

Master 2 Informatique Fondamentale et Appliquée Année 2023-2024

Carole Delporte, Cristina Sirangelo, Ralf Treinen

14/12/2023

Calendrier 23/24

- Deuxième période : du 08/01 au 22/03 (cours/TD/TP)
Examens : 25/03 au 29/03
- Troisième période : du 01/04 au 30/09 (stage)
Soutenance de stage, et jury du M2 : mi-septembre

Rappel : Vous devrez valider 60 crédits

- 11 modules d'informatique (chacun à 3 crédits), modules à choisir selon les règles spécifiques de votre parcours.
- le module d'Anglais, première période (3 crédits)
- le stage en entreprise pendant la troisième période (24 crédits)
- Dans les trois parcours on peut aussi choisir un cours "externe" (par ex au MPRI, LMFI, une autre université) *avec accord du responsable de parcours.*

Les règles du parcours DATA

Anglais + 11 cours d'Info, dont au moins 7 *cours fondamentaux* :

1ère période	2ème période
Architecture des Systèmes de Bases de Données	Architecture des Systèmes d'Information
Base de données spécialisées	Algorithmique Répartie
Fouille de Données et Aide à la Décision	Méthodes Algorithmiques pour l'Accès à l'Information Numérique
Programmation Objets : Concepts Avancées	Programmation Logique et Par Contraintes Avancée
	Grands Réseaux d'Interaction
	Optimisation
	Programmation Répartie
	Introduction Apprentissage Profond

Les règles du parcours IMPAIRS

Anglais + 11 cours d'Info, dont au moins 8 *cours suggérés* :

1ère période	2ème période
Architecture des Systèmes de BD	Architecture des SI
Fouille de données et aide à la décision	Méthodes algorithmiques pour l'accès à l'information numérique
Informatique embarquée	Administration système et réseaux
Ingénierie des protocoles	Algorithmique répartie
Modélisation et Spécification	Programmation répartie
Programmation Synchrone	Grands réseaux d'interaction
Protocoles des services Internet	Mobilité
	Interfaces et Outils de MacOS-X
	Systèmes Avancés

Les règles du parcours LP

Anglais + 11 cours d'Info, dont 2 *cours obligatoires* :

1ère période	2ème période
Programmation Objets : Concepts Avancées	Programmation Comparée

et au moins 6 parmi les *cours recommandés* :

1ère période	2ème période
Architecture des Systèmes de BD	Transformation de Programmes
Méthodes Formelles de Vérification	Programmation Logique et Par Contraintes Avancée
Modélisation et Spécification	Programmation Répartie
Programmation Synchrone	Analyse Statique
	Interfaces et Outils de MacOS-X

Inscriptions pédagogiques à des modules

- Vous choisissez sur silice les cours que vous voulez suivre.
- Sauf les redoublants : email au responsable de parcours.
- Vous choisissez 6 cours pour la deuxième période.
- Si vous souhaitez un cours en plus : envoyer un mail à votre responsable de parcours et on va essayer de vous y inscrire (sans promesse).
- Il n'est pas permis de vous inscrire à des cours qui ont lieu au même moment (OK pour Mobilité + Transformation de Programmes).
- Pour la deuxième période : du 15/12 le matin au 21/12.
- Les cours ont des capacités limitées. La priorité sera donnée aux étudiants pour qui le cours demandé est recommandé.

Le Stage

- Durée minimale de 4 mois.
- Durée maximale de 6 mois (contrainte légale)
- Entre le 1 avril et le 30 septembre.
- Rapport de stage, soutenance (normalement publique) en septembre.
- Stage en entreprise (aussi startup) ou Stage de recherche.
- Stage à l'étranger possible, mais procédure administrative particulière.

Trouver son Stage

- **Le sujet doit être approuvé par le responsable du parcours.**
- Rédaction et signature de la convention : voir http://www.informatique.univ-paris-diderot.fr/formations/masters/m2_stages_fin_etudes
- La signature de la convention prend du temps (en particulier pour des organismes sensibles à la sécurité).
- Commencez tôt (pendant la première période) à vous en occuper.
- Nous transmettons des offres de stages sur la liste de diffusion des parcours concernés ; vérifiez votre adresse email renseignée sur silice.

Informations et contacts

- Le wiki de l'UFR :
`http://www.informatique.univ-paris-diderot.fr`
- Serveur pédagogique : `https://silice.informatique.univ-paris-diderot.fr/ufr`
- Suivi administratif : Sylvia CROCHET
Lu, Je: bureau Sophie-Germain 3002
Ma, Ve: `crochet@informatique.univ-paris-diderot.fr`
- Responsables de parcours : prendre rendez-vous par email
 - DATA : Cristina.Sirangelo `Cristina.Sirangelo@irif.fr`
 - IMPAIRS : Carole Delporte `Carole.Delporte@irif.fr`
 - LP : Ralf Treinen `treinen@irif.fr`
- Ingénieur : Laurent Pietroni
`pietroni@informatique.univ-paris-diderot.fr`

EXAMEN S3 lun. 15 janv. 24											
Lundi 15/01/2024		Mardi 16/01/2024		Mercredi 17/01/2024		Jeudi 18/01/2024		Vendredi 19/01/2024			
08h00											
08h30											
09h00		MAFAE2-IFECY100-CMTD7 Interfaces et outils de MacOS-X CMTD7 Germain - 2027 - 25 08h30 - 10h30 CMTD7 YUNESJEAN BAPTISTE M2 IMPAIRS groupe	MAFAM2-IFMCY060-CMTD7 Methodes formelles approche probabiliste CMTD7 Germain - 1001 - 40 08h30 - 11h30 CMTD7 LAROUSSINIE FRANCOIS M2 DATA groupe M2 MIC groupe M2 IMPAIRS groupe M2 LP groupe M2 MIDS groupe	MAFAE2-IFECY170-CMTD7 Systemes avances CMTD7 Germain - 2027 - 25 08h30 - 11h30 CMTD7 YUNESJEAN BAPTISTE M2 DATA groupe M2 IMPAIRS groupe M2 LP groupe	MAFAE2-IFECY030-CM Algorithmique repartie CM Halle - 234C - 84 08h30 - 11h30 CM BLIN LÉLIA M2 DATA groupe M2 GENIAL groupe M2 IMPAIRS groupe M2 LP groupe M2 MIDS groupe	MAFAE2-IFECY120-CM Mobilité CM Halle - 279F - 52 09h00 - 12h00 CM STEHLIK MATEJ M2 DATA groupe M2 IMPAIRS groupe M2 LP groupe	MAFAE2-IFECY110-CMTD7 Methodes algorithmiques pour l'accès a l'informatique numerique CMTD7 Germain - 2031 - 24 Germain - 2032 - 23				
09h30											
10h00	MAFAG2-IFGCY040-CM Transformation de programmes CM Germain - 2003 - 28 10h00 - 13h00 CM MANZONETTO GIULIO M2 DATA groupe M2 IMPAIRS groupe M2 LP groupe	MAFAE2-IFECY100-CMTD7 Interfaces et outils de MacOS-X CMTD7 Germain - 2027 - 25 10h30 - 12h30 CMTD7 YUNESJEAN BAPTISTE M2 IMPAIRS groupe									
10h30											
11h00											
11h30											
12h00											
12h30											
13h00		MAFAN2-IFGCU280-CMTD7 Introduction a l'apprentissage profond CMTD7 Germain - 0011 - 60 13h00 - 15h00 CM PERIFEL SYLVAIN M2 IMPAIRS groupe M2 DATA groupe M2 LP groupe		MAFAE2-IFECY110-CM Methodes algorithmiques pour l'accès a l'informatique numerique CM Germain - 0011 - 60 13h15 - 14h45 CM PERIFEL SYLVAIN M2 DATA groupe M2 IMPAIRS groupe	MAFAE2-IFECY070-CM Grands reseaux d'interaction CM Halle - 234C - 84 12h45 - 14h45 CM RAVELOMANANA M2 DATA groupe M2 GENIAL groupe M2 IMPAIRS groupe M2 LP groupe	MAFAE2-IFECY070-CMTD7 Grands reseaux d'interaction CMTD7 Germain - 2031 - 24 Germain - 2032 - 23 12h45 - 14h45 TP RAVELOMANANA M2 MIDS groupe M2 LP groupe M2 IMPAIRS groupe					
13h30											
14h00	MAFAE2-IFECY020-CM Administration Systeme et Reseaux CM Germain - 2027 - 25 14h00 - 17h00 CM DAGAND PIERRE-EVARISTE M2 DATA groupe M2 IMPAIRS groupe M2 LP groupe		MAFAG2-IFGCY130-CM Typage et analyse statique CM Germain - 2003 - 28 14h15 - 17h15 CM BERNARDI GIOVANNI M2 DATA groupe M2 IMPAIRS groupe M2 LP groupe								
14h30											
15h00		MAFAN2-IFNFCY280-CMTD7 Introduction a l'apprentissage profond CMTD7 Germain - 2032 - 23 15h00 - 16h00 TP PETRISAN-PORUMBEL DANIELA LUANA	MAFAN2-IFNFCY090-CM Optimisation CM Halle - 279F - 52 15h00 - 16h30 CM VIGER FABIEN M2 DATA groupe	MAFAG2-IFGCY110-CM Programmation Logique et par contraintes avancee CM Germain - 1001 - 40	MAFAE2-IFECY140-CM Programmation repartie CM Halle - 3B - 132 15h00 - 18h00 CM GALLET DELPORTE CAROLE M2 DATA groupe M2 GENIAL groupe M2 IMPAIRS groupe M2 LP groupe M2 MIDS groupe	MAFAG2-IFGCY030-CM Programmation comparee CM Halle - 279F - 52 15h00 - 18h00 CM FEREE HUGO GUATTO ADRIEN M2 GENIAL groupe M2 LP groupe	MAFAN2-IFNFCY050-CMTD Architecture des systemes d'information CMTD7 Halle - 247E - 84 15h00 - 18h00 CMTD7 FUCHS EMMANUEL M2 DATA groupe M2 IMPAIRS groupe M2 LP groupe				
15h30				MAFAN2-IFNFCY090-CMTD7 Optimisation CMTD7 Germain - 2036 - 38 Germain - 2031 - 24 16h30 - 18h30 CMTD7 VIGER FABIEN M2 DATA groupe M2 IMPAIRS groupe				MAFAG2-IFGCY110-CM Programmation Logique et par contraintes avancee CM Germain - 2003 - 28			
16h00											
16h30											
17h00											
17h30											
18h00											
18h30											
19h00											
19h30											
20h00											
20h30											
21h00											
21h30											

