


FUJITSU Software

Technical Computing Suite V4.0L20

A horizontal decorative band with a dark red background. It features several overlapping, glowing red lines and curves that create a sense of motion and depth, with bright white highlights at the intersections.

Development Studio
C/C++ Compiler
Optimization Messages

J2UL-2478-02ENZ0(03)
March 2022

Preface

Purpose of This Manual

This manual describes the C and C++ compiler optimization messages output.

Intended Readers

This manual is intended for the users who develop C and C++ programs by this system.

Structure of This Manual

The structure of this manual is as follows:

[Chapter 1 Format of Optimization Messages](#)

This chapter describes the format of the optimization messages output.

[Chapter 2 Optimization Messages](#)

This chapter describes the optimization messages output.

Export Controls

Exportation/release of this document may require necessary procedures in accordance with the regulations of your resident country and/or US export control laws.

Trademarks

All trademarks are the property of their respective owners.

Date of Publication and Version

Version	Manual code
March 2022, Version 2.3	J2UL-2478-02ENZ0(03)
January 2021, Version 2.2	J2UL-2478-02ENZ0(02)
June 2020, Version 2.1	J2UL-2478-02ENZ0(01)
March 2020, 2nd Version	J2UL-2478-02ENZ0(00)
January 2020, 1st Version	J2UL-2478-01ENZ0(00)

Copyright

Copyright FUJITSU LIMITED 2020-2022

Update History

Changes	Location	Version
Added a note.	Chapter 1	Version 2.3
Message added. - jwd5019	Chapter 2	
Message added. - jwd8325	Chapter 2	Version 2.2
Messages changed. - jwd5202, jwd5228, jwd6202, jwd6228, jwd8218	Chapter 2	Version 2.1
loop-identification-number added to the format.	Chapter 1	2nd Version

Changes	Location	Version
Messages added. - jwd8216, jwd8217, jwd8218, jwd8219, jwd8223, jwd8224, jwd8642, jwd8643, jwd8674	Chapter 2	
Improved the explanation.	-	
Changed the look according to product upgrades.	-	

All rights reserved.
 The information in this manual is subject to change without notice.

Contents

Chapter 1 Format of Optimization Messages.....	1
Chapter 2 Optimization Messages.....	3
2.1 jwd5001p-i.....	3
2.2 jwd5002p-i.....	3
2.3 jwd5004p-i.....	3
2.4 jwd5005p-i.....	3
2.5 jwd5006p-i.....	4
2.6 jwd5007p-i.....	4
2.7 jwd5009p-i.....	4
2.8 jwd5011p-i.....	5
2.9 jwd5012p-i.....	5
2.10 jwd5013p-i.....	5
2.11 jwd5014p-i.....	5
2.12 jwd5015p-i.....	5
2.13 jwd5016p-i.....	6
2.14 jwd5018p-i.....	6
2.15 jwd5019p-i.....	6
2.16 jwd5101p-i.....	6
2.17 jwd5122p-i.....	7
2.18 jwd5123p-i.....	7
2.19 jwd5125p-i.....	7
2.20 jwd5131p-i.....	8
2.21 jwd5133p-i.....	8
2.22 jwd5134p-i.....	8
2.23 jwd5136p-i.....	8
2.24 jwd5137p-i.....	8
2.25 jwd5138p-i.....	9
2.26 jwd5141p-i.....	9
2.27 jwd5142p-i.....	9
2.28 jwd5143p-i.....	9
2.29 jwd5156p-i.....	9
2.30 jwd5201p-i.....	10
2.31 jwd5202p-i.....	10
2.32 jwd5203p-i.....	10
2.33 jwd5204p-i.....	10
2.34 jwd5205p-i.....	11
2.35 jwd5206p-i.....	11
2.36 jwd5207p-i.....	11
2.37 jwd5208p-i.....	12
2.38 jwd5209p-i.....	12
2.39 jwd5210p-i.....	12
2.40 jwd5211p-i.....	12
2.41 jwd5221p-i.....	13
2.42 jwd5222p-i.....	13
2.43 jwd5223p-i.....	13
2.44 jwd5228p-i.....	14
2.45 jwd5302p-i.....	14
2.46 jwd5401i-i.....	14
2.47 jwd5402i-i.....	14
2.48 jwd5403i-i.....	15
2.49 jwd5404i-i.....	15
2.50 jwd6001s-i.....	15
2.51 jwd6002s-i.....	15
2.52 jwd6004s-i.....	16

2.53 jwd6005s-i.....	16
2.54 jwd6007s-i.....	16
2.55 jwd6016s-i.....	17
2.56 jwd6025s-i.....	17
2.57 jwd6101s-i.....	17
2.58 jwd6122s-i.....	17
2.59 jwd6123s-i.....	17
2.60 jwd6125s-i.....	18
2.61 jwd6131s-i.....	18
2.62 jwd6133s-i.....	18
2.63 jwd6134s-i.....	18
2.64 jwd6137s-i.....	18
2.65 jwd6138s-i.....	19
2.66 jwd6141s-i.....	19
2.67 jwd6142s-i.....	19
2.68 jwd6143s-i.....	19
2.69 jwd6144s-i.....	20
2.70 jwd6145s-i.....	20
2.71 jwd6146s-i.....	20
2.72 jwd6156s-i.....	20
2.73 jwd6201s-i.....	21
2.74 jwd6202s-i.....	21
2.75 jwd6203s-i.....	21
2.76 jwd6204s-i.....	21
2.77 jwd6205s-i.....	22
2.78 jwd6206s-i.....	22
2.79 jwd6207s-i.....	22
2.80 jwd6208s-i.....	23
2.81 jwd6209s-i.....	23
2.82 jwd6210s-i.....	23
2.83 jwd6211s-i.....	23
2.84 jwd6221s-i.....	24
2.85 jwd6222s-i.....	24
2.86 jwd6223s-i.....	24
2.87 jwd6228s-i.....	25
2.88 jwd6229s-i.....	25
2.89 jwd6302s-i.....	25
2.90 jwd6402s-i.....	25
2.91 jwd6501s-i.....	25
2.92 jwd6502s-i.....	26
2.93 jwd6504s-i.....	26
2.94 jwd6505s-i.....	26
2.95 jwd6506s-i.....	26
2.96 jwd6510s-i.....	27
2.97 jwd6512s-i.....	27
2.98 jwd6513s-i.....	27
2.99 jwd8101o-i.....	27
2.100 jwd8102o-i.....	28
2.101 jwd8201o-i.....	28
2.102 jwd8202o-i.....	28
2.103 jwd8203o-i.....	28
2.104 jwd8204o-i.....	29
2.105 jwd8205o-i.....	29
2.106 jwd8206o-i.....	29
2.107 jwd8207o-i.....	29
2.108 jwd8208o-i.....	30
2.109 jwd8209o-i.....	30

2.110 jwd8210o-i.....	30
2.111 jwd8211o-i.....	31
2.112 jwd8212o-i.....	31
2.113 jwd8213o-i.....	31
2.114 jwd8214o-i.....	31
2.115 jwd8215o-i.....	32
2.116 jwd8216o-i.....	32
2.117 jwd8217o-i.....	32
2.118 jwd8218o-i.....	32
2.119 jwd8219o-i.....	33
2.120 jwd8220o-i.....	33
2.121 jwd8222o-i.....	33
2.122 jwd8223o-i.....	33
2.123 jwd8224o-i.....	34
2.124 jwd8230o-i.....	34
2.125 jwd8231o-i.....	34
2.126 jwd8300o-i.....	35
2.127 jwd8310o-i.....	35
2.128 jwd8311o-i.....	35
2.129 jwd8312o-i.....	35
2.130 jwd8320o-i.....	35
2.131 jwd8321o-i.....	36
2.132 jwd8322o-i.....	36
2.133 jwd8324o-i.....	36
2.134 jwd8325o-i.....	36
2.135 jwd8330o-i.....	37
2.136 jwd8331o-i.....	37
2.137 jwd8400o-i.....	37
2.138 jwd8450i-w.....	37
2.139 jwd8451i-w.....	38
2.140 jwd8452i-w.....	38
2.141 jwd8642i-u.....	38
2.142 jwd8643i-u.....	38
2.143 jwd8660i-i.....	39
2.144 jwd8662o-i.....	39
2.145 jwd8663o-i.....	39
2.146 jwd8664o-i.....	40
2.147 jwd8665o-i.....	40
2.148 jwd8666o-i.....	40
2.149 jwd8667o-i.....	40
2.150 jwd8668o-i.....	41
2.151 jwd8669o-i.....	41
2.152 jwd8670o-i.....	41
2.153 jwd8671o-i.....	42
2.154 jwd8672o-i.....	42
2.155 jwd8673o-i.....	42
2.156 jwd8674o-i.....	42
2.157 jwd8695i-u.....	43
2.158 jwd8701o-i.....	43

Chapter 1 Format of Optimization Messages

The format of optimization messages is as following:

```
jwdxxxxy-z filename line-number column-number loop-identification-number Message-text
```

jwdxxxxy-z

jwd

Compiler messages prefix.

xxxx

Message serial number.

y

Message type ("i", "o", "s", or "p").

