43
std::_Rb_tree_node<int>::_M_valptr()	435928532	4.10
void miniFE::Hex8::diffusionMatrix_symm<double>(double const*, double const*, do\ uble*)	512000	3.94
std::pair<int const, int>* std::__addressof<std::pair<int const, int> >(std::pa\ ir<int const, int>&)	510695430	3.69
int* std::lower_bound<int*, unsigned long>(int*, int*, unsigned long const&)	32768000	2.63

Without fno-inline

```
Flat profile:

Each sample counts as 0.01 seconds.
  %   cumulative   self              self     total
 time   seconds   seconds    calls   s/call   s/call  name
 64.45      4.55     4.55        1     4.55     4.63  void miniFE::cg_solve<miniFE::CSRMatrix<double, int, int>, miniFE::Vector<double\
, int, int>, miniFE::matvec_std<miniFE::CSRMatrix<double, int, int>, miniFE::Vector<double, int, int> > >(miniFE::CSRMatrix<double, in\
t, int>&, miniFE::Vector<double, int, int> const&, miniFE::Vector<double, int, int>&, miniFE::matvec_std<miniFE::CSRMatrix<double, int\
, int>, miniFE::Vector<double, int, int> >, miniFE::CSRMatrix<double, int, int>::LocalOrdinalType, miniFE::TypeTraits<miniFE::CSRMatri\
x<double, int, int>::ScalarType>::magnitude_type&, miniFE::CSRMatrix<double, int, int>::LocalOrdinalType&, miniFE::TypeTraits<miniFE::\
CSRMatrix<double, int, int>::ScalarType>::magnitude_type&, double*)
  7.93      5.11     0.56   512000     0.00     0.00  void miniFE::Hex8::diffusionMatrix_symm<double>(double const*, double const*, do\
uble*)
  7.65      5.65     0.54        2     0.27     0.27  void miniFE::impose_dirichlet<miniFE::CSRMatrix<double, int, int>, miniFE::Vecto\
r<double, int, int> >(miniFE::CSRMatrix<double, int, int>::ScalarType, miniFE::CSRMatrix<double, int, int>&, miniFE::Vector<double, in\
t, int>&, int, int, int, std::set<miniFE::CSRMatrix<double, int, int>::GlobalOrdinalType, std::less<miniFE::CSRMatrix<double, int, int\
>::GlobalOrdinalType>, std::allocator<miniFE::CSRMatrix<double, int, int>::GlobalOrdinalType> > const&)
  5.10      6.01     0.36   512000     0.00     0.00  void miniFE::sum_in_symm_elem_matrix<miniFE::CSRMatrix<double, int, int> >(unsig\
ned long, miniFE::CSRMatrix<double, int, int>::GlobalOrdinalType const*, miniFE::CSRMatrix<double, int, int>::ScalarType const*, miniF\
E::CSRMatrix<double, int, int>&)
  2.97      6.22     0.21  4096000     0.00     0.00  void miniFE::Hex8::gradients_and_invJ_and_detJ<double>(double const*, double con\
st*, double*, double&)
  2.55      6.40     0.18  4096000     0.00     0.00  void miniFE::Hex8::gradients_and_detJ<double>(double const*, double const*, doub\
le&)
  2.12      6.55     0.15        1     0.15     0.30  int miniFE::generate_matrix_structure<miniFE::CSRMatrix<double, int, int> >(mini\
FE::simple_mesh_description<miniFE::CSRMatrix<double, int, int>::GlobalOrdinalType> const&, miniFE::CSRMatrix<double, int, int>&)
  1.56      6.66     0.11   512000     0.00     0.00  void miniFE::Hex8::sourceVector<double>(double const*, double const*, double*)
```