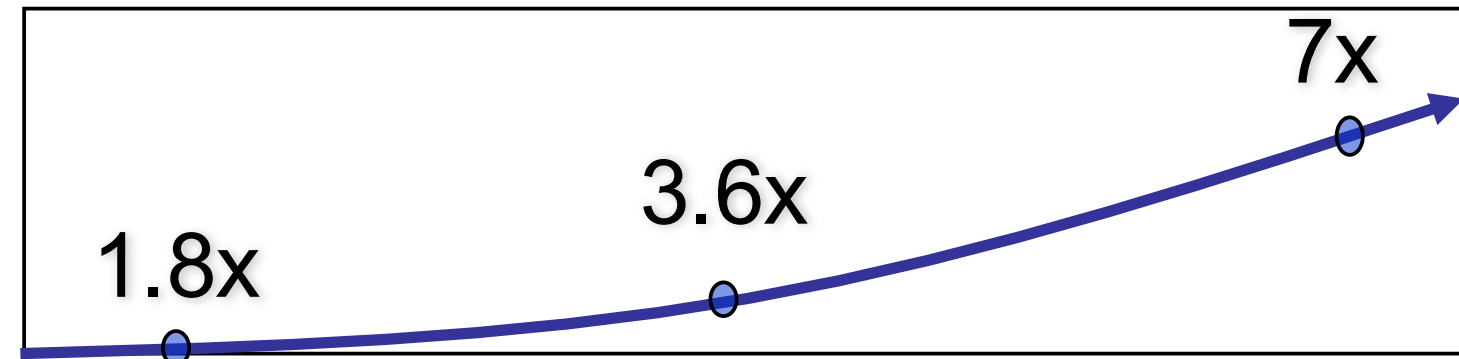
Programmation répartie

Jeudi 15h00-18h00

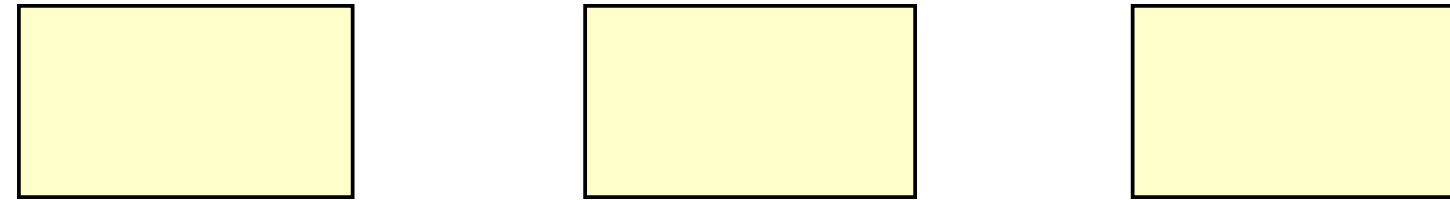
Attention premier cours Gouges 1 et les suivants 3B

