I HC QUC GIA THỊNH PH H CHỜ MINH
TRNG I HC BÒCH KHOA
KHOA KHOA HC VỊ K THUT MÒY TỜNH

H iu hịnh

Bòo còo Lab 3

Tờnh xp x s Pi dứng Single-threaded, Multi-threaded vị Shared Variable Program

---

Giòo viển hng dn:	Hoịng Lổ Hi Thanh	
Sinh viển thc hin:	Mai Xuón Nht	23xxxxx
	Nguyn Thòi Sn	2312968
	Lổ c Tịi	2312995

THỊNH PH H CHỜ MINH, 01/12/2025

Mc lc

1 Approach 1: A Single-Thread program

```
    unsigned long long int count_point_inside
    (unsigned long long int npoints, unsigned char *img){
    unsigned long long inside = 0ULL;
    for (unsigned long long i = 0; i < npoints; i++) {
        double x = -1.0 + (double)rand() /
        (double)RAND_MAX * 2.0; // [-1,1]
        double y = -1.0 + (double)rand() /
        (double)RAND_MAX * 2.0; // [-1,1]
        if (x * x + y * y <= 1.0) inside++;
        int px = (int)((x + 1.0) * 0.5 * (W - 1));
        int py = (int)((y + 1.0) * 0.5 * (H - 1));
        if (px >= 0 && px < W && py >= 0 && py < H) {
            int idx = (py * W + px) * 3;
            double r2 = x*x + y*y;
            img[idx + 0] = (r2 <= 1.0) ? 0 : 255;    // R
            img[idx + 1] = (r2 <= 1.0) ? 255 : 0; // G
            img[idx + 2] = (r2 <= 1.0) ? 0 : 0;    // B
        }
    }
    return inside;
}
```

2 Approach 3: Shares Variables

2.1 Mc tiổu

2 còch trc lị **Approach 1: Separate Variables** vị **Approach 2: Shared Variables**, chững ta õ tơm hiu v còch s dng còc bin riổng bit vị chia s bin trong còc mũ hơnh hc mòy. Trong phn nịy, chững ta s khòm phò còch tip cn th ba, ú lị **Shares Variables** u s dng bin cc b m s im ri vịo hơnh trùn ri cng dn vịo tng khi kt thữc.
Trong còch 3 yổu cu:

- S dng bin toịn cc lu tr s im ri vịo hơnh trùn
- Còc lung (thread) **cp nht trc tip** bin toịn cc nịy mi khi lung thc thi kt thữc.
- S dng c ch **mutex locks** trònh tơnh trng **Race Condition** khi nhiu lung cứng truy cp vị cp nht bin toịn cc.

2.2 Thut toòn

Trong phiổn bn a lung vi bin chia s:

- Chng trơnh to `nThreads` lung.
- Mi lung c gòn mt on ch s im $[start\_idx, end\_idx)$ khũng trứng nhau:
  - $$\texttt{start\_idx} = \texttt{thread\_id} \cdot \frac{\texttt{nPoints}}{\texttt{nThreads}}$$
  - $$\texttt{end\_idx} = \begin{cases} \texttt{start\_idx} + \frac{\texttt{nPoints}}{\texttt{nThreads}} & \text{nu khũng phi lung cui,} \\ \texttt{nPoints} & \text{nu lị lung cui.} \end{cases}$$
- Mi im c sinh da trổn mt seed ph thuc vịo ch s toịn cc:
$$\texttt{seed} = \texttt{base\_seed} + i$$
- S dng mutex locks trònh race condition khi cp nht bin toịn cc.

2.3 Trònh race condition bng mutex

Do `global_count` lị bin chia s, nu nhiu lung cứng thc hin:

```
global_count++;
```

mị khũng cú c ch ng b, chng trơnh s gp *race condition*: còc lung c cứng mt giò tr c, cứng tng ri ghi ố ln nhau, khin giò tr cui cứng ca `global_count` b thiu hoc sai. iu nịy dn ti kt qu xp x π b sai lch.

trònh hin tng ú, mi ln cp nht `global_count` phi c bo v bi mutex:

```
pthread_mutex_lock(&lock);
global_count++;
pthread_mutex_unlock(&lock);
```

Nh vy, ti mi thi im ch cú mt lung c phỗp vịo vứng cp nht, m bo kt qu cui cứng lị ũng.

2.4 xut gii phòp

Mt nhc im ca còch tip cn 3 lị:

- Mi ln cú im ri vịo hơnh trùn, lung phi thc hin thao tòc lock and unlock mutex
- Vi s im ln lổn n $10^7, 10^8, 10^9$ vị nhiu lung =, thơ vic lock/ unlock rt ln dn n overhead nng n

gim overhead mị vn gi ỳ tng s dng bin chia s `global_count`, cú th xut gii phòp:

- Mi lung s dng mt bin m cc b `local_count`.
- Trong vùng lp, lung ch tng `local_count` (khũng lock mutex liổn tc).
- Sau khi hoịn thịnh tt c còc im c gòn, lung mi:

  ```
  pthread_mutex_lock(&lock);
  global_count += local_count;
  pthread_mutex_unlock(&lock);
  ```

- Nh vy, mi lung ch lock/unlock **mt ln** thay vơ hịng triu ln, gim òng k chi phờ ng b.

Còch lịm nịy vn gi ũng yổu cu dứng bin chia s `global_count`, ng thi:

- Trònh race condition nh mutex.
- Gim overhead ng b, giữp Còch 3 tin gn hn v hiu nng so vi Còch 2.

2.4.1 iu chnh c ch sinh s ngu nhiổn tng chờnh xòc

Trong bịi toòn Monte Carlo, cht lng vị còch s dng s ngu nhiổn nh hng trc tip n:

- **n nh** ca kt qu gia còc ln chy.
- **chờnh xòc** ca giò tr xp x π khi s im `nPoints` tng lổn.

S dng seed c nh vị ph thuc vịo ch s toịn cc
Thay vơ khi to seed t thi gian h thng (`time(NULL)`), nhúm s dng mt *seed gc* c nh `base_seed` vị xóy dng seed cho tng im da trổn ch s toịn cc i:

```
unsigned int seed = base_seed + (unsigned int)i;
```

Nh ú:

- Vi mi im s cú 1 seed duy nht, giữp phón tòn tt hn còc im ngu nhiổn.
- Vi cứng giò tr `nPoints`, mi ch s i luũn to ra cứng mt cp (x,y), khũng ph thuc vịo s lung hay còch phón chia cũng vic.
- Tp còc im ngu nhiổn c sinh ra trong phiổn bn n lung vị a lung lị nh nhau (ch khòc còch phón chia gia còc lung).
- Chy li chng trơnh nhiu ln vi cứng cu hơnh luũn cho cứng mt kt qu xp x π, giữp vic so sònh, o *speedup* vị phón tòch tr nổn òng tin cy hn.

Nh vic "lịm mm" c ch ngu nhiổn theo hng *deterministic nhng vn phón tòn*, nhúm va m bo c tònh ngu nhiổn cho thut toòn Monte Carlo, va m bo c tònh lp li vị chờnh xòc cao cho kt qu thc nghim.