<i>y</i>	Description
i	This message is unrelated to the optimization.
o	This message reports the optimization status.
s	This message reports the optimization of using SIMD instructions.
p	This message reports the optimization status of automatic parallelization.

z

Message error level ("i", "w", "s", or "u") and return code

<i>z</i>	Return Code	Description
i	0	The message means information.
w		The message means a warning. The system continues processing.
s	1	The message means a serious error. The system ignores the erroneous statement and continues processing other statements.
u		The message means an unrecoverable error. The system terminates processing.

filename

The filename where the message event was detected.

line-number

The line number where the message event was detected.



Note

Due to the influence of optimizations, incorrect line numbers may output.

column-number

The column number of the line where the message event was detected. The column number may be added to diagnostic messages.

loop-identification-number

The loop identification number of the loop generated internally by the compiler optimizations, such as loop fission.

 Note

The loop identification number does not guarantee the execution order of loops.

Message-text

The message text which is issued in either English or Japanese depending on user environment variables, such as LC_MESSAGES and LANG.

Chapter 2 Optimization Messages

2.1 jwd5001p-i

This loop with loop variable '*parm*' is parallelized.

[Message Explanation]

The loop is parallelized.

[Parameters Explanation]

parm : Loop variable of the loop automatically parallelized

[System Behavior]

Performs automatic parallelization to the loop.

2.2 jwd5002p-i

This loop is parallelized.

[Message Explanation]

The loop is parallelized.

[System Behavior]

Performs automatic parallelization to the loop.

2.3 jwd5004p-i

This loop with loop variable '*parm*' containing reduction operation is parallelized.

[Message Explanation]

This loop is parallelized by changing the evaluation order of the expression which contains commutative operation such as addition or multiplication according to the commutative law.

When this message is output, side effects (calculation errors or runtime exceptions) may occur in the execution results.

[Parameters Explanation]

parm : Loop variable of the automatically parallelized loop

[System Behavior]

Performs automatic parallelization to the loop.

[Programmer Response]

When it seems that a side effect by optimization in the execution result, re-compile without the `-Kreduction` and/or `-Keval` options, then re-run the job and verify the result.

2.4 jwd5005p-i

This loop with loop variable '*parm*' is partially parallelized.

[Message Explanation]

The loop is partially parallelized.

[Parameters Explanation]

parm : Loop variable of the loop partially parallelized

[System Behavior]

Performs partially automatic parallelization to the loop.

2.5 jwd5006p-i

Because the iteration count of parallelized loop is *num*, parallel speedup ratio of *num2* or more is not achieved.

[Message Explanation]

When a program is executed with more threads than the iteration count of parallelized loop, parallel speedup ratio in accordance with the number of threads is not achieved.

[Parameters Explanation]

num : Iteration count of a loop

num2 : One added to iteration count of a loop

[System Behavior]

Performs automatic parallelization to the loop.

2.6 jwd5007p-i

This loop containing reduction operation is parallelized.

[Message Explanation]

The evaluation order of the expression which contains commutative operation such as addition or multiplication was changed according to the commutative law.

When this message is output, side effects (calculation errors or runtime exceptions) may occur in the execution results.

[System Behavior]

Performs automatic parallelization to the loop.

[Programmer Response]

When it seems that a side effect by optimization in the execution result, re-compile without the `-Kreduction` and/or `-Keval` options, then re-run the job and verify the result.

2.7 jwd5009p-i

When nested loop with loop variable '*parm*' is executed, the loop to obtain the parallel performance is selected.

[Message Explanation]

The loop of automatic parallel execution in a nested loop is determined dynamically at execution depending on the iteration count of the loop in order to obtain the parallel performance.

[Parameters Explanation]

parm : Loop variable of the nested loop

[System Behavior]

The loop of execution in parallel is determined at the execution time according to the execution environment.

2.8 jwd5011p-i

When nested loop is executed, the loop to obtain the parallel performance is selected.

[Message Explanation]

The loop of automatic parallel execution in a nested loop is determined dynamically at execution depending on the iteration count of the loop in order to obtain the parallel performance.

[System Behavior]

The loop of execution in parallel is determined at the execution time according to the execution environment.

2.9 jwd5012p-i

This loop with loop variable '*parm*' is pipeline parallelized.

[Message Explanation]

The loop is pipeline parallelized.

[Parameters Explanation]

parm: Loop variable of the loop that applied pipeline parallelization

[System Behavior]

Applies pipeline parallelization.

2.10 jwd5013p-i

The temp_private specifier in Optimization Control Line (OCL) is applied to variable '*parm*'.

[Message Explanation]

The temp_private specifier in Optimization Control Line (OCL) is applied to variable.

[Parameters Explanation]

parm: Variable to which the temp_private specifier in Optimization Control Line (OCL) is applied

[System Behavior]

Applies the temp_private specifier in Optimization Control Line (OCL).

2.11 jwd5014p-i

The first_private specifier in Optimization Control Line (OCL) is applied to variable '*parm*'.

[Message Explanation]

The first_private specifier in Optimization Control Line (OCL) is applied to variable.

[Parameters Explanation]

parm: Variable to which the first_private specifier in Optimization Control Line (OCL) is applied

[System Behavior]

Applies the first_private specifier in Optimization Control Line (OCL).

2.12 jwd5015p-i

The last_private specifier in Optimization Control Line (OCL) is applied to variable 'parm'.

[Message Explanation]

The last_private specifier in Optimization Control Line (OCL) is applied to variable.

[Parameters Explanation]

parm : Variable to which the last_private specifier in Optimization Control Line (OCL) is applied

[System Behavior]

Applies the last_private specifier in Optimization Control Line (OCL).

2.13 jwd5016p-i

This loop is partially parallelized.

[Message Explanation]

This loop is partially parallelized.

[System Behavior]

Performs partially automatic parallelization to the loop.

2.14 jwd5018p-i

This loop is parallelized by cyclic distribution. The block size is *num*.

[Message Explanation]

The loop is automatically parallelized by cyclic distribution. The block size of cyclic distribution is *num*.

[Parameters Explanation]

num : Block size

[System Behavior]

The loop is automatically parallelized by cyclic distribution.

2.15 jwd5019p-i

The parallel regions are expanded.

[Message Explanation]

The parallel regions are expanded to reduce automatic parallelization overhead.

[System Behavior]

The parallel regions are expanded.

[Programmer Response]

The execution performance may be degraded on loops with small parallelization effects. In this case, the `-Knoregion_extension` option may improve execution performance.

2.16 jwd5101p-i

A statement in this loop prevents loop parallelization.

[Message Explanation]

The loop contains either of the following statement that prevents parallelization.

- Function call that generate by compiler
- Pointer assignment

[System Behavior]

Suppresses automatic parallelization to the loop.

2.17 jwd5122p-i

This loop is not parallelized because a function call that prevents parallelization exists in the loop.

[Message Explanation]

The loop contains a function call that prevents parallelization.

[System Behavior]

Suppresses automatic parallelization to the loop.