Carole Delporte cd@irif.fr

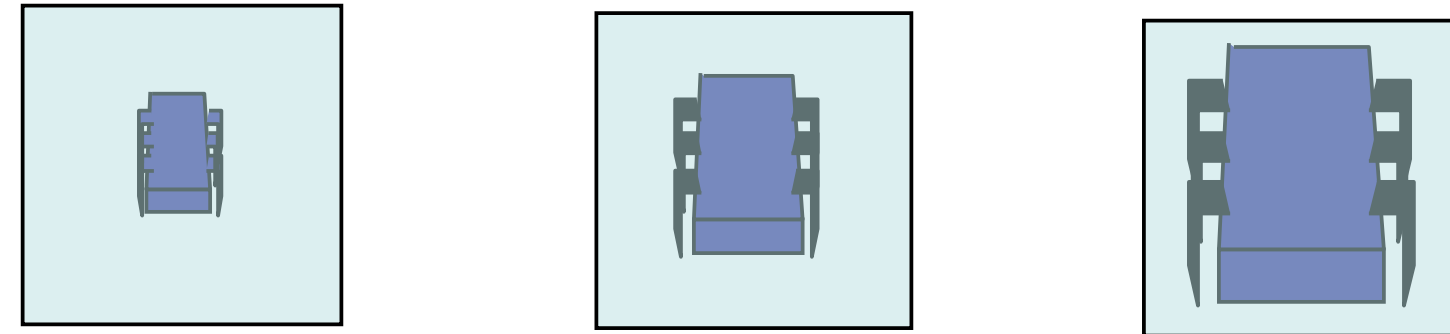
Accélération



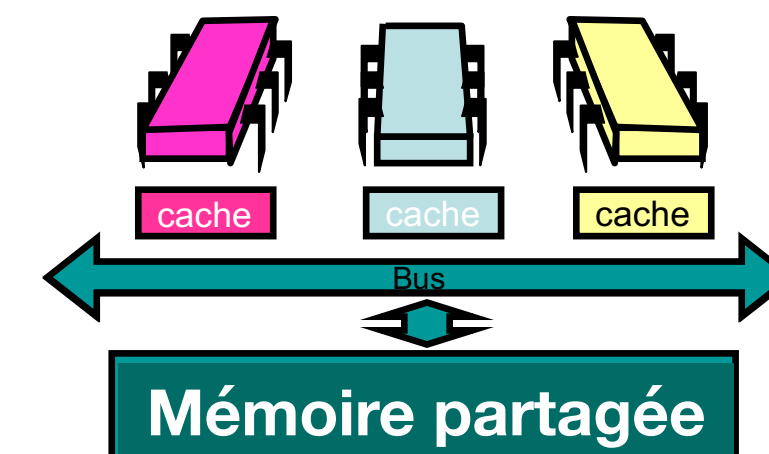
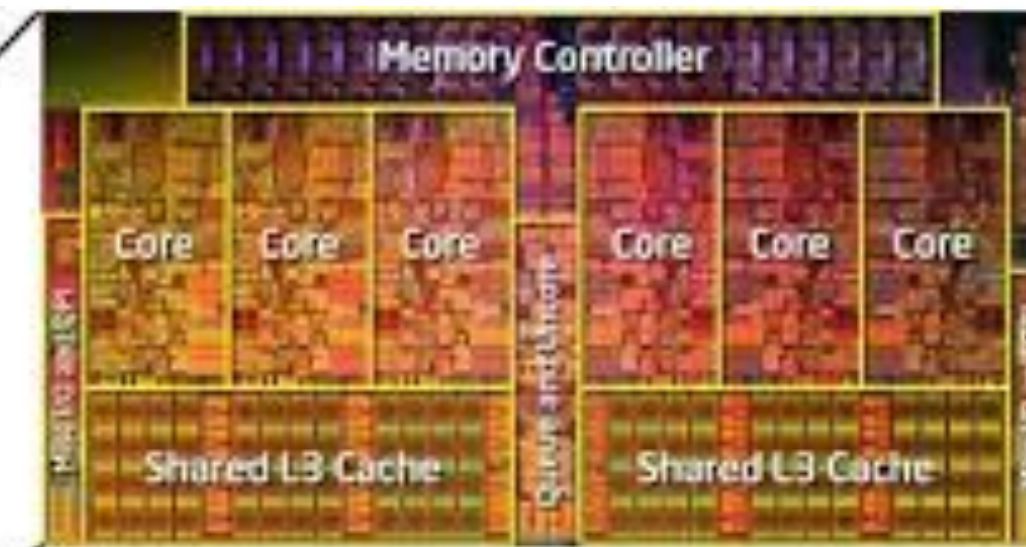
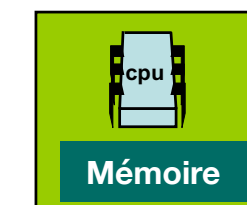
Code utilisateur

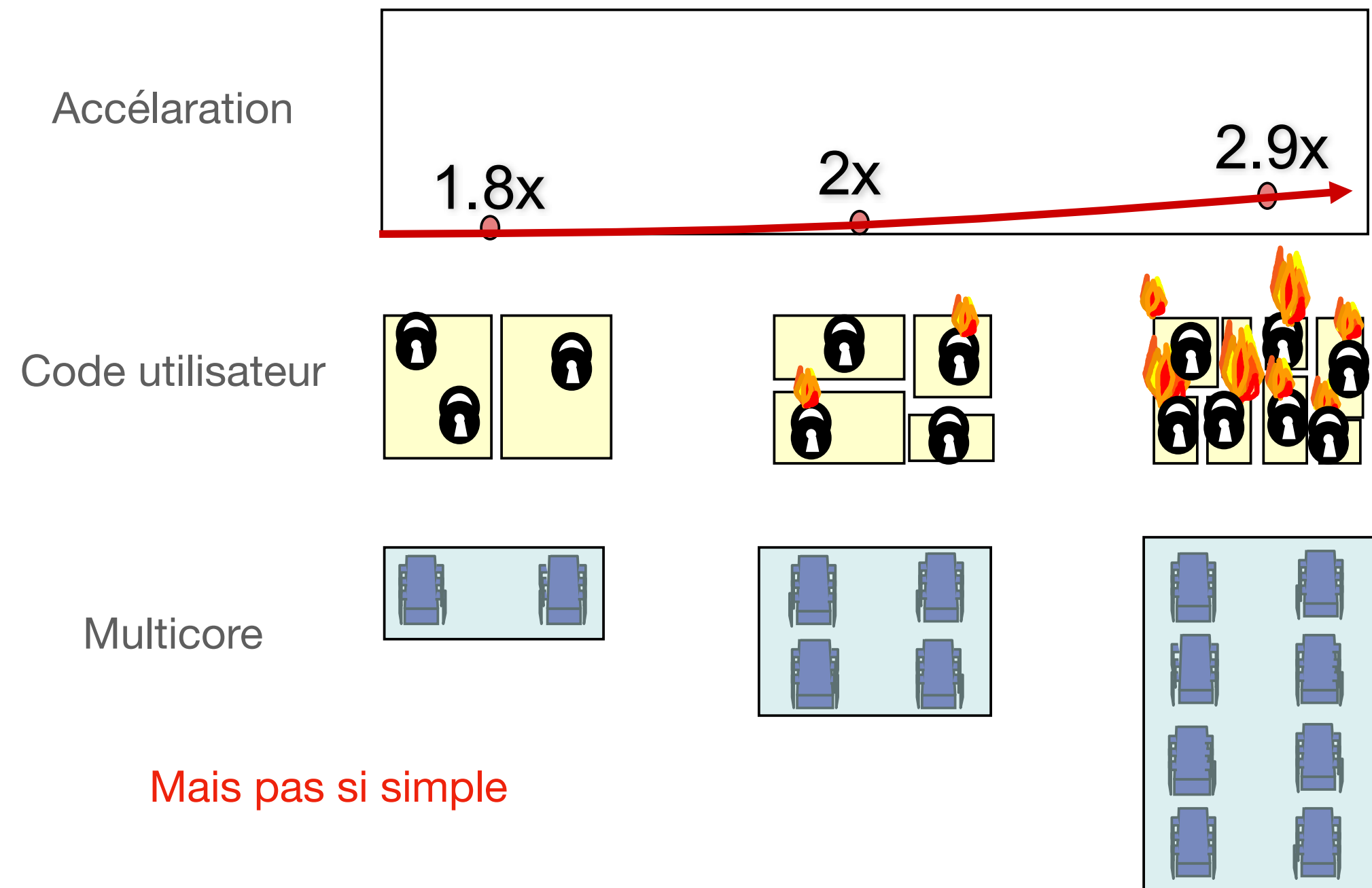
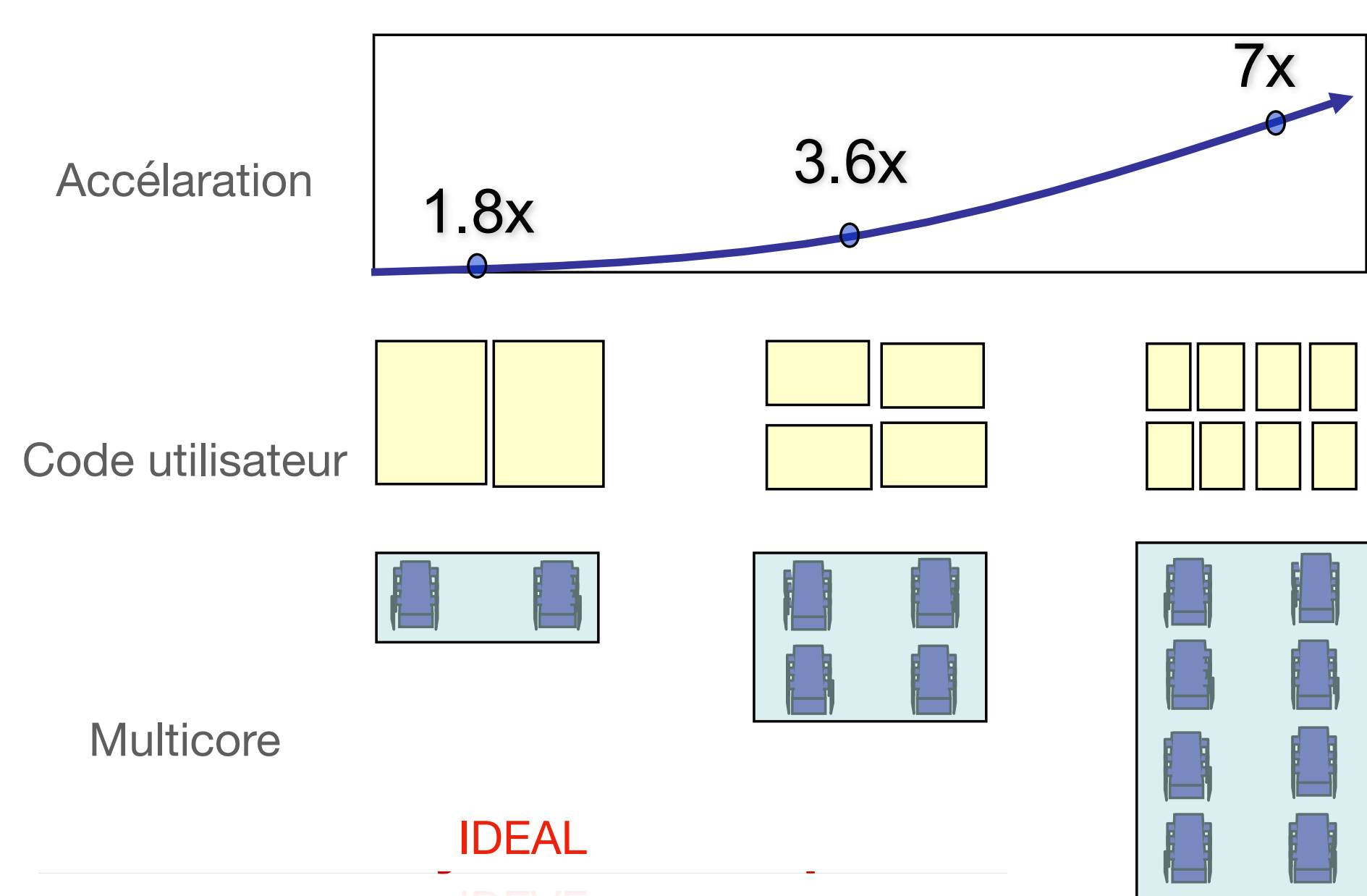


