循环优化系列第八讲

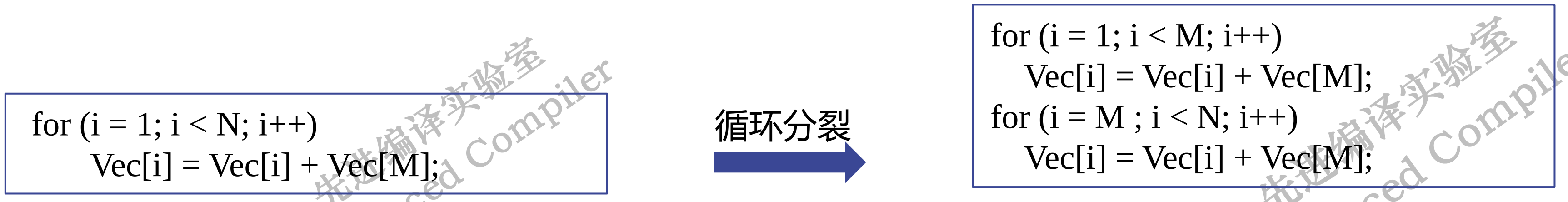
循环分裂

嘉宾：柴赟达

循环分裂

- **基础概念**

循环分裂是将循环的迭代次数拆成两段或者多段，拆分后的循环不存在主体循环之说，也就是拆分成迭代次数都比较多的两个或者多个循环

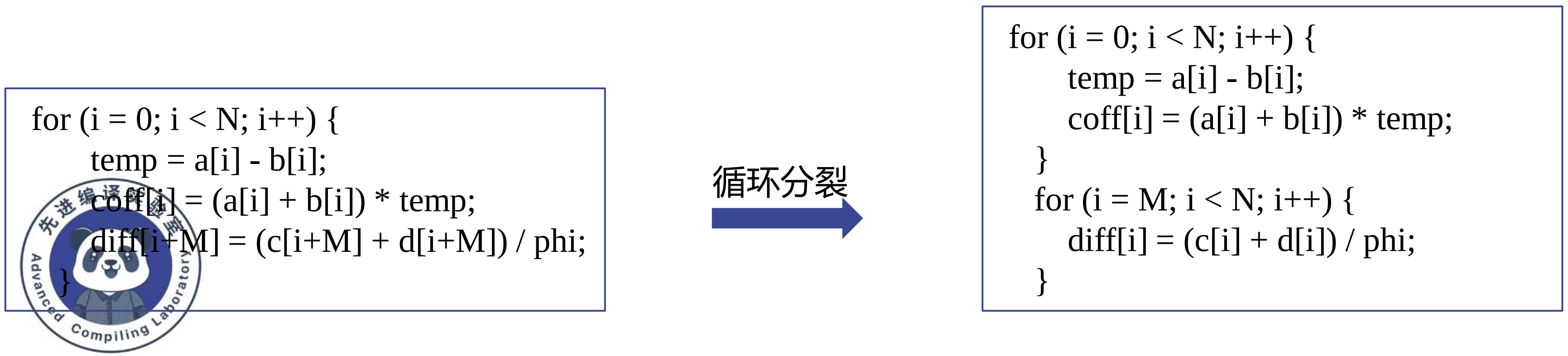


```
for (i = 1; i < N; i++)
      Vec[i] = Vec[i] + Vec[M];
```

循环分裂

```
for (i = 1; i < M; i++)
   Vec[i] = Vec[i] + Vec[M];
for (i = M ; i < N; i++)
   Vec[i] = Vec[i] + Vec[M];
```

- **循环分裂的有利性**



```
for (i = 0; i < N; i++) {
      temp = a[i] - b[i];
      coff[i] = (a[i] + b[i]) * temp;
      diff[i+M] = (c[i+M] + d[i+M]) / phi;
  }
```