[Programmer Response]

If the execution result should not be different even if the loop is parallelized, the `independent` specifier in Optimization Control Line (OCL) can be used to promote automatic parallelization to the loop.

However automatic parallelization to the loop is not effective in the following cases though independent specifier is specified.

- The result of function is structure type.
- The result of function is array type or pointer type.
- The function contains structure type parameter.
- The function contains pointer type parameter.

2.18 jwd5123p-i

This loop is not parallelized because a structure data that prevents parallelization exists in the loop.

[Message Explanation]

The loop contains a structure type data that prevents parallelization.

[System Behavior]

Suppresses automatic parallelization to the loop.

2.19 jwd5125p-i

This loop is not parallelized because a standard library function that prevents parallelization exists in the loop.

[Message Explanation]

The loop contains a standard library function that is unable to parallelize.

[System Behavior]

Suppresses automatic parallelization to the loop.

2.20 jwd5131p-i

This loop is not parallelized because two or more exits of loop exist.

[Message Explanation]

The loop contains multiple exits that are unable to parallelize.

[System Behavior]

Suppresses automatic parallelization to the loop.

2.21 jwd5133p-i

This loop is not parallelized because this loop is too complicated.

[Message Explanation]

The complicated loop is unable to parallelize.

[System Behavior]

Suppresses automatic parallelization to the loop.

2.22 jwd5134p-i

This loop is not parallelized because non-integer type loop variable exists.

[Message Explanation]

The type of loop variable is not integer.

[System Behavior]

Suppresses automatic parallelization to the loop.

2.23 jwd5136p-i

This loop is not parallelized because this loop increment parameter may be redefined in the loop.

[Message Explanation]

The increment parameter of the loop appears at either of following:

- argument
- left side of the assignment statement

[System Behavior]

Suppresses automatic parallelization to the loop.

2.24 jwd5137p-i

This loop is not parallelized because the loop variable is external variable or union member.

[Message Explanation]

The loop variable is either of following:

- argument
- external variable

- union member

[System Behavior]

Suppresses automatic parallelization to the loop.

2.25 jwd5138p-i

This loop is not parallelized because the loop variable may be redefined in the loop.

[Message Explanation]

The loop variable of the loop appears at either of following:

- argument
- left side of an assignment statement

[System Behavior]

Suppresses automatic parallelization to the loop.

2.26 jwd5141p-i

This loop is not parallelized because too many statements exist in the function.

[Message Explanation]

This loop is not parallelized because there are too many executable statements in the function.

[System Behavior]

Suppresses automatic parallelization to the loop.

2.27 jwd5142p-i

This loop is not parallelized because its iteration count is uncertainty.

[Message Explanation]

The iteration of the loop is not countable for parallelization.

[System Behavior]

Suppresses automatic parallelization to the loop.

2.28 jwd5143p-i

This loop is not parallelized because its iteration count is too few.

[Message Explanation]

The iteration of the loop is insufficient for parallelization.

[System Behavior]

Suppresses automatic parallelization to the loop.

2.29 jwd5156p-i

This loop is not parallelized because the serial specifier in Optimization Control Line (OCL) is specified.

[Message Explanation]

The `serial` specifier in Optimization Control Line (OCL) is specified for the loop.

[System Behavior]

Suppresses automatic parallelization to the loop.

2.30 jwd5201p-i

This loop is not parallelized because the order of the definition and reference to the array '*parm*' is different from serial execution because of data dependency.

[Message Explanation]

The loop contains an array data dependence that prevents parallelization.

[Parameters Explanation]

parm: Array that prevents parallelization

[System Behavior]

Suppresses automatic parallelization to the loop.

2.31 jwd5202p-i

This loop is not parallelized because the order of the definition and reference to the variable '*parm*' is different from serial execution because of data dependency.

[Message Explanation]

The loop contains a variable reference that prevents parallelization.

[Parameters Explanation]

parm: Variable that prevents parallelization

[System Behavior]

Suppresses automatic parallelization to the loop.

2.32 jwd5203p-i

This loop is not parallelized because the order of the definition and reference to the array '*parm2*' may be different from serial execution because of data dependency with variable '*parm*' in subscript expression.

[Message Explanation]

The loop contains an array that prevents parallelization.

[Parameters Explanation]

parm: Subscript of *parm2*

parm2: Array that prevents parallelization

[System Behavior]

Suppresses automatic parallelization to the loop.

2.33 jwd5204p-i

This loop is not parallelized because uncertain relation between the variables '*parm*' and '*parm2*' may cause different results from serial execution.

[Message Explanation]

The loop contains variables that prevent parallelization.

[Parameters Explanation]

parm : Variable that prevents parallelization

parm2 : Variable that prevents parallelization

[System Behavior]

Suppresses automatic parallelization to the loop.

2.34 jwd5205p-i

This loop is not parallelized because the same array (the rank *num* of array '*parm*') is referred using different subscripts.

[Message Explanation]

The loop contains an array reference that prevents parallelization.

[Parameters Explanation]

parm : Array that prevents parallelization

num : Dimension number of *parm*

[System Behavior]

Suppresses automatic parallelization to the loop.

2.35 jwd5206p-i

This loop is not parallelized because unpredictable induction variable '*parm2*' in the dimension *num* of array '*parm*' may cause different results from serial execution.

[Message Explanation]

The loop contains an array reference that prevents parallelization.

[Parameters Explanation]

parm : Array that prevents parallelization

num : Dimension number of *parm*

parm2 : Induction variable that prevents parallelization

[System Behavior]

Suppresses automatic parallelization to the loop.

2.36 jwd5207p-i

This loop is not parallelized because the uncertain sign of loop increment parameter '*parm*' may cause different results from serial execution.

[Message Explanation]

The loop contains a variable that prevents parallelization.

[Parameters Explanation]

parm : Loop increment parameter

[System Behavior]

Suppresses automatic parallelization to the loop.

2.37 jwd5208p-i

This loop is not parallelized because the uncertain order of the definition and reference to variable '*parm*' may cause different results from serial execution.

[Message Explanation]

The loop contains a variable that prevents parallelization.

[Parameters Explanation]

parm : Variable that prevents parallelization

[System Behavior]

Suppresses automatic parallelization to the loop.

2.38 jwd5209p-i

This loop is not parallelized because the variable '*parm2*' in subscript expression of array '*parm*' is defined in the loop.

[Message Explanation]

The loop contains a variable in subscript expression of array that prevents parallelization.

[Parameters Explanation]

parm : Array that prevents parallelization

parm2 : Subscript of *parm*

[System Behavior]

Suppresses automatic parallelization to the loop.

2.39 jwd5210p-i

This loop is not parallelized because too complicated expressions in for statement expressions exist in the loop.

[Message Explanation]

The complicated loop is unable to parallelize.

[System Behavior]

Suppresses automatic parallelization to the loop.

2.40 jwd5211p-i

This loop is not parallelized because the induction variable '*parm*' in subscript expression of array '*parm2*' is defined between references.

[Message Explanation]

The loop contains a variable that prevents parallelization.

[Parameters Explanation]

parm : Subscript of *parm2*

parm2 : Array that prevents parallelization

[System Behavior]

Suppresses automatic parallelization to the loop.

2.41 jwd5221p-i

The operation of arrays is not parallelized because the arrays '*parm*' and '*parm2*' which are external variables or union members may cause different results from serial execution.

[Message Explanation]

The loop contains arrays that prevent parallelization.

[Parameters Explanation]

parm : Array that prevents parallelization

parm2 : Array that prevents parallelization

[System Behavior]

Suppresses automatic parallelization to the loop.

2.42 jwd5222p-i

The operation of variable is not parallelized because the variable '*parm*' is external variable or union member.

[Message Explanation]

The loop contains a variable reference that prevents parallelization as follows:

- external variable
- union member

[Parameters Explanation]

parm : Variable that prevents parallelization

[System Behavior]

Suppresses automatic parallelization to the loop.

2.43 jwd5223p-i

The operation of variables is not parallelized because the variables '*parm*' and '*parm2*' are members of same union.

[Message Explanation]

The loop contains variables that prevent parallelization.