Multicore



Temps : Loi de Moore





Apprenez les principes fondamentaux de la programmation de plusieurs threads accédant à la mémoire partagée des simples verrous aux primitives plus évoluées

Programmez (en Java) des structures de données concurrentes

Algorithmique répartie

Chargée de cours: Lélia Blin (lelia.blin@irif.fr)

L'algorithmique répartie, c'est l'art de faire fonctionner en ensemble des entités de calculs autonomes. Dans ce cours nous découvrirons comment, les entités de calculs peuvent s'organiser entre elles pour résoudre des tâches telles que pratiquées dans les réseaux.

(Il est conseillé de suivre aussi le cours de M2 programmation distribuée.

Plan (sous-réserve)

- Introduction
- Horloges
- Communications
- Etats globaux consistants
- Election
- Partage de ressources
- Construction d'arbres couvrants
- Tolérance aux fautes transitoires

Evaluation

- 2 devoirs
- Un examen final

Note finale: moyenne des 3

Transformation de programmes

Giulio Manzonetto

Transformation de programmes, késako ?

- ▶ Ancien nom : compilation avancée
- ▶ Accent sur les étapes *intermédiaires* d'un compilateur:
- ▶ Remaniement d'AST (arbres de syntaxe)
- ▶ Passages progressifs de langages haut-niveau à bas-niveau
- ▶ Comment rendre les récursivités terminales ?
- ▶ Qu'est-ce qu'une mise en CPS (Continuation Passing Style) ?

Un cours pour qui?

- ▶ Les étudiants qui ont bien aimé le cours de compilation
- ▶ Les étudiants qui n'ont pas apprécié la compilation puisque trop 'bas niveau' (on ne fera ni parsing ni assembleur x86)

Prérequis

Plutôt des recommandations :

- ▶ Avoir suivi M1 Compilation est un plus
 - ▶ thème commun, même si on verra des techniques différentes
- ▶ Connaître au moins un langage fonctionnel
 - ▶ On programmera dans le fragment fonctionnel de Scala

Projet

Transformation d'un langage idéalisé mais représentatif (traits impératifs + fonctionnels) vers du bytecode JVM en assurant une récursivité efficace.

Evaluation

65% projet + 35% examen

Administration Système et Réseaux



Figure: Birth of Unix: D. Ritchie, K. Thompson & PDP11 (~1970)

Administration Système et Réseaux

From: RMS%MIT-OZ@mit-eddie (Richard Stallman)
Newsgroups: net.unix-wizards,net.usoft
Subject: new Unix implementation
Date: Tue, 27-Sep-83 12:35:59 EST

Free Unix!

Starting this Thanksgiving I am going to write a complete
Unix-compatible software system called GNU (for Gnu's Not Unix),
and give it away free to everyone who can use it. Contributions
of time, money, programs and equipment are greatly needed.

Administration Système et Réseaux

From: torvalds@klaava.Helsinki.FI (Linus Benedict Torvalds)

Newsgroups: comp.os.minix

Subject: What would you like to see most in minix?

Date: 25 Aug 91 20:57:08 GMT

Hello everybody out there using minix -

I'm doing a (free) [operating system](#) (just a hobby, won't be big and professional like gnu) [for 386\(486\) AT clones](#). This has been brewing since april, and is starting to get ready. I'd like any feedback on things people like/dislike in minix, as my OS resembles it somewhat (same physical layout of the file-system (due to practical reasons) among other things).

Administration Système et Réseaux

Posted on **Fr 30 April 2010**

Lennart Poettering

Rethinking PID 1

If you are well connected or good at reading between the lines you might already know what this blog post is about. But even then you may find this story interesting. So grab a cup of coffee, sit down, and read what's coming.

This blog story is long, so even though I can only recommend reading the long story, here's the one sentence summary:
we are experimenting with a new init system and it is fun.

“Any sufficiently advanced technology
is indistinguishable from magic.”

Arthur C. Clarke

Administration Système et Réseaux

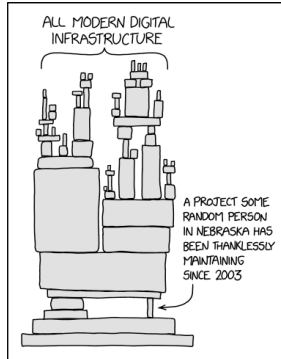


Figure: xkcd #2347

Administration Système et Réseaux

Objectives

- ▶ Craft a system by hand
- ▶ Summon a network with the command line
- ▶ Ascend to the Cloud

Coursework

- ▶ Lab work (not graded)
- ▶ Final exam

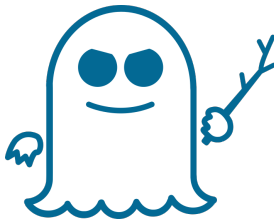
Importance of finding bugs

'worst ever' CPU bugs affect virtually all computers

S.Gibbs, The Guardian



Meltdown



Spectre

- ▶ Am I affected by the vulnerability? **Most certainly, yes.**
- ▶ Has Meltdown or Spectre been abused in the wild? **We don't know.**

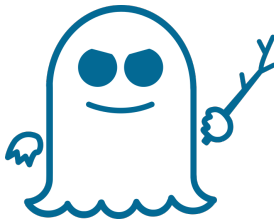
Importance of finding bugs

'worst ever' CPU bugs affect virtually all computers

S.Gibbs, The Guardian



Meltdown



Spectre

"Industries and companies don't spend 100 billion dollars or devote these personnel resources to a problem they think is not serious" — 20 Years Later, the Y2K Bug Seems Like a Joke

Aim

Build correct tools to find bugs in code

```
1. public class AClass {  
2.     public static void main(String args[]) {  
3.         int x = 1;  
4.         int y = 3/(x++ - 1);  
5.     }}
```

Aim

Build correct tools to find bugs in code

```
[ ...]$ analyse AClass.java  
AClass.java:3:  error:  Divide By Zero  
Expression '0' could be zero at line 3.
```

```
1. public class AClass {  
2. public static void main(String args[]) {  
3. >   int x = 1;  
4.     int y = 3/(x++ - 1);  
5. }}
```

Found 1 issue

```
Issue Type(ISSUED_TYPE_ID): #  
Divide By Zero(DIVIDE_BY_ZERO): 1
```

Aim

Build sound languages to write bugless code

```
let _ =  
"foo" + 42;;
```

Aim

Build sound languages to write bugless code

```
[ ...]$ ocaml aprog.ml
```

```
let _ =
```

```
"foo" + 42;;
```

```
Line 2, characters 2-7:
```

```
2 | "foo" > 42;;
```

```
~~~~~
```