Function name	Num of calls	% of exe time for all call
void miniFE::cg_solve<miniFE::CSRMatrix<double, int, int>, miniFE::Vector<double\ , int, int>, miniFE::matvec_std<miniFE::CSRMatrix<double, int, int>, miniFE::Vector<double, int, int> > >(miniFE::CSRMatrix<double, in\ t, int>&, miniFE::Vector<double, int, int> const&, miniFE::Vector<double, int, int>&, miniFE::matvec_std<miniFE::CSRMatrix<double, int\ , int>, miniFE::Vector<double, int, int> >, miniFE::CSRMatrix<double, int, int>::LocalOrdinalType, miniFE::TypeTraits<miniFE::CSRMatri\ x<double, int, int>::ScalarType>::magnitude_type&, miniFE::CSRMatrix<double, int, int>::LocalOrdinalType&, miniFE::TypeTraits<miniFE::\ CSRMatrix<double, int, int>::ScalarType>::magnitude_type&, double*)	1	64.45
void miniFE::Hex8::diffusionMatrix_symm<double>(double const*, double const*, do\ uble*)	512000	7.93
void miniFE::impose_dirichlet<miniFE::CSRMatrix<double, int, int>, miniFE::Vecto\ r<double, int, int> >(miniFE::CSRMatrix<double, int, int>::ScalarType, miniFE::CSRMatrix<double, int, int>&, miniFE::Vector<double, in\ t, int>&, int, int, int, std::set<miniFE::CSRMatrix<double, int, int>::GlobalOrdinalType, std::less<miniFE::CSRMatrix<double, int, int\	2	7.65

>::GlobalOrdinalType>, std::allocator<miniFE::CSRMatrix<double, int, int>::GlobalOrdinalType> > const&)		
void miniFE::sum_in_symm_elem_matrix<miniFE::CSRMatrix<double, int, int> >(unsig\ned long, miniFE::CSRMatrix<double, int, int>::GlobalOrdinalType const*, miniFE::CSRMatrix<double, int, int>::ScalarType const*, miniF\E::CSRMatrix<double, int, int>&)	512000	5.10
void miniFE::Hex8::gradients_and_invJ_and_detJ<double>(dou ble const*, double con\st*, double*, double&)	4096000	2.97
void miniFE::Hex8::gradients_and_detJ<double>(double const*, double const*, doub\le&)	4096000	2.55
int miniFE::generate_matrix_structure<miniFE::CSRMatrix<do uble, int, int> >(mini\FE::simple_mesh_description<miniFE::CSRMatrix<double, int, int>::GlobalOrdinalType> const&, miniFE::CSRMatrix<double, int, int>&)	1	2.12
void miniFE::Hex8::sourceVector<double>(double const*, double const*, double*)	512000	1.56

(b)

For fno-inline version
Speedup = 1/[(1-P) + P/N]
We have p = 32.57%, N = 5
So speedup = 1/[(1-32.57%)+%32.57/5]
= 1.352

For non in-line version
P = 64.45%, N=5
So speedup = 1/[(1-64.45%)+%64.45/5]
= 2.064

(c)

When executing perf stat -e NameOfEvents ./miniFE.x -nx 40 -ny 80 -nz 160:

Events name	counts
Instructions	39,728,420,823
cpu-cycles	18,034,225,414
branch-instructions	6,074,937,474
branch-misses	12,113,197
cache-references	338,735,911
L1-dcache-load-misses	739,576,192
L1-icache-load-misses	659,103
LLC-loads	643,154,678
LLC-load-misses	344,891,752
dTLB-load-misses	12,124,448

Problem4 :

Inner Loop Type	I-J-K	J-K-I	I-K-J
Running time (seconds) (performance)	14.720	27.319	1.101
cache-misses	33,772,230	17,095,562	470,552
L1-dcache-load-misses	2,161,257,167	3,238,061,406	139,676,147
L1-icache-load-misses	439,820	370,250	85,686
LLC-load-misses	7,728,561	4,354,056	125,988

According to different inner loop type, I-J-K should have 1.125 misses per iteration, J-K-I should have 2 misses per iteration, I-K-J should 0.25 misses per iteration. So I-K-J should have shortest running time, I-J-K should have second shortest running time and J-K-I should have longest running time. Which is exactly as experimental data shown(performance row).

The misses of cache data also prove our expectation, (eg: 470,552 is roughly 1/8 of 33,772,230). Data for L1 Dcache and LLC also obey the theory from numerical relationship of each inner loop.