[Parameters Explanation]

parm : Variable that prevents parallelization

parm2: Variable that prevents parallelization

[System Behavior]

Suppresses automatic parallelization to the loop.

2.44 jwd5228p-i

This loop is not parallelized because the order of the definition and reference to the variable is different from serial execution because of data dependency.

[Message Explanation]

The loop contains variables that prevent parallelization.

[System Behavior]

Suppresses automatic parallelization to the loop.

2.45 jwd5302p-i

This loop is not parallelized because the performance of a partial parallelization execution may not be improved.

[Message Explanation]

This loop is not parallelized because the performance of a partial parallelization execution may not be improved.

[System Behavior]

Suppresses automatic parallelization to the loop.

2.46 jwd5401i-i

Because the collapse clause including the innermost loop was specified, there is a possibility that the execution performance decreases.

[Message Explanation]

The execution performance might decrease because the instructions such as division are generated to the loop body by the collapse clause including an innermost loop, and SIMD conversion and the software pipelining are not applied.

[System Behavior]

Continues processing.

[Programmer Response]

Optimization is applied to the innermost loop by excluding the innermost loop from the loops which are associated by the collapse clause, and the execution performance decrease might be able to be prevented.

Do either the following to exclude the innermost loop from the loops which are associated by the collapse clause:

- Modify the program.
- Specify `-Kopenmp_collapse_except_innermost` option.

2.47 jwd5402i-i

The innermost loop was excluded from the loops which are associated by the collapse clause.

[System Behavior]

Exclude the innermost loop from the loops which are associated by the collapse clause because -Kopenmp_collapse_except_innermost option was specified.

2.48 jwd5403i-i

SIMD version of the function specified with the declare simd directive was not created.

[Message Explanation]

SIMD version of the function specified with the declare simd directive might not be created because of the clauses specified in the declare simd directive or the arguments of the function.

[System Behavior]

Continues processing.

[Programmer Response]

SIMD version of the function specified with the declare simd directive might be created by deleting the simklen clause of the declare simd directive.

2.49 jwd5404i-i

The simklen clause was ignored.

[Message Explanation]

The simklen clause was ignored.

[System Behavior]

Ignores the simklen clause, and continues processing.

[Programmer Response]

Delete the simklen clause of the declare simd directive.

2.50 jwd6001s-i

SIMD conversion is applied to this loop with the loop variable '*parm*'.

[Message Explanation]

The loop is optimized using SIMD instructions.

[Parameters Explanation]

parm: Loop variable of the optimized loop using SIMD instructions

[System Behavior]

SIMD instructions are used.

2.51 jwd6002s-i

SIMD conversion is applied to this loop.

[Message Explanation]

The loop is optimized using SIMD instructions.

[System Behavior]

SIMD instructions are used.

2.52 jwd6004s-i

SIMD conversion is applied to this loop with the loop variable '*parm*'. The loop contains a reduction operation.

[Message Explanation]

SIMD conversion is applied to loop by changing the evaluation order of the expression which contains commutative operation such as addition or multiplication according to the commutative law.

When this message is output, side effects (calculation errors) may occur in the execution results.

[Parameters Explanation]

parm : Loop variable of the loop to which SIMD conversion is applied

[System Behavior]

SIMD instructions are used.

[Programmer Response]

When it seems that a side effect by optimization in the execution result, re-compile without the -Kreduction and -Keval options, then re-run the job and verify the result.

2.53 jwd6005s-i

SIMD conversion is applied partially to this loop with the loop variable '*parm*'.

[Message Explanation]

The loop is partially optimized using SIMD instructions.

[Parameters Explanation]

parm : Loop variable of the partially optimized loop using SIMD instructions

[System Behavior]

SIMD instructions are partially used.

2.54 jwd6007s-i

SIMD conversion is applied to this loop which contains a reduction operation.

[Message Explanation]

The evaluation order of the expression which contains commutative operation such as addition or multiplication was changed according to the commutative law.

When this message is output, side effects (calculation errors) may occur in the execution results.

[System Behavior]

SIMD instructions are used.

[Programmer Response]

When it seems that a side effect by optimization in the execution result, re-compile without the -Kreduction and -Keval options, then re-run the job and verify the result.

2.55 jwd6016s-i

SIMD conversion is applied partially to this loop.

[Message Explanation]

The loop is partially optimized using SIMD instructions.

[System Behavior]

SIMD instructions are partially used.

2.56 jwd6025s-i

List vector conversion is applied to the executable statements in "if" statement.

[Message Explanation]

List vector conversion is applied to the executable statements in "if" statement.

[System Behavior]

Applies list vector conversion to the executable statements in "if" statement.

2.57 jwd6101s-i

SIMD conversion is not applied because a statement that prevents SIMD conversion exists.

[Message Explanation]

The loop contains either of the following statement that prevents SIMD conversion.

- Function call that generated by compiler
- Pointer assignment

[System Behavior]

SIMD instructions are not used.

2.58 jwd6122s-i

SIMD conversion is not applied to this loop because a function call that prevents SIMD conversion exists.

[Message Explanation]

The loop contains a function call that prevents SIMD conversion.

[System Behavior]

SIMD instructions are not used.

2.59 jwd6123s-i

SIMD conversion is not applied to this loop because a type of data that prevents SIMD conversion exists.

[Message Explanation]

The loop contains either of the following types that prevent SIMD conversion.

- short int type
- long double type

- long double complex type
- character type
- structure type

[System Behavior]

SIMD instructions are not used.

2.60 jwd6125s-i

SIMD conversion is not applied to this loop because a standard library function that prevents SIMD conversion exists.

[Message Explanation]

The loop contains a standard library function that prevents SIMD conversion.

[System Behavior]

SIMD instructions are not used.

2.61 jwd6131s-i

SIMD conversion is not applied to this loop because the loop has two or more exits.

[Message Explanation]

The loop contains multiple exits that prevent SIMD conversion.

[System Behavior]

SIMD instructions are not used.

2.62 jwd6133s-i

SIMD conversion is not applied to this loop because the loop is too complicated.

[Message Explanation]

The complicated loop prevents SIMD conversion.

[System Behavior]

SIMD instructions are not used.

2.63 jwd6134s-i

SIMD conversion is not applied to this loop because non-integer type loop variable exists.

[Message Explanation]

The loop variable type of the loop is not integer.

[System Behavior]

SIMD instructions are not used.

2.64 jwd6137s-i

SIMD conversion is not applied to this loop because the loop variable is external variable or union member.

[Message Explanation]

The loop variable is either of following:

- argument
- external variable
- union member

[System Behavior]

SIMD instructions are not used.

2.65 jwd6138s-i

SIMD conversion is not applied to this loop because the loop variable may be redefined in the loop.

[Message Explanation]

The loop variable of the loop appears at either of following:

- argument
- left side of an assignment statement

[System Behavior]

SIMD instructions are not used.

2.66 jwd6141s-i

SIMD conversion is not applied to this loop because too many statements exist in the function.

[Message Explanation]

SIMD conversion cannot be applied to this loop because too many executable statements exist in the function.

[System Behavior]

SIMD instructions are not used.

2.67 jwd6142s-i

SIMD conversion is not applied to this loop because the iteration count is uncertainty.

[Message Explanation]

SIMD conversion cannot be applied to loop since its iteration count is uncertainty.

[System Behavior]

SIMD instructions are not used.

2.68 jwd6143s-i

SIMD conversion is not applied to this loop because the iteration count is smaller than the SIMD width.

[Message Explanation]

SIMD conversion cannot be applied to this loop because the value of its iteration count is smaller than the SIMD width.

[System Behavior]

SIMD instructions are not used.

2.69 jwd6144s-i

SIMD conversion is not applied to this loop because the effect cannot be expected.

[Message Explanation]

SIMD conversion is not applied to this loop because the effect cannot be expected.

[System Behavior]

SIMD instructions are not used.

2.70 jwd6145s-i

SIMD conversion is not applied to this loop because the definition and the reference to data are non-continuous accesses.

[Message Explanation]

SIMD conversion is not applied to this loop because SIMD instructions to read or write non-continuous memory data are not supported.

