Parallel computing platforms

Jeremy Iverson

College of Saint Benedict & Saint John's University

plan for the day

- Naive Bayes assignment updates (40min + 5min break)
- parallel computer organization (40min + 5min break)
- classes of parallel computation (40min + 5min break)
- our first parallel algorithms (40min + 5min break)

Naive Bayes in C

parallel computer organization

parallel computer organization

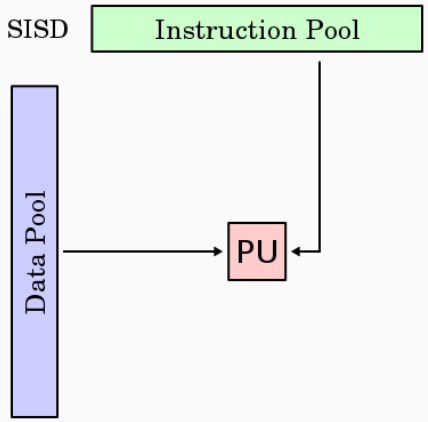
- control mechanism

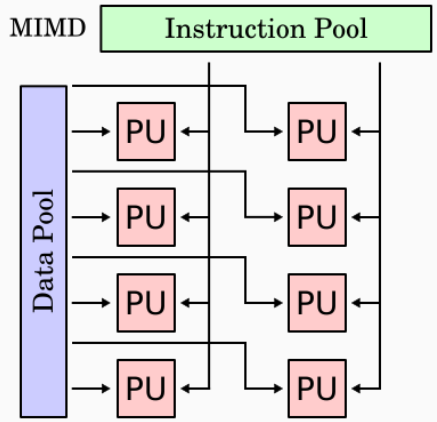
parallel computer organization

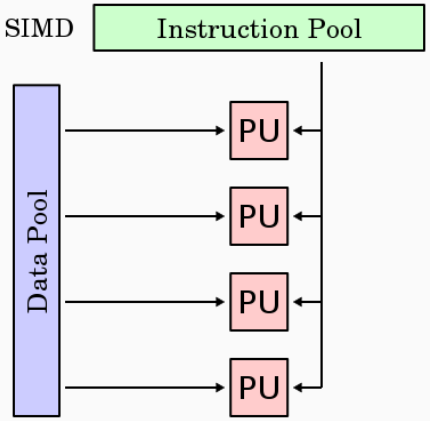
- control mechanism
- communication model

flynn's taxonomy

- based on the number of instruction streams and data streams available in the architecture

