

La programmation fonctionnelle et Scala

Foudil BRÉTEL

CC-IN2P3
<http://cc.in2p3.fr>

2014-10-15

Definition (Von Neumann bottleneck)

Pour accomplir une tâche, le programmeur pense d'abord en termes de cases mémoire: variables. Programmer c'est bâtir avec des **affectation**, **déclarations (statements)**.

cf. John Backus, *Can Programming Be Liberated from the von. Neumann Style?*, Turing Award Lecture 1978.

- Décrire des abstractions (polynomes?)
- **Fonctions** associent des ensembles, mais ne changent pas les éléments

Definition (Programmation fonctionnelle)

- éviter variables mutables, affectations, boucles \Rightarrow récursion
 - fonctions comme **objets de premier ordre** pouvant être produits, consommés et composés.
-
- Exemples
Lisp, XSLT, Haskell, Ruby, JavaScript

```
class Person(val name: String, val age: Int)

val people: Array[Person] = new Array(2)
people(0) = new Person("Max", 16)
people(1) = new Person("Joe", 22)

val (minors, adults) = people partition (_.age < 18)
// val (minors, adults) = people.par partition (_.age < 18)
```

```
import java.util.ArrayList;

public class Person {
    public final String name ;
    public final int age ;
    Person(String name, int age) {
        this.name = name;
        this.age = age;
    }
}

Person[] people = new Person[2];
people[0] = new Person("Max", 16);
people[1] = new Person("Joe", 22);

Person[] minors;
Person[] adults;

{ ArrayList<Person> minorsList = new ArrayList<Person>();
  ArrayList<Person> adultsList = new ArrayList<Person>();

  for (int i = 0; i < people.length ; i++)
      (people[i]. age < 18 ? minorsList : adultsList)
          .add(people [i]);

  minors = minorsList.toArray(people);
  adults = adultsList.toArray(people);
}
```

- Caractéristiques
 - **unification** Fonctionnel et Orienté-Objets
 - concis, expressif, très riche
 - typage fort + inférence de type
 - langage «académique»: bien pensé, fruit de longues recherches
 - jvm \Rightarrow interopérable avec Java
- Vecteurs d'adoption
 - plateformes web (backend twitter)
 - plateforme de trading (comme ocaml)
 - modélisation financière (comme ocaml)
 - simulation

Les signes d'une renaissance

- C++11, Java8, Swift, Closure, MapReduce, Spark...
- Le parallélisme, une nécessité

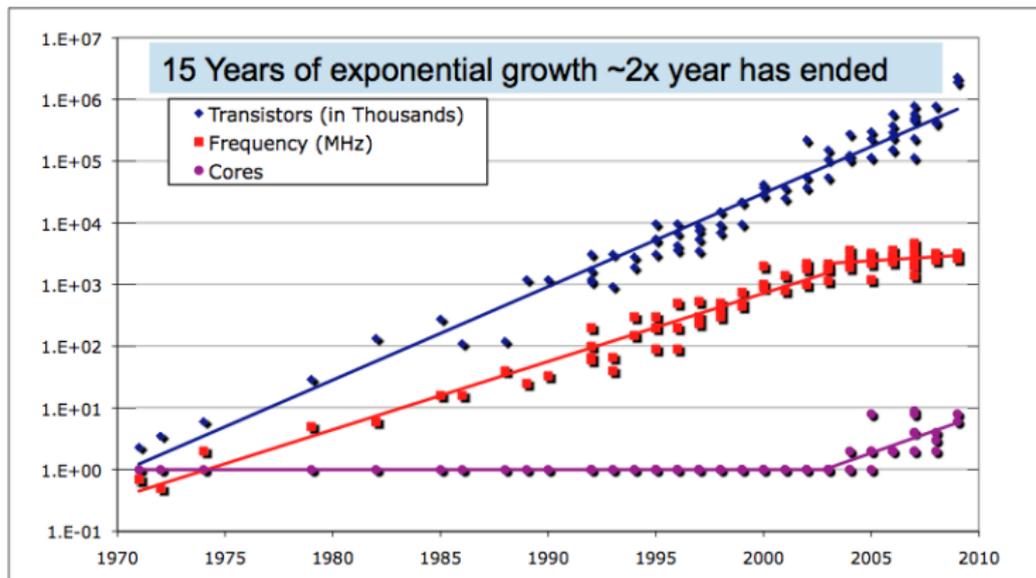


Figure: Loi de Moore ?

Definition (Parallélisme vs. Concurrency)

- Parallélisme: une tâche répartie
- Concurrency: plusieurs tâches qui collaborent