```
Error: This expression has type string  
but an expression was expected of type int
```

Prerequisites ₁

Know either OCAML or HASKELL

Prerequisites ₁

Know either OCAML or HASKELL

Prerequisites ₂

Know either OCAML or HASKELL

Analyse Statique 2023-2024

Further details

- ▶ Git repository with teaching material: [click here](#)
- ▶ `syllabus.md` and `overview.pdf`
on-line in the repo
- ▶ Lectures either in English or French
- ▶ Research internships at SAP France, IMDEA Software, and at IRIF
- ▶ First lecture: Jan 8 2024

Arrive *already* organised in team of *two* students.

Analyse Statique 2023-2024

Foundations order theory, results on fixed-points

one lecture

Dataflow Analysis

- ▶ WHILE_{NNH}
- ▶ Heart of optimizing compilers
- ▶ Objective: implement an analysis

Abstract Interpretation

- ▶ WHILE_{RY}
- ▶ INFER, ABSINT, MOPSA, FRAMA-C, ... written in OCAML
- ▶ Objective: implement an abstract interpreter

Along with historical remarks



or digressions on industrial tools.

Analyse Statique 2023-2024



<https://fbinfer.com/>

Analyse Statique 2023-2024

Final grade

100 % project

- ▶ implementation of monotone frameworks
 - generic framework
 - instance to perform constant propagation
- ▶ implementation of abstract interpretation

Méthodes formelles

Approches probabilistes

François Laroussinie

2023-2024

Mardi 8h30-11h30, (à partir du 16 janvier !!!)

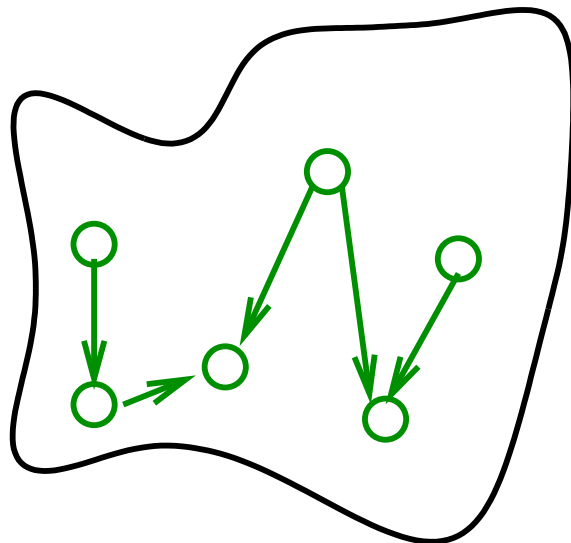
Model-checking

Vérification automatique

System

Properties

Formalizing step



?
 $\models$

φ

Modèles:

- ▶ chaînes de Markov
- ▶ processus de décision de Markov

Propriétés:

- ▶ Accessibilité
- ▶ Logique temporelle probabiliste (PCTL)

Important: apprécier l'algorithmique (notamment dans les graphes) et la logique (temporelle).

Introduction à l'apprentissage profond

Mercredi 13h15-16h15 | Cours : S. Perifel | TP : D. Petrisan & S. Perifel

Cours théorique & pratique :

- Principes théoriques, tenseurs
- Réseaux de neurones à propagation avant
- Rétropropagation
- Réseaux de convolution, transformeurs

TP :

- Programmation avec Pytorch
- Applications du cours
- Projet

Évaluation : 50% examen + 50% projet

Méthodes algorithmiques pour l'accès à l'information numérique

MAAIN

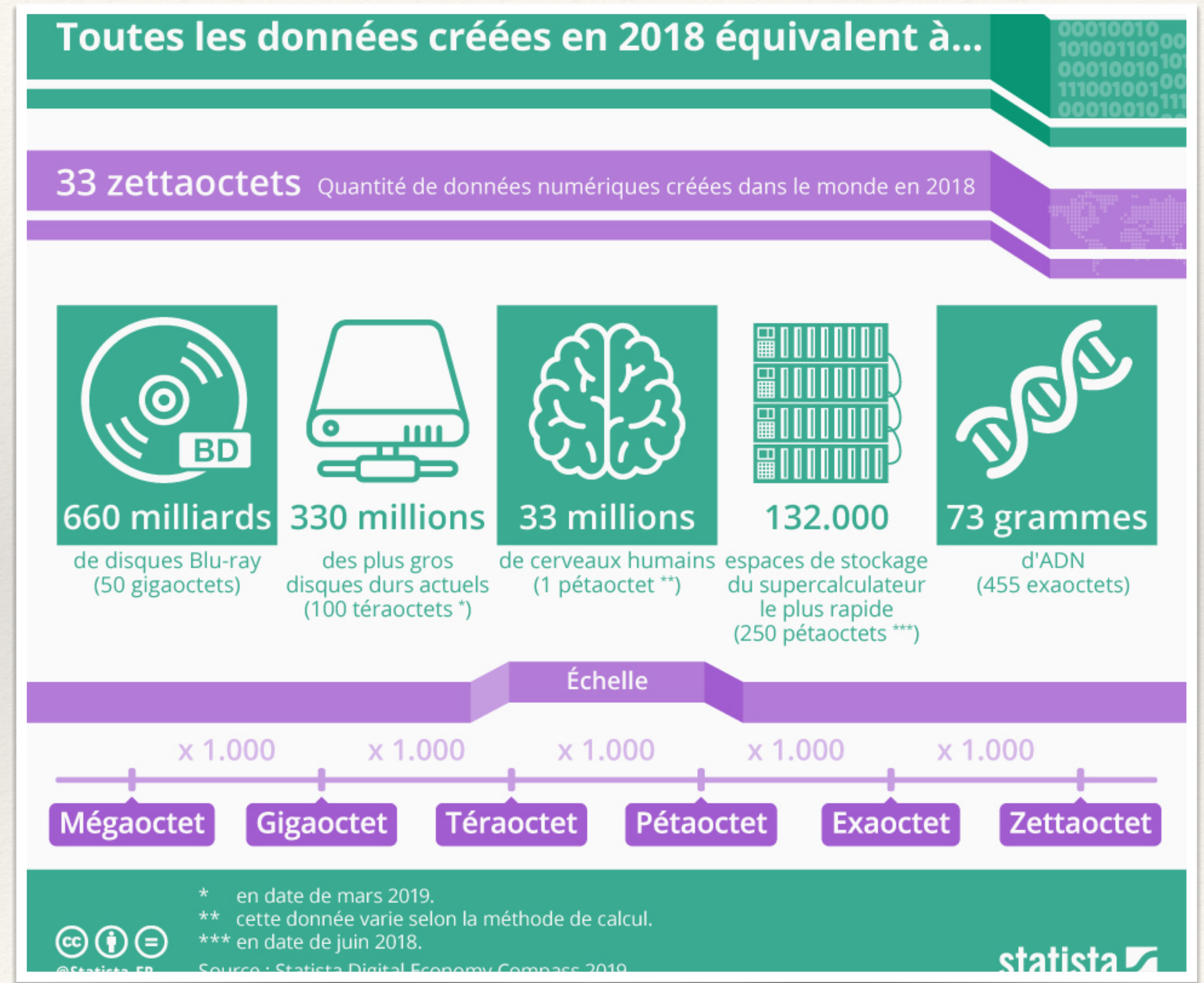
Cours-td & TP : Sylvain Perifel

M2 2022-2023

Trouver une aiguille dans une botte de web

- ❖ Des milliards de pages sur internet
- ❖ Des zettaoctets de données (1 Zo = 10^{12} Go)

➔ Trouver l'information pertinente ???





Graphe du web (Criteo)

1. Moteurs de recherche

Théorie et pratique :

- Aspects algorithmiques
- Programmation

Moteurs de recherche

Algorithmique

Exploration du web

Structuration des données

Calcul de la pertinence des pages (*pagerank*)

Traitement de la requête utilisateur

Projet en binômes en TP

Programmation d'un « **vrai** » **moteur de recherche** sur les pages Wikipédia :

Parser – traitement – pagerank – déploiement

Difficulté gestion mémoire (taille compressée 5 Go) et parsing – langage libre

Le serveur est **opérationnel** à la fin du cours

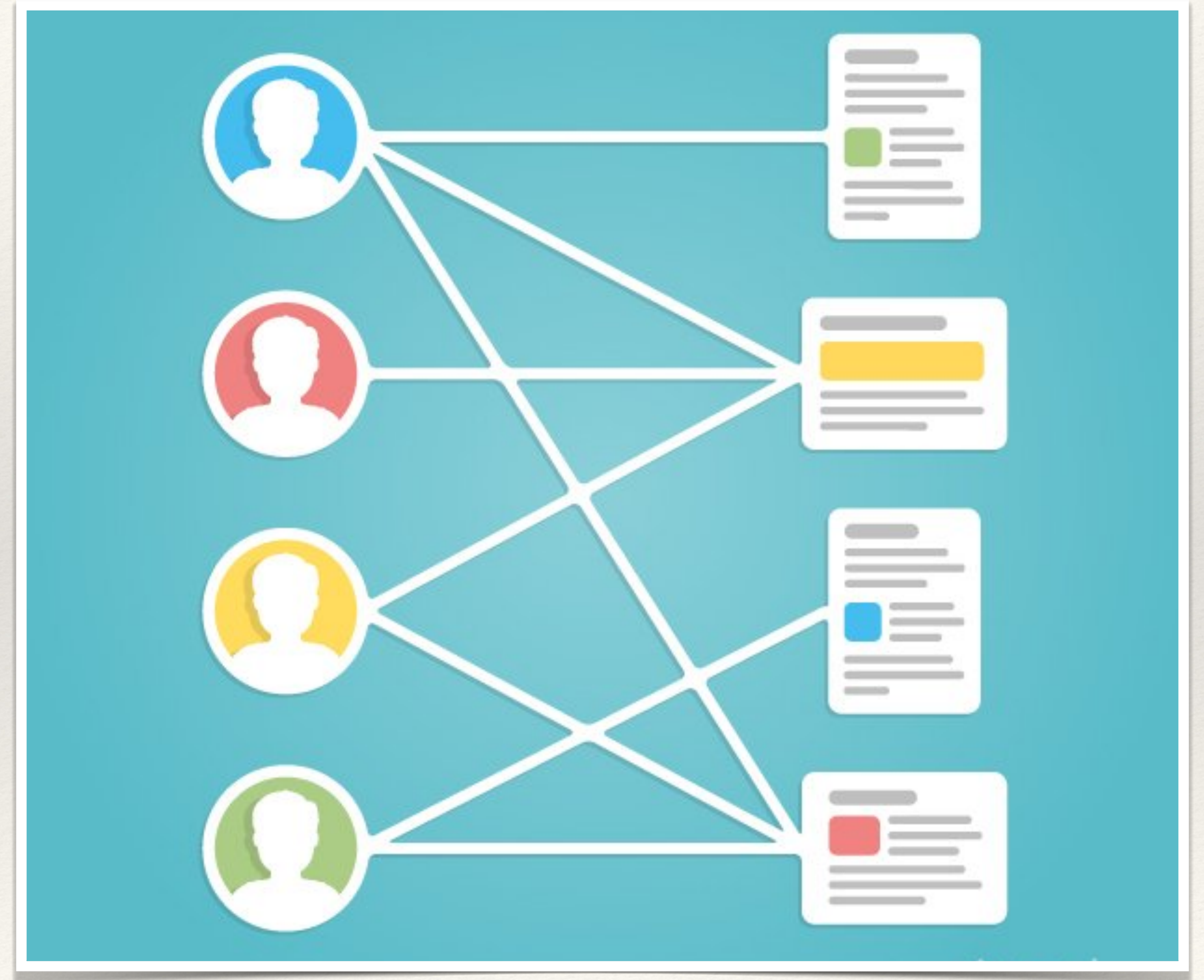
2. Systèmes de recommandation

Recommandations

Comment suggérer la prochaine vidéo pertinente

Comment proposer de nouveaux produits

➔ Algorithmes de recommandation



Prérequis & organisation

Prérequis :

- un soupçon d'algèbre linéaire
- une once d'algorithmique

1h30 de cours (+ TD)

2h de TP

Évaluation : 50 % examen écrit

50 % projet (TP)

« MAAIN, c'est bien. »

-Un étudiant de M2, 1^{er} avril 2022

Optimisation Combinatoire

<http://fabien.viger.free.fr/oc>

Qu'est-ce que l'Optimisation Combinatoire ?

Exemple: Création des emplois du temps

2020/2021 - Semestre 1 : Emploi du temps «L3-INFO»

	lundi	mardi	mercredi	jeudi	vendredi
8h00-8h15					
8h15-8h30					
8h30-8h45					
8h45-9h00					
9h00-9h15					
9h15-9h30					
9h30-9h45					
9h45-10h00	TP SY55 2032 MI E. Haniel-Berle Court	TP CP05 Groupe 1 2031 (+2033) I. Fagnot			TP SY55 Groupe 2 2031 (+2033) D. Poullahon
10h00-10h15		COURS_TD MD5 2027 demi-groupe (ceux qui ont systèmes en présentiel) G. Couteau			TP SY55 Groupe 2 2031 (+2033) I. Klimann
10h15-10h30		COURS_TD MD5 Groupe 4 1020 demi-groupe (ceux qui ont systèmes en présentiel) A. Buccarelli			COURS_TD MD5 Groupe 1 2027 demi-groupe A-H S. Laplante