[System Behavior]

SIMD instructions are not used.

2.71 jwd6146s-i

SIMD conversion is not applied to this loop because the performance of a loop including "if" statement may not be improved.

[Message Explanation]

SIMD conversion is not applied to this loop because the performance of a loop including "if" statement may not be improved.

[System Behavior]

SIMD instructions are not used.

[Programmer Response]

Specify the `-Ksimd=2` option to promote SIMD conversion.

In order to redundantly execute instructions within "if" statements, performance may be reduced depending on the true ratio of the "if" statements.

In addition, as with the `-Kpreex` option, speculative executions are performed within an expression in "if" statements, so this optimization may execute an instruction which would not be executed based on the logic of the program, causing an error.

2.72 jwd6156s-i

SIMD conversion is not applied to this loop because the `nosimd` specifier in Optimization Control Line (OCL) is specified.

[Message Explanation]

The `nosimd` specifier is specified for the loop.

[System Behavior]

SIMD instructions are not used.

2.73 jwd6201s-i

SIMD conversion is not applied to this loop because the order of the definition and reference to the array '*parm*' is different from serial execution because of data dependency.

[Message Explanation]

The loop contains an array data reference that prevents SIMD conversion.

[Parameters Explanation]

parm : Array that prevents SIMD conversion

[System Behavior]

SIMD instructions are not used.

2.74 jwd6202s-i

SIMD conversion is not applied to this loop because the order of the definition and reference to the variable '*parm*' is different from serial execution because of data dependency.

[Message Explanation]

The loop contains a variable reference that prevents SIMD conversion.

[Parameters Explanation]

parm : Variable that prevents SIMD conversion

[System Behavior]

SIMD instructions are not used.

2.75 jwd6203s-i

SIMD conversion is not applied to this loop because the order of the definition and reference to the array '*parm2*' may be different from serial execution because of data dependency with variable '*parm*' in subscript expression.

[Message Explanation]

The loop contains an array reference that prevents SIMD conversion.

[Parameters Explanation]

parm : Subscript of *parm2*

parm2 : Array that prevents SIMD conversion

[System Behavior]

SIMD instructions are not used.

2.76 jwd6204s-i

SIMD conversion is not applied to this loop because uncertain relation between the variables '*parm*' and '*parm2*' may cause different results from serial execution.

[Message Explanation]

The loop contains an array reference that prevents SIMD conversion.

[Parameters Explanation]

parm : Variable that prevents SIMD conversion

parm2 : Variable that prevents SIMD conversion

[System Behavior]

SIMD instructions are not used.

2.77 jwd6205s-i

SIMD conversion is not applied to this loop because the same array (the rank *num* of array '*parm*') is referred using different subscripts.

[Message Explanation]

The loop contains an array reference that prevents SIMD conversion.

[Parameters Explanation]

parm : Array that prevents SIMD conversion

num : Dimension number of *parm*

[System Behavior]

SIMD instructions are not used.

2.78 jwd6206s-i

SIMD conversion is not applied to this loop because unpredictable induction variable '*parm2*' in the dimension *num* of array '*parm*' may cause different results from serial execution.

[Message Explanation]

The loop contains an array reference that prevents SIMD conversion.

[Parameters Explanation]

parm : Array that prevents SIMD conversion

num : Dimension number of *parm*

parm2 : Induction variable that prevents SIMD conversion

[System Behavior]

SIMD instructions are not used.

2.79 jwd6207s-i

SIMD conversion is not applied to this loop because the uncertain sign of loop increment parameter '*parm*' may cause different results from serial execution.

[Message Explanation]

The loop contains a variable that prevents SIMD conversion.

[Parameters Explanation]

parm : Loop incrementation parameter which sign is uncertainty

[System Behavior]

SIMD instructions are not used.

2.80 jwd6208s-i

SIMD conversion is not applied to this loop because the uncertain order of the definition and reference to variable '*parm*' may cause different results from serial execution.

[Message Explanation]

The loop contains a variable that prevents SIMD conversion.

[Parameters Explanation]

parm : Variable that prevents SIMD conversion

[System Behavior]

SIMD instructions are not used.

2.81 jwd6209s-i

SIMD conversion is not applied to this loop because the variable '*parm2*' in subscript expression of array '*parm*' is defined in the loop.

[Message Explanation]

The loop contains a variable that prevents SIMD conversion.

[Parameters Explanation]

parm : Array that prevents SIMD conversion

parm2 : Subscript of *parm*

[System Behavior]

SIMD instructions are not used.

2.82 jwd6210s-i

SIMD conversion is not applied to this loop because too complicated expressions in for statement expressions exist in the loop.

[Message Explanation]

The complicated loop is unable to SIMD conversion.

[System Behavior]

SIMD instructions are not used.

2.83 jwd6211s-i

SIMD conversion is not applied to this loop because the induction variable '*parm*' in subscript expression of array '*parm2*' is defined between references.

[Message Explanation]

The loop contains a variable that prevents SIMD conversion.

[Parameters Explanation]

parm : Subscript of *parm2*

parm2 : Array that prevents SIMD conversion

[System Behavior]

SIMD instructions are not used.

2.84 jwd6221s-i

SIMD conversion is not applied to the operation of arrays because the arrays '*parm*' and '*parm2*' which are external variables or union members may cause different results from serial execution.

[Message Explanation]

The loop contains arrays that prevent SIMD conversion.

[Parameters Explanation]

parm: Array that prevents SIMD conversion

parm2: Array that prevents SIMD conversion

[System Behavior]

SIMD instructions are not used.

2.85 jwd6222s-i

SIMD conversion is not applied to this loop because the variable '*parm*' is external variable or union member.

[Message Explanation]

The loop contains a variable reference that prevents SIMD conversion as follows:

- external variable
- union member

[Parameters Explanation]

parm: Variable that prevents SIMD conversion

[System Behavior]

SIMD instructions are not used.

2.86 jwd6223s-i

SIMD conversion is not applied to the operation of variables because '*parm*' and '*parm2*' are members of same union.

[Message Explanation]

The loop contains variables that prevent SIMD conversion.

[Parameters Explanation]

parm: Variable that prevents SIMD conversion

parm2: Variable that prevents SIMD conversion

[System Behavior]

SIMD instructions are not used.

2.87 jwd6228s-i

SIMD conversion is not applied to this loop because the order of the definition and reference to the variable is different from serial execution because of data dependency.

[Message Explanation]

The loop contains a variable that prevents SIMD conversion.

[System Behavior]

SIMD instructions are not used.

2.88 jwd6229s-i

SIMD conversion is not applied to this loop because "if" statements exist in the loop.

[Message Explanation]

SIMD conversion is not applied to loop that contains "if" statements.

[System Behavior]

SIMD instructions are not used.

[Programmer Response]

Specify the `-Ksimd=2` option to promote SIMD conversion.

In order to redundantly execute instructions within "if" statements, performance may be reduced depending on the true ratio of the "if" statements.

In addition, as with the `-Kpreex` option, speculative executions are performed within an expression in "if" statements, so this optimization may execute an instruction which would not be executed based on the logic of the program, causing an error.

2.89 jwd6302s-i

SIMD conversion is not applied to this loop because the performance of a partial SIMD execution may not be improved.

[Message Explanation]

SIMD conversion is not applied to this loop because the performance of a partial SIMD execution may not be improved.

[System Behavior]

SIMD instructions are not used.

2.90 jwd6402s-i

SIMD conversion is applied partially to this function by UXSIMD.

[Message Explanation]

The function is partially applied SIMD conversion by UXSIMD.

[System Behavior]

The function is automatically applied SIMD conversion by UXSIMD.

2.91 jwd6501s-i

This loop with variable '*parm*' is vectorized by OpenMP SIMD.

[Message Explanation]

Vectorized by OpenMP SIMD.

[Parameter Explanation]

parm : Loop variable of the vectorized loop

[System Behavior]

Vectorized by OpenMP SIMD.

2.92 jwd6502s-i

This loop is vectorized by OpenMP SIMD.

