⑬ Feedback-Directed Selection Techniques of Compiler Directives

Kiyomi WADA[1,2], Makoto SATOH[1,2], Keiko MOTOKAWA[1,2]
[1] APC Technology Group [2] Hitachi, Ltd.

Table of Contents
1

 HITACHI Inspire the Next

Motive and Purpose
2

- Motive

The maximum performance of a program does not necessarily come out just applying compile-time optimizations, because there are a lot of alternatives such as

- The combination of optimizations and their application order,
- The parameters of each optimization.

For obtaining the optimal code, we must try a large number of those combinations.

- Purpose

To develop a tool that enables users to select the number of trials and the performance of their codes
(though there exists a trade-off between them).

 HITACHI Inspire the Next

Conventional Technology(1)
3

Developer (Tool name)	Target to Opt. (Options/Directives)	Selection technique	Problem
Purdue Univ. (ADAPT)	Whole Program (Compile Options)	Linear Search	Interaction among options is not considered well.l
	Individual loop (Directives)	Linear Search to Each Loop	The best combination of directives cannot be found to the whole nested loop.
Intel	Whole Program (Compile Options)	Fractional Factorial Design	The best optimization for each loop nest cannot be selected.

 HITACHI Inspire the Next

Conventional Technology(2)
4

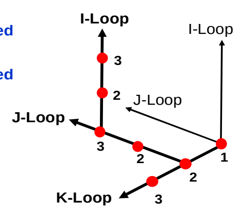
- Linear Search -

- To determine the best parameters of multiple directives, linear search method changes the parameter of one directive at a time.
- The interaction among directives is ignored.

step1	step2	step3
*UNR(1..3)	*UNR(2)←Decided	*UNR(2) ←Decided
do K=1, N	do K=1, N	do K=1, N
*UNR(1)	*UNR(2..3)	*UNR(3) ←Decided
do J=1, N	do J=1, N	do J=1, N
*UNR(1)	*UNR(1)	*UNR(2..3)
do I=1, N	do I=1, N	do I=1, N

Selection process

UNR : Unroll



 HITACHI Inspire the Next

Conventional Technology(3)
5

- Fractional Factorial Design -

Definition: (Interaction)

Program P, Performance 1.0 —Trans A→ P_A 1.2 → P_{A*B} 1.44?
Program P, Performance 1.0 —Trans B→ P_B 1.2 → P_{A*B}

P_{A*B}
1.44 No
1.6 Synergistic
0.8 Antagonistic
} Interaction between A and B

- Higher grades of interaction is opt to provide smaller influence to the performance.

Main effect ：A, B, C
Interaction of 2 factors ：A*B, B*C, …
Interaction of 3 factors ：A*B*C

- So, we ignore the Interaction of lots of factors → The number of trials decreases

 HITACHI Inspire the Next

Target of This Research
6

Developer (Tool name)	Target to Opt. (Options/Directives)	Selection technique	Problem
Purdue Univ. (ADAPT)	Whole Program (Compile Options)	Linear Search	Interaction among options is not considered well
	Individual loop (Directives)	Linear Search to Each Loop	The best combination of directives cannot be found to the whole nested loop
Intel	Whole Program (Compile Options)	Fractional Factorial Design	The best optimization for each loop nest cannot be selected.
APC	Whole Nested Loop (Directives)	Fractional Factorial Design	

 HITACHI Inspire the Next

Methods(1)
7

1. Reduction of the combination of optimizations
Fractional Factorial Design is applied.

2. Reduction of the combination of loops and optimizations
On one trial, the same optimization is applied to all nested loops and the execution time of each nested loop is measured.
After all trials, the best optimization can be selected for each nested loop.