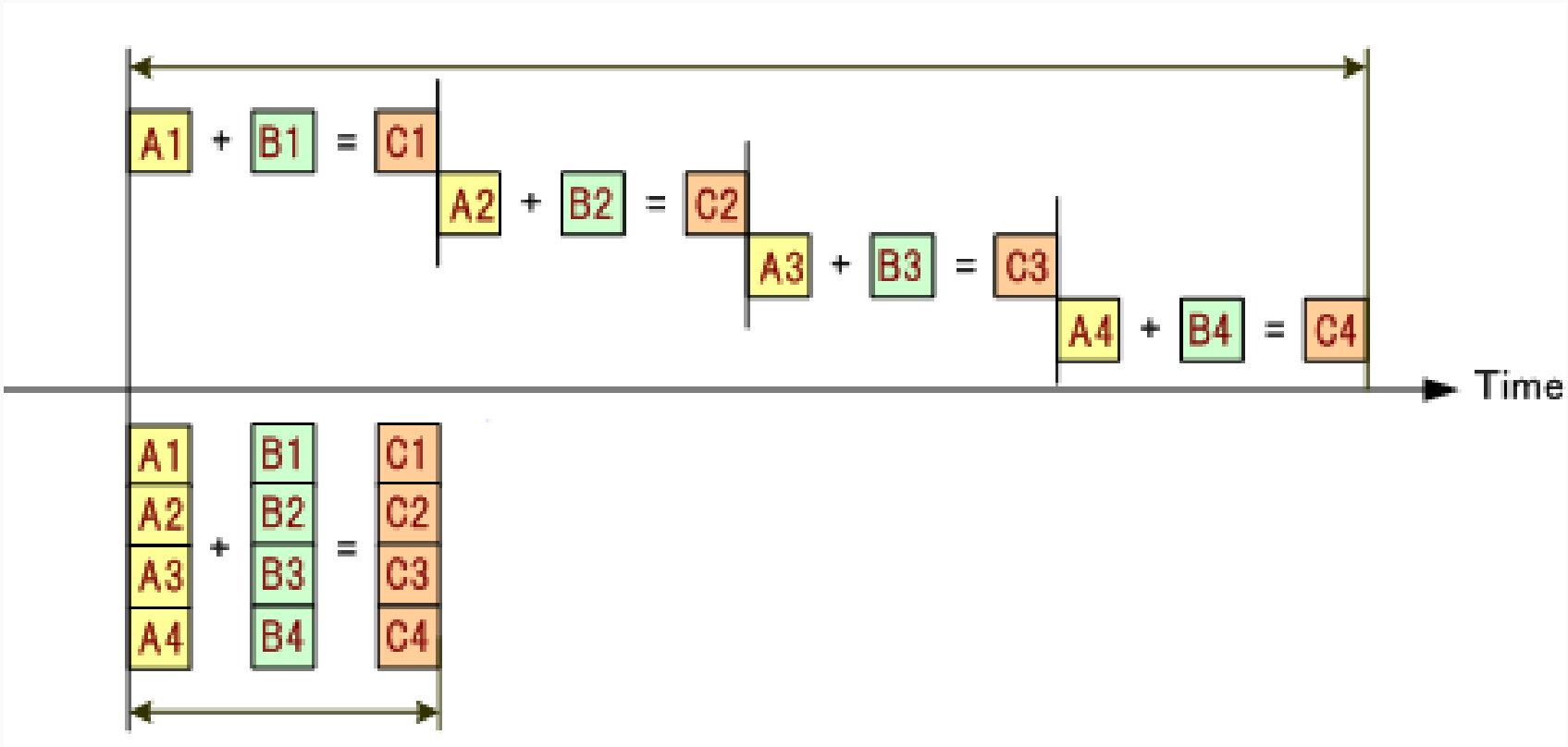






Flynn's taxonomy by Cburnett / CC BY 3.0 / presenting the four together

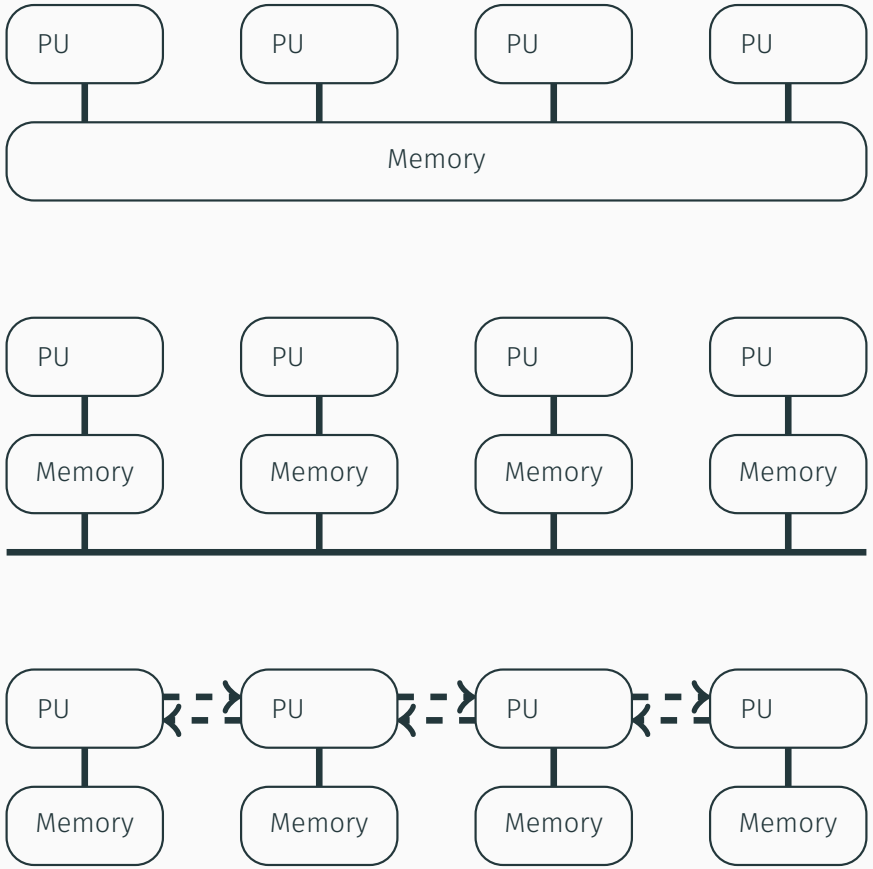
simd



SIMD / cropped from original

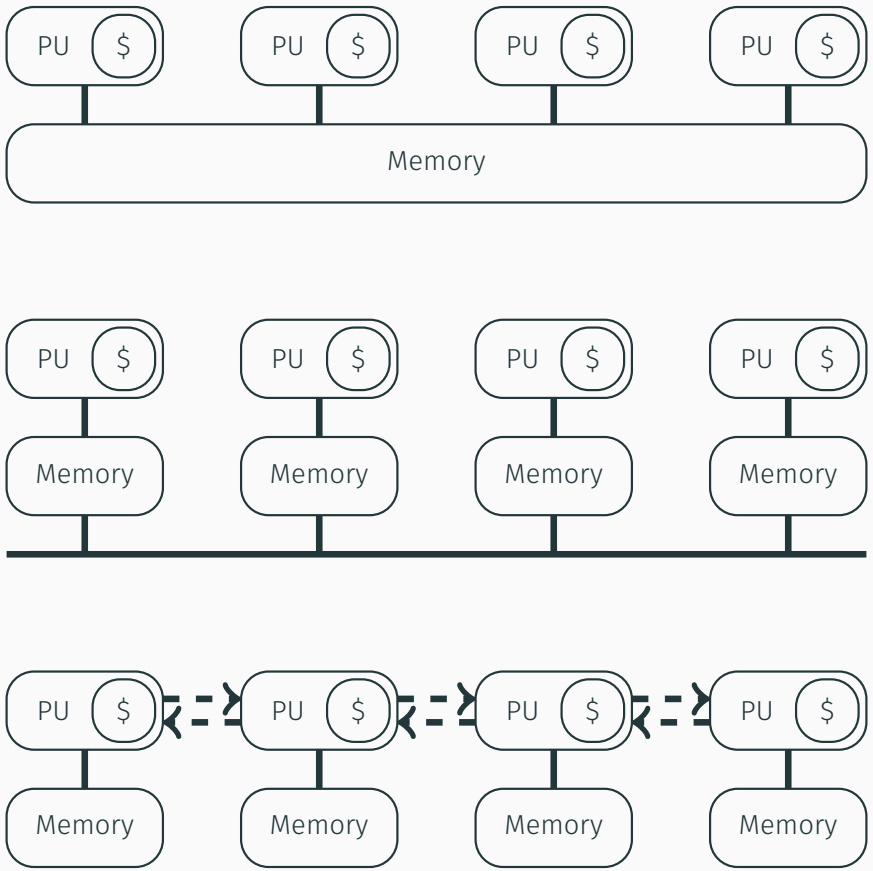
communication models

- shared-address space
- message-passing



communication models

- shared-address space
- message-passing



classes of parallel computation

classes of parallel computation

Imagine you are the head chef responsible for preparing a wedding banquet. The meal will have four courses: appetizer, salad, main course, and dessert. You have P chefs at under your supervision.

classes of parallel computation

Imagine you are the head chef responsible for preparing a wedding banquet. The meal will have four courses: appetizer, salad, main course, and dessert. You have P chefs at under your supervision.

How best to get the meal served to the guests as quickly as possible?

classes of parallel computation

Imagine you are the head chef responsible for preparing a wedding banquet. The meal will have four courses: appetizer, salad, main course, and dessert. You have P chefs at under your supervision.

How best to get the meal served to the guests as quickly as possible?

- Each chef works on N/P meals independently of the others.
- Each chef works on a different task related to the preparation of a meal, i.e., cutting carrots, cooking soup, icing cake, etc.

classes of parallel computation