10h30-10h45					
10h45-11h00					
11h00-11h15					
11h15-11h30					
11h30-11h45					
11h45-12h00	COURS SY55 Info + MI R6 par demi-promo, 1 semaine/2 autre demi-promo (distanciel asynchrone) I. Klimann				
12h00-12h15					
12h15-12h30	COURS_TD MD5 Groupe 4 1020 demi-groupe (ceux qui n'ont pas systèmes en présentiel) A. Buccarelli				
12h30-12h45					
12h45-13h00					
13h00-13h15					
13h15-13h30					
13h30-13h45					
13h45-14h00					
14h00-14h15					
14h15-14h30					
14h30-14h45					
14h45-15h00	TP PF5 Groupe 4 2003 (+2036) A. Guatto				
15h00-15h15					
15h15-15h30					
15h30-15h45					
15h45-16h00					
16h00-16h15					
16h15-16h30					
16h30-16h45					
16h45-17h00					
17h00-17h15					
17h15-17h30	TD AL5 Groupe 4 2036 R. Mantact				
17h30-17h45					
17h45-18h00					
18h00-18h15					
18h15-18h30					
18h30-18h45					
18h45-19h00					
19h00-19h15					
19h15-19h30					
19h30-19h45					
19h45-20h00					
20h00-20h15					
20h15-20h30					
20h30-20h45					
20h45-21h00					
21h00-21h15					
21h15-21h30					
21h30-21h45					
21h45-22h00					
22h00-22h15					
22h15-22h30					
22h30-22h45					
22h45-23h00					
23h00-23h15					
23h15-23h30					
23h30-23h45					
23h45-24h00					
24h00-24h15					
24h15-24h30					
24h30-24h45					
24h45-25h00					
25h00-25h15					
25h15-25h30					
25h30-25h45					
25h45-26h00					
26h00-26h15					
26h15-26h30					
26h30-26h45					
26h45-27h00					
27h00-27h15					
27h15-27h30					
27h30-27h45					
27h45-28h00					
28h00-28h15					
28h15-28h30					
28h30-28h45					
28h45-29h00					
29h00-29h15					
29h15-29h30					
29h30-29h45					
29h45-30h00					
30h00-30h15					
30h15-30h30					
30h30-30h45					
30h45-31h00					
31h00-31h15					
31h15-31h30					
31h30-31h45					
31h45-32h00					
32h00-32h15					
32h15-32h30					
32h30-32h45					
32h45-33h00					
33h00-33h15					
33h15-33h30					
33h30-33h45					
33h45-34h00					
34h00-34h15					
34h15-34h30					
34h30-34h45					
34h45-35h00					
35h00-35h15					
35h15-35h30					
35h30-35h45					
35h45-36h00					
36h00-36h15					
36h15-36h30					
36h30-36h45					
36h45-37h00					
37h00-37h15					
37h15-37h30					
37h30-37h45					
37h45-38h00					
38h00-38h15					
38h15-38h30					
38h30-38h45					
38h45-39h00					
39h00-39h15					
39h15-39h30					
39h30-39h45					
39h45-40h00					
40h00-40h15					
40h15-40h30					
40h30-40h45					
40h45-41h00					
41h00-41h15					
41h15-41h30					
41h30-41h45					
41h45-42h00					
42h00-42h15					
42h15-42h30					
42h30-42h45					
42h45-43h00					
43h00-43h15					
43h15-43h30					
43h30-43h45					
43h45-44h00					
44h00-44h15					
44h15-44h30					
44h30-44h45					
44h45-45h00					
45h00-45h15					
45h15-45h30					
45h30-45h45					
45h45-46h00					
46h00-46h15					
46h15-46h30					
46h30-46h45					
46h45-47h00					
47h00-47h15					
47h15-47h30					
47h30-47h45					
47h45-48h00					
48h00-48h15					
48h15-48h30					
48h30-48h45					
48h45-49h00					
49h00-49h15					
49h15-49h30					
49h30-49h45					
49h45-50h00					
50h00-50h15					
50h15-50h30					
50h30-50h45					
50h45-51h00					
51h00-51h15					
51h15-51h30					
51h30-51h45					
51h45-52h00					
52h00-52h15					
52h15-52h30					
52h30-52h45					
52h45-53h00					
53h00-53h15					
53h15-53h30					
53h30-53h45					
53h45-54h00					
54h00-54h15					
54h15-54h30					
54h30-54h45					
54h45-55h00					
55h00-55h15					
55h15-55h30					
55h30-55h45					
55h45-56h00					
56h00-56h15					
56h15-56h30					
56h30-56h45					
56h45-57h00					
57h00-57h15					
57h15-57h30					
57h30-57h45					
57h45-58h00					
58h00-58h15					
58h15-58h30					
58h30-58h45					
58h45-59h00					
59h00-59h15					
59h15-59h30					
59h30-59h45					
59h45-60h00					
60h00-60h15					
60h15-60h30					
60h30-60h45					
60h45-61h00					
61h00-61h15					
61h15-61h30					
61h30-61h45					
61h45-62h00					
62h00-62h15					
62h15-62h30					
62h30-62h45					
62h45-63h00					
63h00-63h15					
63h15-63h30					
63h30-63h45					
63h45-64h00					
64h00-64h15					
64h15-64h30					
64h30-64h45					
64h45-65h00					
65h00-65h15					
65h15-65h30					
65h30-65h45					
65h45-66h00					
66h00-66h15					
66h15-66h30					
66h30-66h45					
66h45-67h00					
67h00-67h15					
