

Subscribe to DeepL Pro to edit this document.  
Visit [www.DeepL.com/pro](https://www.deepl.com/pro?cta=edit-document)for more information.

CRSim Documentation

Table of Contents

[1 Overview 2](#_Toc90394034)

[2 Development Details 2](#_Toc90394035)

[2.1 simulate\_cr Function 2](#_Toc90394036)

[2.2 optimize\_cr function 4](#_Toc90394037)

# summary

### (a) simulate\_cr function:.

### Simulates multilevel C/R and outputs the results.

### (b) optimize\_cr function:.

### Find the interval and L2\_freq that maximizes the efficiency of the simulate\_cr function, and output the simulation results at that time.

# Development Details

## simulate\_cr function

### tuple simulate\_cr(interval, L2ckpt\_freq, L1ckpt\_overhead, L2ckpt\_latency, ckptRestartTimes, failRates, N, SN, G, g, alpha, check\_interval, n\_check\_ok, n\_failure\_max, efficiency\_log)

### Argument (Input)

|  |  |  |  |
| --- | --- | --- | --- |
| **argument name** | **Description.** | **type (e.g. of machine, goods, etc.)** |  |
| **Interval** | L1 Checkpoint interval | int | indispensable |
| **L2ckpt\_freq** | L2 Checkpoint Frequency | int | indispensable |
| **L1ckpt\_overhead** | Synchronization L1 Checkpoint time | int | indispensable |
| **L2ckpt\_latency** | Asynchronous L2 checkpoint time | int | indispensable |
| **ckptRestartTimes** | Array of length 2 containing the time required for L1,L2 recovery  = [L1 recovery time,L2 recovery time]. | List  [int,int]. | indispensable |
| **failRates** | Array of length 2 containing the number of failures requiring L1,L2 recovery per unit time = [L1 failure times  Number,L2 Failure Count] | List [float,float]. | indispensable |
| **N** | Total number of computation nodes | int | indispensable |
| **SN** | Number of spare nodes \*Parameters added to the specification | int | indispensable |
| **g** | L1 Checkpoint group size | int | indispensable |
| **g** | L1 Checkpoint Fault Tolerance | int | indispensable |
| **alpha** | Threshold of change in efficiency that terminates the simulation  value | float | indispensable |
| **check\_interval** | Frequency of Efficiency change checks  Parameters added to the specification | int | Optional, Default=1 |
| **n\_check\_ok** | Because it is judged to be finished by the change amount check of Efficiency  Number of consecutive times of \*Parameters added to the specification | int | Optional, Default=1 |
| **n\_failure\_max** | Maximum number of failures \*Parameters added to the specification | int | Optional, Default=500000 |
| **efficiency\_log** | Turn on/off the historical output of the Efficiency change check  Parameters added to the specification | bool | Optional, Default=False |

### Return value (output): tuple type data = (X,A,B,C,D,E,F)

|  |  |  |
| --- | --- | --- |
| **argument name** | **Description.** | **type (e.g. of machine, goods, etc.)** |
| **an unknown** | Efficiency = A/(B+C+D+F) | float |
| **A** | real computation time | float |
| **B** | Time spent in the calculation state | float |
| **c** | L1 Time spent at checkpoint | float |
| **D** | L1 Time spent in recovery | float |
| **E** | L2 Time spent on checkpoints | float |
| **f** | Time spent on L2 recovery | float |

## optimize\_cr function

### tuple optimize\_cr (L1ckpt\_overhead, L2ckpt\_latency, ckptRestartTimes, failRates, N, SN, G, g, alpha, check\_interval, n\_check\_ok, n\_failure\_max, n\_steps, log\_interval)