[Message Explanation]

Vectorized by OpenMP SIMD.

[System Behavior]

Vectorized by OpenMP SIMD.

2.93 jwd6504s-i

SIMDLEN(*num*) clause is ignored. SIMDLEN is *num2*.

[Message Explanation]

Vectorized by OpenMP SIMD, but SIMDLEN(*num*) clause is ignored.

[Parameters Explanation]

num : The value of specified SIMDLEN

num2 : The value of effective SIMDLEN

[System Behavior]

Vectorized by OpenMP SIMD.

2.94 jwd6505s-i

This function '*parm*' is not vectorized by DECLARE SIMD.

[Message Explanation]

Vectorized by OpenMP SIMD, but this function is not SIMD execution.

[Parameters Explanation]

parm : Function name of not SIMD execution

[System Behavior]

Vectorized by OpenMP SIMD.

2.95 jwd6506s-i

This function '*parm*' does not match any clause.

[Message Explanation]

Vectorized by OpenMP SIMD, but this function does not match INBRANCH, NOTINBRANCH, LINEAR, or UNIFORM clause.

[Parameters Explanation]

parm : Function name of not match

[System Behavior]

Vectorized by OpenMP SIMD.

2.96 jwd6510s-i

OpenMP SIMD conversion cannot be applied to loop : the loop is not innermost.

[Message Explanation]

OpenMP SIMD conversion cannot be applied to loop, because outer-loop vectorization is not supported.

[System Behavior]

This loop is not vectorized.

2.97 jwd6512s-i

OpenMP SIMD conversion cannot be applied to loop : SIMDLEN(1) is specified.

[Message Explanation]

OpenMP SIMD conversion cannot be applied to loop, because SIMDLEN(1) is specified.

[System Behavior]

This loop is not vectorized.

2.98 jwd6513s-i

OpenMP SIMD conversion cannot be applied to loop : SAFELEN(1) is specified.

[Message Explanation]

OpenMP SIMD conversion cannot be applied to loop, because SAFELEN(1) is specified.

[System Behavior]

This loop is not vectorized.

2.99 jwd8101o-i

Inline expansion is applied to the user defined function '*parm*'.

[Message Explanation]

The user-defined function is expanded inline.

[Parameters Explanation]

parm : Name of the user-defined function that was expanded inline

[System Behavior]

Performs inline expansion for the user-defined function at the referenced position.

2.100 jwd8102o-i

Inline expansion is not applied to the user defined function '*parm*' because of recursive call.

[Message Explanation]

The user-defined function is not expanded inline because it calls itself directly or indirectly.

[Parameters Explanation]

parm : Name of the user-defined function that is not expanded inline

[System Behavior]

Suppresses inline expansion for the user-defined function at the referenced position.

2.101 jwd8201o-i

Invariant expression within loop is moved to outside loop.

[Message Explanation]

An invariant (expression that value remains unchanged) in a statement that was optionally executed according to the conditions moved to outside of the loop.

This message is output when the -Kpreex option is effective.

When this message is output, side effects (calculation errors or runtime exceptions) may occur in the execution results.

[System Behavior]

Moves loop invariant to outside of the loop.

[Programmer Response]

When it seems that a side effect by optimization in the execution result, re-compile without the -Kpreex option, then re-run the job and verify the result.

2.102 jwd8202o-i

Loop unrolling expanding *num* times is applied to this loop.

[Message Explanation]

Expands executable statements in the loop *num* times.

[Parameters Explanation]

num : Number of times that executable statements in the loop were expanded

[System Behavior]

Applies loop unrolling.

2.103 jwd8203o-i

Loop full unrolling is applied to this loop.

[Message Explanation]

Expands executable statements by the number of iterations of the loop.

[System Behavior]

Applies full unrolling.

2.104 jwd8204o-i

This loop is software pipelined.

[Message Explanation]

This loop is applied software pipelining.

[System Behavior]

Applies software pipelining to the loop.

2.105 jwd8205o-i

The software-pipelined loop is chosen at run time when the iteration count is greater than or equal to *num*.

[Message Explanation]

The software-pipelined loop is chosen at run time when the iteration count is greater than or equal to *num*.

[Parameters Explanation]

num : The minimum iteration count of the loop in the source program that software pipelining requires

[System Behavior]

Applies software pipelining to the loop and inserts a branch instruction to choose the software-pipelined loop or the original loop at run time.

[Programmer Response]

Set the iteration count of the loop greater than or equal to *num* in order to choose the software-pipelined loop at run time. Change the application situation of optimizations to the loop, such as unrolling times, in order to change *num*.

2.106 jwd8206o-i

Division is changed to multiplication by reciprocal.

[Message Explanation]

A division is changed to multiplication by the reciprocal.

This message is output when the -Keval option is effective.

When this message is output, side effects (calculation errors or runtime exceptions) may occur in the execution results.

[System Behavior]

Performs the optimization that changes the method of operator evaluation.

[Programmer Response]

When it seems that a side effect by optimization in the execution result, re-compile without the -Keval option, then re-run the job and verify the result.

2.107 jwd8207o-i

Complex division method is changed.

[Message Explanation]

A complex division is changed to other calculating method of complex division.

This message is output when the -Keval option is effective.

When this message is output, side effects (calculation errors or runtime exceptions) may occur in the execution results.

[System Behavior]

Performs the optimization that changes the method of operator evaluation.

[Programmer Response]

When it seems that a side effect by optimization in the execution result, re-compile without the -Keval option, then re-run the job and verify the result.

2.108 jwd8208o-i

Method of calculating sum or product is changed.

[Message Explanation]

A sum or product is changed to other calculating method of sum or product.

This message is output when the -Keval option is effective.

When this message is output, side effects (calculation errors or runtime exceptions) may occur in the execution results.

[System Behavior]

Performs the optimization that changes the method of operator evaluation.

[Programmer Response]

When it seems that a side effect by optimization in the execution result, re-compile without the -Keval option, then re-run the job and verify the result.

2.109 jwd8209o-i

Evaluation order of polynomial expression is changed according to commutative law of addition and multiplication.

[Message Explanation]

The evaluation order of the polynomial expression which contains commutative operation such as addition and multiplication is changed according to the commutative law.

This message is output when the -Keval option is effective.

When this message is output, side effects (calculation errors or runtime exceptions) may occur in the execution results.

[System Behavior]

Performs the optimization that changes the method of operator evaluation.

[Programmer Response]

When it seems that a side effect by optimization in the execution result, re-compile without the -Keval option, then re-run the job and verify the result.

2.110 jwd8210o-i

The loops are fused. (line: *num*)

[Message Explanation]

The neighboring loops are fused.

[Parameters Explanation]

num : Line number of fused loop

[System Behavior]

Fuses neighboring loops.

2.111 jwd8211o-i

The loops are interchanged. (line: *num*)

[Message Explanation]

The nested loops are interchanged.

[Parameters Explanation]

parm : Line number of interchanged loop

[System Behavior]

Interchanges nested loops.

2.112 jwd8212o-i

The loop is fissioned into *num* loops.

[Message Explanation]

The loop is fissioned into multiple loops.

[Parameters Explanation]

num : The number of loops after fission

[System Behavior]

Fissions loop.

2.113 jwd8213o-i

Exponentiation method is changed.

[Message Explanation]

An exponentiation changed to other calculating method of exponentiation.

This message is output when the -Keval option is effective.

When this message is output, side effects (calculation errors or runtime exceptions) may occur in the execution results.

[System Behavior]

Performs the optimization that changes the method of operator evaluation.

[Programmer Response]

When it seems that a side effect by optimization in the execution result, re-compile without the -Keval option, then re-run the job and verify the result.

2.114 jwd8214o-i

Loop versioning is applied to this loop.

[Message Explanation]

This loop is applied loop versioning.

[System Behavior]

Applies loop versioning.

2.115 jwd8215o-i

Imperfectly nested loop is fissioned into perfectly nested loops.

[Message Explanation]

Imperfectly nested loop was fissioned into perfectly nested loops.

[System Behavior]

Imperfectly nested loop is fissioned into perfectly nested loops.