67h15-67h30					
67h30-67h45					
67h45-68h00					
68h00-68h15					
68h15-68h30					
68h30-68h45					
68h45-69h00					
69h00-69h15					
69h15-69h30					
69h30-69h45					
69h45-70h00					
70h00-70h15					
70h15-70h30					
70h30-70h45					
70h45-71h00					
71h00-71h15					
71h15-71h30					
71h30-71h45					
71h45-72h00					
72h00-72h15					
72h15-72h30					
72h30-72h45					
72h45-73h00					
73h00-73h15					
73h15-73h30					
73h30-73h45					
73h45-74h00					
74h00-74h15					
74h15-74h30					
74h30-74h45					
74h45-75h00					
75h00-75h15					
75h15-75h30					
75h30-75h45					
75h45-76h00					
76h00-76h15					
76h15-76h30					
76h30-76h45					
76h45-77h00					
77h00-77h15					
77h15-77h30					
77h30-77h45					
77h45-78h00					
78h00-78h15					
78h15-78h30					
78h30-78h45					
78h45-79h00					
79h00-79h15					
79h15-79h30					
79h30-79h45					
79h45-80h00					
80h00-80h15					
80h15-80h30					
80h30-80h45					
80h45-81h00					
81h00-81h15					
81h15-81h30					
81h30-81h45					
81h45-82					

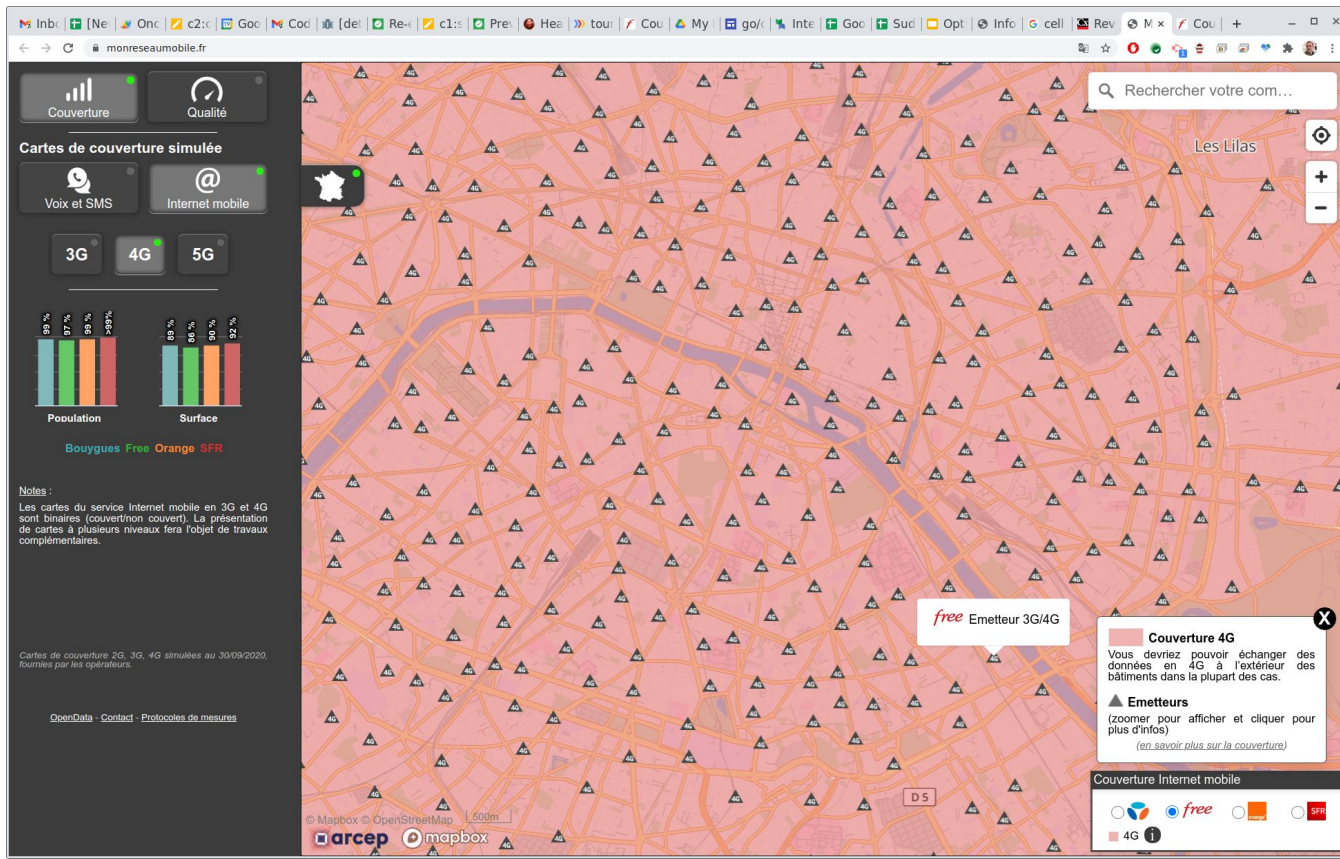
Qu'est-ce que l'Optimisation Combinatoire ?

Autre exemple: Sudoku (cliquer sur l'image)

Input Grid		6			5			2	
								9	
	7							1	
			6		3		4		
			4		7		1		
			5		9		8		
		4				1			6
		3				8			
		2			4			5	

Qu'est-ce que l'Optimisation Combinatoire ?

Autre exemple: où placer les émetteurs 4G ?



Organisation du Cours

Mercredi 15h-18h30, Sophie Germain: 1h30 cours + 1h45 TD

Note = **50% Examen Final** (sur papier) et **50% TD**

Cours au tableau : pas de support de cours, mais vidéos dispo après chaque cours pour réviser si besoin.

TD en C++, avec **Auto-Tests**; corrigés dispos après le TD.

- Adaptés aux débutants en C++ (mais aux experts aussi)
- TD = application *directe* du cours.

Plan indicatif du Cours

1. **Graphes.** Implémentations (C++). Composantes [fortement] connexes. Tables de hachage. Parcours (BFS, DFS). Dijkstra.
2. **Arbres** et leurs algorithmes. Autres graphes particuliers (DAG, **Tri topologique**). Programmation dynamique: 1er exemple.
3. **Programmation dynamique:** autres exemples.
4. **Algorithmes Gloutons:** Arbre couvrant minimum et autres.
5. **Heuristiques, Monte-Carlo** (exemples: diamètre d'un graphe. Miller-Rabin).
6. **Programmation linéaire** (*aka* LP)
7. Programmation linéaire **entière** (*aka* **MIP**)
8. MIP: Modélisation avancée.
9. **Examen blanc** commenté

Extensions possible du Cours

- **SAT** (CDCL)
- **Max-Flow**, Min-cost flow
- Revisions d'algorithmique: Hash tables. Tas. Complexité.

Tout est dispo sur :

<http://fabien.viger.free.fr/oc>

Programmation Logique et Par Contraintes Avancée

Ralf Treinen

December 11, 2023

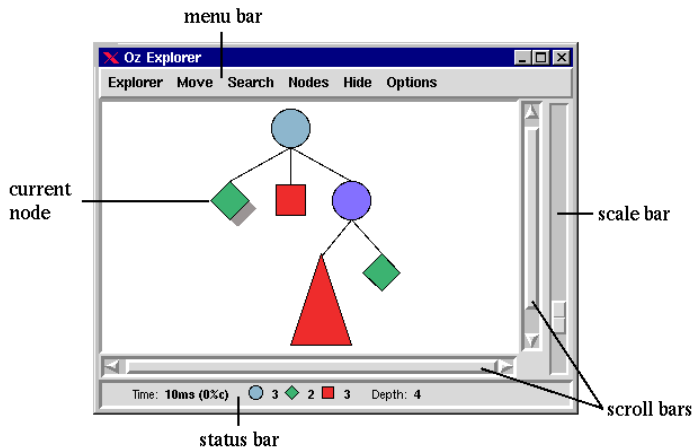
Programmation par contraintes : pourquoi ?

- ▶ Résoudre des problèmes qui demandent la *recherche* d'une solution
- ▶ Résolution de problèmes combinatoires:
 - ▶ Problèmes de planification
 - ▶ Problèmes d'ordonnancement optimal
 - ▶ Problèmes de placement optimal sous contraintes
 - ▶ Résolution des dépendances entre paquet Linux
 - ▶ Des puzzles ...

Programmation par contraintes, est-ce du Prolog ?

- ▶ Non !
- ▶ Nous allons programmer en *Oz* qui est un vrai langage de programmation.
- ▶ Oz permet de faire la programmation fonctionnelle, impérative, orienté objet, des interfaces graphiques, de la distribution ...
- ▶ Construction d'un arbre de recherche peut être seulement une composante d'un programme complexe : la recherche est *encapsulée*.
- ▶ Mécanisme de calcul original, pas de Résolution.

Outil graphique pour interagir avec un arbre de recherche



Pre-requis

- ▶ On auriez besoin de :
 - ▶ Logique : logique propositionnelle et les bases de la logique du premier ordre, notion de base de la résolution de contraintes (simplification d'un système d'équations, par exemple)
 - ▶ Bases de la programmation fonctionnelle (OCaml, Haskell, Scala, F#, ...)
- ▶ Mais on vous n'auriez **pas** besoin de :
 - ▶ Des connaissances de Prolog
 - ▶ Le cours de Programmation Logique du M1

Organisation du cours

- ▶ Format : 1.5h de cours, puis 1.5h de TP.
- ▶ Examen final (2h)
- ▶ Contrôle continu : TP noté dans la deuxième moitié du semestre
- ▶ Note : 50% CC + 50% Examen
- ▶ <http://www.irif.fr/~treinen/teaching/plpc/>