### Argument (Input)

|  |  |  |  |
| --- | --- | --- | --- |
| **argument name** | **Description.** | **type (e.g. of machine, goods, etc.)** |  |
| **L1ckpt\_overhead** | Synchronization L1 Checkpoint time | int | indispensable |
| **L2ckpt\_latency** | Asynchronous L2 checkpoint time | int | indispensable |
| **ckptRestartTimes** | Array of length 2 containing the time required for L1,L2 recovery  = [L1 recovery time,L2 recovery time]. | List  [int,int]. | indispensable |
| **failRates** | Array of length 2 containing the number of failures requiring L1,L2 recovery per unit time = [L1 failure times  Number,L2 Failure Count] | List [float,float]. | indispensable |
| **N** | Total number of computation nodes | int | indispensable |
| **SN** | Number of spare nodes \*Parameters added to the specification | int | indispensable |
| **g** | L1 Checkpoint group size | int | indispensable |
| **g** | L1 Checkpoint Fault Tolerance | int | indispensable |
| **alpha** | Threshold of change in efficiency that terminates the simulation  value | float | indispensable |
| **check\_interval** | Frequency of Efficiency change checks  Parameters added to the specification | int | Optional, Default=1 |
| **n\_check\_ok** | Because it is judged to be finished by the change amount check of Efficiency  Number of consecutive times of \*Parameters added to the specification | int | Optional, Default=1 |
| **n\_failure\_max** | Maximum number of failures \*Parameters added to the specification | int | Optional, Default=500000 |
| **n\_steps** | Number of optimization iterations \*Parameters added to specification | int | Optional, Default=5000 |
| **log\_interval** | Log output interval for optimization, 0 means no output  Parameters added to the specification | int | Optional, Default=100 |

### Return value (output): tuple type data=(X,A,B,C,D,E,F, interval, L2ckpt\_freq)

|  |  |  |
| --- | --- | --- |
| **argument name** | **Description.** | **type (e.g. of machine, goods, etc.)** |
| **an unknown** | Efficiency = A/(B+C+D+F) at interval, L2ckpt\_freq of optimization results | float |
| **A** | interval of optimization results, real computation time at L2ckpt\_freq | float |
| **B** | interval of optimization results, time spent in the computation state at L2ckpt\_freq | float |
| **C** | interval of optimization results, time spent on L1 checkpoint at L2ckpt\_freq | float |
| **D** | interval of optimization results, time spent for L1 recovery at L2ckpt\_freq | float |

|  |  |  |
| --- | --- | --- |
| **E** | interval of optimization results, time spent on L2 checkpoints during L2ckpt\_freq | float |
| **f** | interval of optimization results, time spent for L2 recovery at L2ckpt\_freq | float |
| **interval** | L1 checkpoint interval for optimization results | int |
| **L2ckpt\_freq** | Frequency of L2 checkpoints for optimization results | int |

### Optimization Methodology

### An annealing method was used as the optimization technique.

### Initial state

### Of the following combinations of interval and L2\_freq\_freq (24 combinations), the one with the highest efficiency is implemented as the initial state.

### interval = 1000, 2500, 5000, 8000, 12000, 24000

### L2\_freq\_freq = 1, 2, 5, 10

### State Transition

### The following four methods were considered for state transitions.

### Method 1.

### 1. randomly select which value of interval or L2ckpt\_freq to change 2. increase/decrease the selected parameter by 2%.

### Method 2.

### 1. randomly select which value of interval or L2ckpt\_freq to change 2. increase or decrease the selected parameter by a random value within 5%.

### Method 3

### 1. increase/decrease both interval and L2ckpt\_freq by a random value within 0-5%.

### Method 4

### 1. randomly select which value of interval or L2ckpt\_freq to change 2. increase/decrease the selected parameter by a fixed value

### As a result of the study, Method 1 was adopted because none of the methods showed much difference except for Method 4 (\*).

### Because the interval has a wide range, when increasing or decreasing it by a fixed value, a small value causes too many times to move within the range, while a large value causes too large a change on the small side.

### The above state transition methods can be changed to any of the above methods with a simple source code modification. The 2% and 5% numbers can also be changed only by modifying the corresponding parts of the source code.