2.116 jwd8216o-i

This loop is not fissioned because no result of loop fission is obtained.

[Message Explanation]

This loop is not fissioned because no result of loop fission is obtained.

[System Behavior]

Not apply loop fission to the loop.

2.117 jwd8217o-i

Temporary arrays for loop fission are generated. (The number of arrays: *num*)

[Message Explanation]

The compiler generates temporary arrays to pass data between fissioned loops.

[Parameters Explanation]

num : The number of temporary arrays

[System Behavior]

Apply the loop fission and generates temporary arrays.

2.118 jwd8218o-i

Store the data referenced by the subsequent loops in the temporary arrays with loop fission. (The number of used arrays: *num*)

[Message Explanation]

Store the data referenced by the subsequent loops in the temporary array with the loop fission.

[Parameters Explanation]

num : The number of temporary arrays

[System Behavior]

Use temporary arrays generated by the compiler.

2.119 jwd8219o-i

The ratio of the number of instructions in the fissioned loop to the original loop is *num* %.

[Message Explanation]

Indicates the percentage of the number of instructions in the particular loop after fissioned, to the original loops.

[Parameters Explanation]

num : Ratio of the number of instructions

[System Behavior]

Apply the loop fission.

2.120 jwd8220o-i

Optimizations that may cause side effect are applied.

[Message Explanation]

At least one of the following optimizations is performed; a side effect (calculation errors or runtime exceptions) may occur in the execution results.

- Pre-execution of invariant expression
Performed when the `-Kpreex` option is effective.
- Modification of operation evaluation method
Performed when the `-Keval` option is effective.

[System Behavior]

Performs optimization that may cause side effect.

[Programmer Response]

When this message is output, keep the program compilation list or note the program name down.

When it seems that a side effect by optimization in the execution result, re-compile without the `-Kpreex` and `-Keval` options, then re-run the job and verify the result.

2.121 jwd8222o-i

The prefetch instructions were generated because the number of prefetch required in this loop is greater than the allowable number of hardware-prefetch.

[Message Explanation]

The prefetch instructions are generated against the number of prefetch judged necessary in the loop which is greater than the allowable number of hardware-prefetch. This message is output when `-Kprefetch_sequential=auto` option is effective.

[System Behavior]

The prefetch instructions are generated.

2.122 jwd8223o-i

Speculative execution optimization of the load instruction is applied.

[Message Explanation]

Applies the speculative execution optimization of the load instruction.

This message is output only when specified the -Kpreload option.

If this message is output, side effects of optimization (an abnormal end) may occur during execution.

[System Behavior]

Applies the speculative execution optimization of the load instruction.

[Programmer Response]

If the program is abnormally ended, compile the program without the -Kpreload option and execute again to check for side effects of optimization.

2.123 jwd8224o-i

Loop fission is not applied because the specifier which prevents loop fission is specified.

[Message Explanation]

Loop fission is not applied because one or more of the following specifiers and function which prevent loop fission are specified.

- builtin_prefetch function
- simd construct of OpenMP

[System Behavior]

Not apply loop fission to the loop.

[Programmer Response]

Take one of the following actions to promote loop fission:

- Disable the specifiers which prevent loop fission.
- Change the source program to fission loop.

2.124 jwd8230o-i

Unroll-and-jam optimization is applied to this loop with the loop variable '*parm*'. (Unrolling times : '*num*')

[Message Explanation]

Unroll-and-jam optimization whose number of unrolling times is '*num*' is applied to this loop with variable '*parm*'.

[Parameters Explanation]

parm : Loop variable of the loop to which unroll-and-jam optimization is applied

num : The number of unrolling times

[System Behavior]

Applies unroll-and-jam optimization.

2.125 jwd8231o-i

Unroll-and-jam optimization is applied to this loop. (Unrolling times : '*num*')

[Message Explanation]

Unroll-and-jam optimization whose number of unrolling times is '*num*' is applied to this loop.

[Parameters Explanation]

num : The number of unrolling times

[System Behavior]

Applies unroll-and-jam optimization.

2.126 jwd8300o-i

The intrinsic function is changed to a multi-operation function.

[Message Explanation]

The standard library function is changed to a multi-operation function.

[System Behavior]

Changes the standard library function to a multi-operation function.

2.127 jwd8310o-i

Loop striping is applied to this loop with the loop variable '*parm*'.

[Message Explanation]

This loop is applied loop striping.

[Parameters Explanation]

parm : Loop variable of the loop to which loop striping is applied

[System Behavior]

Applies loop striping.

2.128 jwd8311o-i

Loop striping is applied partially to this loop with the loop variable '*parm*'.

[Message Explanation]

This loop is applied partially loop striping.

[Parameters Explanation]

parm : Loop variable of the loop to which loop striping is applied partially

[System Behavior]

Applies partially loop striping.

2.129 jwd8312o-i

Loop striping is applied to this loop.

[Message Explanation]

This loop is applied loop striping.

[System Behavior]

Applies loop striping.

2.130 jwd8320o-i

Loop blocking by block size *num* is applied to this loop.

[Message Explanation]

The loop blocking for optimization is performed.

[Parameters Explanation]

num : Block size

[System Behavior]

Adopts loop blocking to the loop with size *num*.

2.131 jwd8321o-i

Loop unswitching is applied to this loop.

[Message Explanation]

The loop is unswitched.

[System Behavior]

This loop is unswitched.

2.132 jwd8322o-i

Loop strip-mining by strip size *num* is applied to this loop.

[Message Explanation]

The loop strip-mining optimization is performed.

[Parameters Explanation]

num : Strip size

[System Behavior]

Applies loop strip-mining to the loop with size *num*.

2.133 jwd8324o-i

Clone optimization is applied to this loop.

[Message Explanation]

This loop is applied clone optimization.

[System Behavior]

Applies clone optimization to this loop.

2.134 jwd8325o-i

Assume that the maximum number of iterations for this loop is *parm* from the array declaration.

[Message Explanation]

The iterations count of this loop is assumed to be less than or equal to *parm*, based on the optimization by the option -Karray_declaration_opt.

If there is an illegal access in the loop that exceeds the declared range of an array (even if the address is computationally correct), the intended result may not be obtained.

[Parameters Explanation]

parm : Maximum iteration count of a loop

[System Behavior]

Assume that the iteration count of the loop is *parm* or less.

[Programmer Response]

If the result is not as intended, specify the option `-Knoarray_declaration_opt`.

2.135 jwd8330o-i

Loop collapse is applied to the nested loops with loop variables '*parm*',...,'*parm2*'.

[Message Explanation]

The nested loops are collapsed into a single loop.

[Parameters Explanation]

parm,parm2 : Loop variable of the loop collapsed into a single loop

[System Behavior]

Collapses nested loop into a single loop.

2.136 jwd8331o-i

This loop is changed to the `matmul` library function call.

[Message Explanation]

This loop was changed to the library call (`matmul`).

[System Behavior]

Changes this loop to the library call (`matmul`).

2.137 jwd8400o-i

Link time optimization is applied.

[Message Explanation]

Link time optimization is performed.

[System Behavior]

Performs link time optimization.

2.138 jwd8450i-w

Link time optimization is not applied because the shared object '*parm*' is not found.

[Message Explanation]

Link time optimization is not applied because the shared object is not found.

[Parameters Explanation]

parm : Absolute path to shared object

[System Behavior]

Not apply link time optimization.

[Programmer Response]

Contact the system manager.

2.139 jwd8451i-w

Link time optimization is not applied because this object file was generated without -Klto option.

[Message Explanation]

Link time optimization is not applied because this object file was generated without -Klto option.

[System Behavior]

Not apply link time optimization.

[Programmer Response]

Re-compile with -Klto option and re-generate object file.

2.140 jwd8452i-w

Link time optimization is not applied because the compiler version or level does not match.

[Message Explanation]

Link time optimization is not applied because the version or level of compiler, which generated the object file, does not match it at linking time.

[System Behavior]

Not apply link time optimization.

[Programmer Response]

Re-compile with -Klto option and re-generate object file.

2.141 jwd8642i-u

Static data area or common block is too large.