3. Reduction of the number of target loops
Profile information is used.

4. Four effective optimizations
We limit the loop optimizations to the following :
- Parallelization, Interchange, Tiling, and Unrolling

 HITACHI Inspire the Next

Methods(2)
8

2. Reduction of the combination of loops and optimizations

Source Program
```
    DO 10 I=1,M
    ...
10 CONTINUE
    DO 20 J=1,N
    ...
20 CONTINUE
```

insert UNR(1) to all loops
```
  T1=CLOCK
*UNR(1)
    DO 10 I=1,M
    ...
10 CONTINUE
  T1=CLOCK-T1
  T2=CLOCK
*UNR(1)
    DO 20 J=1,N
    ...
20 CONTINUE
  T2=CLOCK-T1
```

・・・

insert UNR(4) to all loops
```
  T1=CLOCK
*UNR(4)
    DO 10 I=1,M
    ...
10 CONTINUE
  T1=CLOCK-T1
  T2=CLOCK
*UNR(4)
    DO 20 J=1,N
    ...
20 CONTINUE
  T2=CLOCK-T1
```

Measure the exec.time of each loop nest

Optimized Program
```
*UNR(2)
    DO 10 I=1,M
    ...
10 CONTINUE
*UNR(3)
    DO 20 J=1,N
    ...
20 CONTINUE
```

 HITACHI Inspire the Next

Methods(3)
9

4. Four effective optimizations
- Interchange / Tiling / Parallelization generates high-speed OpenMP program
- Unrolling is applied to the above result to obtain higher-speed OpenMP program

Source Program
```
DO 20 K=1, N
  DO 20 I=1, N
    DO 20 J=1, M
      ・・・
```

1st phase →

Tilling is applied.
```
*TILE(40)
  DO 20 K=1, N
*TILE(40)
    DO 20 I=1, N
      DO 20 J=1, M
        ・・・
```

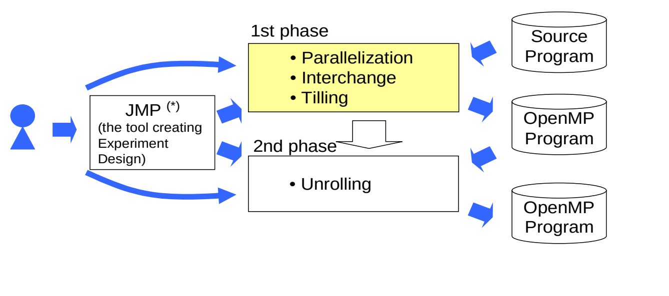
OpenMP after Tilling
```
!$OMP PARALLEL DO
  DO 20 KK=1, N, 40
    DO 20 II=1, N, 40
      DO 20 K=KK, KK+39
        DO 20 I=II, II+39
          DO 20 J=1, M
            ・・・
```

2nd phase →

Unrolling is applied.
```
!$OMP PARALLEL DO
  DO 20 KK=1, N, 40
    DO 20 II=1, N, 40
*UNR(4)
      DO 20 K=KK+1, 39
*UNR(4)
        DO 20 I=II+1, 39
          DO 20 J=1, M
            ・・・
```

 HITACHI Inspire the Next

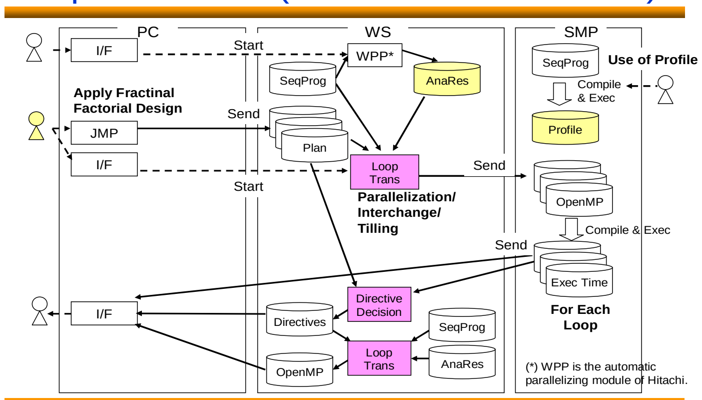
Implementation(Structure)
10



(*) JMP(TM) is the trademark of SAS Institute Inc. in U.S.