Imagine you are the head chef responsible for preparing a wedding banquet. The meal will have four courses: appetizer, salad, main course, and dessert. You have P chefs at under your supervision.

How best to get the meal served to the guests as quickly as possible?

- *data parallelism*
- *task parallelism*

y := a*x + y

```
1  void
2  saxpy(size_t const n,
3        float  const a,
4        float  const * const x,
5        float        * const y)
6  {
7    for (size_t i = 0; i < n; i++) {
8      y[i] = a * x[i] + y[i];
9    }
10 }
```

reduction

```
1  float
2  min(size_t const n,
3      float * const x,
4  {
5    /* x[0] will always contain the current
6     * minimum */
7    for (size_t i = 0; i < n; i++) {
8      if (x[i] < x[0]) {
9        x[0] = x[i];
10     }
11   }
12   return x[0];
13 }
```

reduction cont'd

```
1  float
2  min(size_t const n,
3      float * const x,
4  {
5    for (size_t i = 1; i < n; i *= 2) {
6      /* x[j] will always contain the current
7       * minimum in interval [j,j+2i)*/
8      for (size_t j = 0; j < n; j += 2 * i) {
9        if (x[j + i] < x[j]) {
10         x[j] = x[j + i];
11       }
12     }
13   }
14   return x[0];
15 }
```

sorting

```
count = array of k+1 zeros
for x in input do
    count[x] += 1

total = 0
for i in 0, 1, ... k do
    counti = count[i]
    count[i] = total
    total += counti

for x in input do
    output[count[x]] = x
    count[x] += 1

return output
```