Grands Réseaux d'Interactions

VLADY RAVELOMANANA

IRIF – Université Paris Cité.

vlad@irif.fr

Présentation générale



Plan de la présentation

- Quelques exemples de réseaux.
- Problèmes appréhendés.
- Apports du cours.

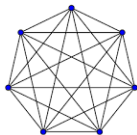


Figure: a. Graphe complet

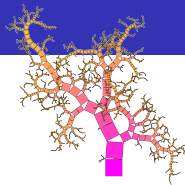


Figure: b. Arbre aléatoire (de Luc Devroye)

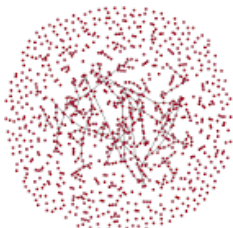


Figure: c. Graphe d'Erdős Rényi

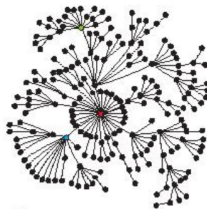


Figure: d. Attachement préférentiel (type www ou internet)

Plan de la présentation

- Quelques exemples de réseaux.
- Problèmes appréhendés.
- Apports du cours.

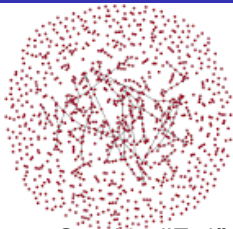


Figure: c. Graphe d'Erdős Rényi

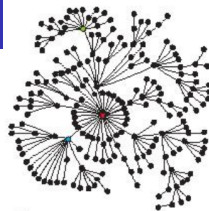


Figure: d. Attachement préférentiel
(type www ou internet)

Exemples de problèmes

- Comment les **construire** et que représentent/modélisent ces objets?
- Comment les **reconnaître**?
- Quelles sont leurs propriétés typiques: **degrés**, “**connexité**”, combien de noeuds sont “**centraux**”?
- **Algorithmes**: comment détecter des communautés (graphes sociaux), comment diffuser des messages efficacement (**broadcast/gossip**)?

Plan de la présentation

- Quelques exemples de réseaux.
- Problèmes appréhendés.
- Apports du cours.

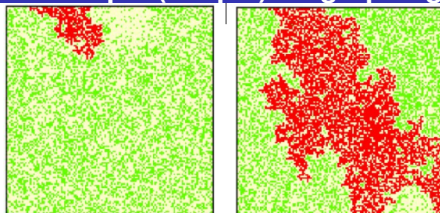


Figure: e. Graphe aléatoire géométrique et simulation feu de forêt

Modèle, algorithmes et questions.

- Un nombre fixé de points (les arbres) sont jetés aléatoirement sur une surface donnée.
- Après le début du feu, la propagation entre arbres est modélisée selon un aléa donné.
- Les questions qui se posent sont **naturelles**:
 - 1 le modèle est-il réaliste (vent, obstacles, ...)?
 - 2 proportion de **destruction/contangion**
 - 3 extension sur un modèle épidémiologique type *COVID19* (graphes dynamiques)?
 - 4 ...



Figure: f. Graphe de co-auteurs.



Figure: g. Graphe de tweets.

Le cours propose : des modèles, des algorithmes pour répondre à certaines questions.

- Attribuer une arête par pair de co-auteurs (ou par 'like').
- Les questions (algorithmiques) qui se posent sont (encore) **naturelles**:
 - 1 Détecter les auteurs dans un domaine ('bioinformatique', 'architecture des ordinateurs', 'informatique théorique', ...). **Idem** pour les influenceurs/chanteurs ('rock', 'reggae', 'mooc', 'cours online', 'documentaires', 'sports', ...)
 - 2 Proportion des domaines, influences des domaines?
 - 3 Visuel des résultats
 - 4 ...



Mobilitéé

Matěj Stehlík
matej@irif.fr



Aperçu du cours

- Nous étudierons des problèmes et algorithmes liés à la mobilité, comme par exemple :
 - le calcul d'itinéraire
 - l'optimisation du trafic dans un réseau
 - l'allocation de fréquences
 - les algorithmes *online*
- Il s'agit d'un cours *théorique*, qui s'appuie beaucoup sur la notion des *graphes* et parfois aussi sur la théorie des jeux.
- Le but est de donner des bases théoriques des différents algorithmes, mais il y aura aussi du travail sur l'ordinateur pour s'appropriier ces concepts, et pour implémenter les algorithmes.

Vous apprendrez. . .

- que trouver le meilleur itinéraire est (beaucoup) plus complexe pour les livreurs que pour les postiers.
- de créer un planning de vols pour une flotte d'avions.
- d'attribuer des fréquences aux antennes d'un réseau cellulaire.
- que (paradoxalement) fermer une route peut améliorer le trafic dans une ville !



Informations complémentaires

- Pré-requis théoriques : graphes (on fera un rappel des notions de base lors du premier cours)
- Pré-requis pratiques : Python ou Java
- Apportez vos propres ordinateurs.
- L'évaluation sera basée sur une épreuve écrite et éventuellement un projet/devoir maison.



Architecture des Systèmes d'Information

Emmanuel Fuchs



Université de Paris



Programme du cours

- Architecture d'Entreprise
- Infrastructure Matérielle des systèmes d'information
- Infrastructure Réseaux des systèmes d'information
- Infrastructure Logicielle des systèmes d'information
- Architecture SI Orientée Service (SOA)
- Architecture SI Orientée Service WEB
- Architecture SI Orientée Service REST
- Framework d'entreprise



Université de Paris