 HITACHI Inspire the Next

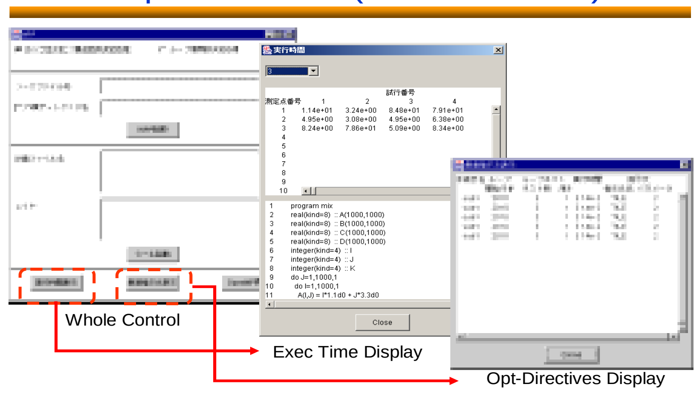
Implementation(Structure of 1st Phase)
11



(*) WPP is the automatic parallelizing module of Hitachi.

 HITACHI Inspire the Next

Implementation(User Interface)
12



 HITACHI Inspire the Next

Preliminary Evaluation(SPEC® fp95/mgrid)
13

- We tune up two 3-nested loops which occupy 40% and 20% of the total execution time.
- We look for the optimal unrolling directives.

OpenMP Program
```
!$OMP PARALLEL
UNR1
  do K=2, N-1
UNR2
    do J=2, N-1
UNR3
      do I=2, N-1
        ・・・
```

Input to JMP

Dir-Name	Unroll-Param		
UNR1	1	2	3
UNR2	1	2	3
UNR3	1	2	3

Fractioal Factorial Design (19Trials)

Trial#	UNR1	UNR2	UNR3
1	1	1	3
2	3	3	3
3	3	2	2
・・・		・・・	
18	2	1	1
19	3	1	3

Full Factorial Design (27Trials)

Trial#	UNR1	UNR2	UNR3
1	1	1	1
2	2	1	1
3	3	1	1
・・・		・・・	
26	2	3	3
27	3	3	3

 HITACHI Inspire the Next

Preliminary Evaluation (Result)
14

Perform Parallel Execution on SR8000(*)

Method (kinds)	Time[Sec]
No Unrolling	17.7
Fractional Factorial Design (19)	11.9
Full Factorial Design (27)	11.3

Opt. Program
```
do K=2, N-1, 2
  do J=2, N-1
    do I=2, N-1, 2
      ・・・
```

Fractional Factorial Design

Rank	Trtal	UNR1	UNR2	UNR3	Time[Sec]
1	12	2	1	2	11.9
2	1	1	1	3	12.3
3	13	1	2	3	12.3
4	18	2	1	1	12.7
5	4	3	2	1	12.8
6	17	3	3	1	12.9
7	5	1	2	2	13.0
8	8	1	3	1	13.3
9	11	3	1	1	13.5
10	14	2	2	1	13.6
11	9	1	1	2	13.9
12	19	3	1	3	14.2
13	6	3	3	2	15.2
14	3	3	2	2	15.3
15	15	2	3	2	15.4
16	10	2	2	3	16.0
17	16	2	3	3	16.8
18	7	1	3	3	16.9
19	2	3	3	3	18.1

(*) SR8000 is the Hitachi SMP machine whose node consisnts of 8PEs.(12GFlops/PE)

 HITACHI Inspire the Next

Conclusion
15

- We have developed a tool that enables users to select the number of trials and the performance of their codes.

That tool
- Reduces the combination of optimizations
- Reduces the combination of loops and optimizations,
- Reduces the number of target loops by profile, and
- applies four effective optimizations.

- We applied unrolling to two 3-nested loops in SPEC95/mgrid.
  - The near-optimal code was obtained in smaller number of combinations using Fractional Factorial Design.

 HITACHI Inspire the Next