循环分裂

```
for (i = 0; i < N; i++) {
      temp = a[i] - b[i];
      coff[i] = (a[i] + b[i]) * temp;
   }
   for (i = M; i < N; i++) {
      diff[i] = (c[i] + d[i]) / phi;
   }
```

循环分裂

- **优化效果**

```
#include <stdio.h>
#define N 1024
int main()
{
    int Vec[N];
    int i, M = 512;
    for (i = 0; i < N; i++) {
        Vec[i] = i;
    }
    for (i = 0; i < N; i++)
        Vec[i] = Vec[i] + Vec[M];
    printf("%d \n",Vec[3]);
    return 0;
}
```

gcc 1.c -O3 -fopt-info

循环分裂

```
#include <stdio.h>
#define N 1024
int main()
{
    int Vec[N],A[N];
    int i;
    int  M = 512;
    for (i = 0; i < N; i++) {
        Vec[i] = i;
    }
    for (i = 0; i < M; i++)
        Vec[i] = Vec[i] + Vec[M];
    for (i = M ; i < N; i++)
        Vec[i] = Vec[i] + Vec[M];

    printf("%d \n",Vec[3]);
    return 0;
}
```

gcc 1-opt.c -fopt-info -O3

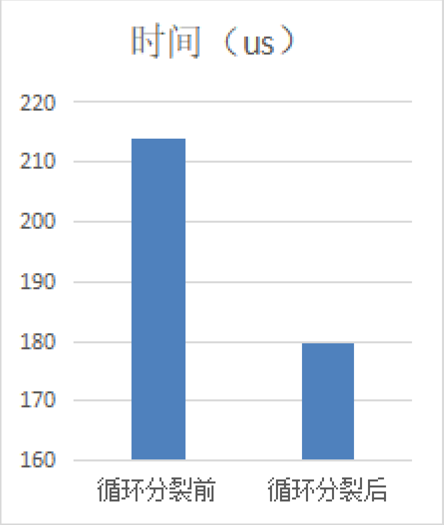
循环分裂

- **优化效果**

```
#include <stdio.h>
#include <sys/time.h>
#define N 8192
int main() {
  int i, temp, phi;
  int M = 4096;
  int a[N], b[N], c[N], d[N], coff[N], diff[N];
  struct timeval  time_start, time_end;
  temp = 2;
  phi = 2;
  for (i = 0; i < N; i++) {
    a[i] = i;
    b[i] = i + 1;
    c[i] = i + 2;
    d[i] = i + 3;
  }
  gettimeofday(&time_start, NULL);
  for (i = 0; i < N; i++) {
    temp = a[i] - b[i];
    coff[i] = (a[i] + b[i]) * temp;
    diff[i+M] = (c[i+M] + d[i+M]) / phi;
  }
  gettimeofday(&time_end, NULL);
  printf("used time %ld us\n", time_end.tv_usec -
time_start.tv_usec);
}
```

循环分裂

```
#include <stdio.h>
#include <sys/time.h>
#define N 8192
int main() {
int i, temp, phi;
int M = 4096;
    int a[N], b[N], c[N], d[N], coff[N], diff[N];
    struct timeval  time_start, time_end;
    temp = 2;
    phi = 2;
    for (i = 0; i < N; i++) {
        a[i] = i;
        b[i] = i + 1;
        c[i] = i + 2;
        d[i] = i + 3;
    }
    gettimeofday(&time_start, NULL);
    for (i = 0; i < N; i++) {
        temp = a[i] - b[i];
        coff[i] = (a[i] + b[i]) * temp;
    }
    for (i = M; i < N; i++) {
        diff[i] = (c[i] + d[i]) / phi;
    }
    gettimeofday(&time_end, NULL);
    printf("used time %ld us\n", time_end.tv_usec -
time_start.tv_usec);
}
```



分享完毕，感谢聆听！

参考文献：

[1] Optimizing Compilers for Modern Architectures: A Dependence-Based Approach [Book Review][J]. Computer,2002,35(4).

[2]