[Message Explanation]

The static data area or a common block exceeds the system limit.

[System Behavior]

Stops the compilation.

[Programmer Response]

Reduce the program by dividing.

2.142 jwd8643i-u

Data area allocated to the stack is too large.

[Message Explanation]

The dynamic data area exceeds the system limit.

[System Behavior]

Stops the compilation.

[Programmer Response]

Reduce the program by dividing.

2.143 jwd8660i-i

This loop has the possibility of L1 instruction cache miss, and it may decrease the execution performance.

[Message Explanation]

L1 instruction cache miss may occur in the loop.

[Programmer Response]

Decrease the use of the L1 instruction cache in the loop.

For instance, the following methods may be effective.

- Suppress the optimizations which increase instructions in the loop such as loop unrolling, loop striping, etc.
- Reduce the loop by dividing.

2.144 jwd8662o-i

This loop is not software pipelined because no schedule is obtained.

[Message Explanation]

This loop is not software pipelined because applying software pipelining results in no schedule due to exceeding the internal time limit.

[System Behavior]

Not apply software pipelining to the loop.

[Programmer Response]

Specify -Kswp_strong option or the swp specifier in Optimization Control Line (OCL) in order to promote software pipelining by easing the condition. In case data dependence is known, give hints to the compiler, such as the norecurrence specifier in Optimization Control Line (OCL), or modify the source program to ease the dependence.

2.145 jwd8663o-i

This loop is not software pipelined because the software pipelining does not improve the performance.

[Message Explanation]

Obtains only worthless scheduling results although the compiler finished software pipelining,

[System Behavior]

Not apply software pipelining to the loop.

[Programmer Response]

Give hints to the compiler, such as the norecurrence specifier in Optimization Control Line (OCL), or modify the source program in order to promote software pipelining in case data dependence is known.

2.146 jwd8664o-i

This loop cannot be software pipelined because the loop contains an instruction, such as function call, which is not supported by software pipelining.

[Message Explanation]

This loop is not software pipelined because the loop contains an instruction not covered by the software pipelining, such as function call.

[System Behavior]

Not apply software pipelining to the loop.

[Programmer Response]

Delete the function call in the loop by inline expansion in order to promote software pipelining.

2.147 jwd8665o-i

This loop cannot be software pipelined because of shortage of integer registers.

[Message Explanation]

Obtains only scheduling results in which integer registers are short although the compiler finished software pipelining.

[System Behavior]

Not apply software pipelining to the loop.

[Programmer Response]

Take one of the following actions to promote software pipelining:

- Specify the `-Kswp_strong` option or the `swp` specifier in Optimization Control Line (OCL).
- Specify the `-Kswp_ireg_rate=N` option or the `swp_ireg_rate` specifier in Optimization Control Line (OCL) to increase available integer registers in software pipelining.
- Reduce the number of used integer registers by loop fission or any other way.

2.148 jwd8666o-i

This loop cannot be software pipelined because of shortage of floating-point registers.

[Message Explanation]

Obtains only scheduling results in which floating-point registers are short although the compiler finished software pipelining.

[System Behavior]

Not apply software pipelining to the loop.

[Programmer Response]

Take one of the following actions to promote software pipelining:

- Specify the `-Kswp_strong` option or the `swp` specifier in Optimization Control Line (OCL).
- Specify the `-Kswp_freg_rate=N` option or the `swp_freg_rate` specifier in Optimization Control Line (OCL) to increase available floating-point registers in software pipelining.
- Reduce the number of used floating-point registers by loop fission or any other way.

2.149 jwd8667o-i

This loop cannot be software pipelined because the function containing the loop is too large.

[Message Explanation]

This loop is not software pipelined because the function containing the loop is too large.

[System Behavior]

Not apply software pipelining to the loop.

[Programmer Response]

Divide the function into smaller functions in order to promote software pipelining.

2.150 jwd8668o-i

This loop is not software pipelined because the loop contains too many instructions.

[Message Explanation]

This loop is not software pipelined because the loop contains too many instructions.

[System Behavior]

Not apply software pipelining to the loop.

[Programmer Response]

Specify the `-Kswp_strong` option or the `swp` specifier in Optimization Control Line (OCL) in order to promote software pipelining by easing the condition. Reduce the number of instructions in the loop by loop fission or any other way.

2.151 jwd8669o-i

This loop is not software pipelined because the loop contains too many instructions accessing memory.

[Message Explanation]

This loop is not software pipelined because the loop contains too many instructions accessing memory.

[System Behavior]

Not apply software pipelining to the loop.

[Programmer Response]

Specify the `-Kswp_strong` option or the `swp` specifier in Optimization Control Line (OCL) in order to promote software pipelining by easing the condition. Reduce the number of instructions accessing memory in the loop by loop fission or any other way.

2.152 jwd8670o-i

This loop cannot be software pipelined because the loop contains a branch instruction.

[Message Explanation]

This loop is not software pipelined because the loop contains a branch instruction which is not for loop iteration.

[System Behavior]

Not apply software pipelining to the loop.

[Programmer Response]

Specify options, such as the `-Ksimd=2` option, or modify the source program to delete the branch instruction in the loop in order to promote software pipelining.

2.153 jwd8671o-i

This loop cannot be software pipelined because the shape of the loop is not covered by software pipelining.

[Message Explanation]

This loop is not software pipelined because the shape of the loop is not covered by the software pipelining.

[System Behavior]

Not apply software pipelining to the loop.

[Programmer Response]

Modify the source program to delete a jump out from the loop or jump into the loop in order to promote software pipelining.

2.154 jwd8672o-i

This loop is not software pipelined because 8-byte or greater type data could not be aligned on 8-byte boundaries.

[Message Explanation]

This loop is not software pipelined because 8-byte or greater type data could not be aligned on 8-byte boundaries and instructions not covered by software pipelining could be generated in the loop.

[System Behavior]

Not apply software pipelining to the loop.

[Programmer Response]

Specify the `-Kdalign` option in order to apply software pipelining to the loop.

2.155 jwd8673o-i

This loop cannot be software pipelined because of shortage of predicate registers.

[Message Explanation]

Obtains only scheduling results in which predicate registers are short although the compiler finished software pipelining.

[System Behavior]

Not apply software pipelining to the loop.

[Programmer Response]

Take one of the following actions to promote software pipelining:

- Specify the `-Kswp_strong` option or the `swp` specifier in Optimization Control Line (OCL).
- Specify the `-Kswp_preg_rate=N` option or the `swp_preg_rate` specifier in Optimization Control Line (OCL) to increase available predicate registers in software pipelining.
- Reduce the number of used predicate registers by loop fission or any other way.

2.156 jwd8674o-i

This loop is not software pipelined because the iteration count is smaller than *num* and the software pipelining does not improve the performance.

[Message Explanation]

This loop is not software pipelined because the iteration count is smaller than *num* and the software-pipelined loop is not chosen at run time.

[Parameters Explanation]

num : The minimum iteration count of the loop in the source program that software pipelining requires.

[System Behavior]

Not apply software pipelining to the loop.

[Programmer Response]

Set the iteration count of the loop greater than or equal to *num* in order to apply software pipelining to the loop. Change the application situation of optimizations to the loop, such as unrolling times, in order to change *num*.

2.157 jwd8695i-u

The compilation was canceled because the compilation of function '*parm*' was forecasted to take a long time.

[Message Explanation]

The compilation is canceled because it is forecasted to take a long time.

[Parameters Explanation]

parm : Function name

[System Behavior]

Cancels the compilation.

[Programmer Response]

Take one of the following actions:

- Specify the `-Nnocancel_overtime_compilation` option to avoid canceling the compilation.
- Specify `-O0` option to reduce the compilation time, or divide the source program.

2.158 jwd8701o-i

The Optimization Control Line (OCL) is disabled.

[Message Explanation]

The system does not apply the optimization for the following reasons, although an optimization control line was specified.

- It was judged that optimization was not effective to specified function, loop, or operation.
- It was not possible to optimize, because it found complex processing to specified function, loop, or operation.

[System Behavior]

Disables the Optimization-Control-Line.