

# Parallel and Distributed Computer Science (PDCS)

Master Informatique  
Faculté des Sciences d'Orsay  
Université Paris-Saclay

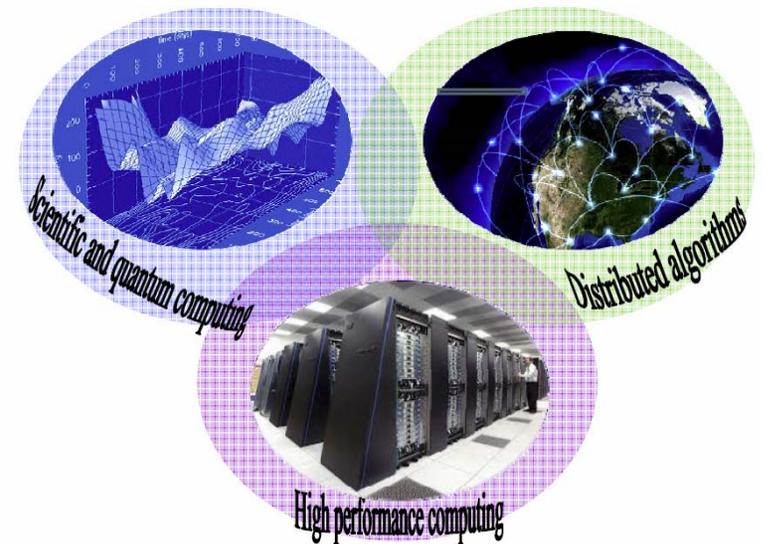
Janna Burman et Oguz Kaya  
janna.burman@lri.fr, oguz.kaya@lri.fr

# PDCS

Systemes complexes  
**distribués et parallèles**

Calcul, algorithmes, programmation  
parallèles, distribués et quantique

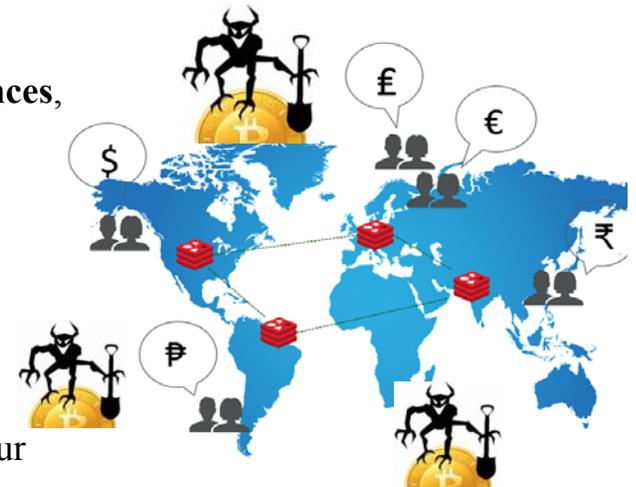
Modélisation, analyse,  
preuves, optimisation



# Présentation de cours PDCS

# Algorithmes distribués robustes (M1)

- **Description (Objectives)**
  - donner les bases de l'**algorithmique distribuée**,
  - **faire comprendre les problèmes** qui se posent lors de la conception d'un système réparti robuste et donner des solutions à ces problèmes,
  - aborder les notions de **preuve** d'algorithme réparti et d'analyse de complexité,
  - sensibiliser les étudiants à l'aspect lié à la **tolérance aux défaillances**, et présenter la **technique de réplication, suivie du consensus**.
- **Pré-requis**
  - notions de base en : réseaux, systèmes, algorithmique classique et algorithmique de graphes
- **MCC/évaluation**
  - Contrôle Continu \* 30% [un ou deux devoirs maison ou devoirs sur table] + Examen Écrit \* 70%
- **Responsables** : Janna Burman et Thomas Nowak



# Auto-stabilisation (M1)

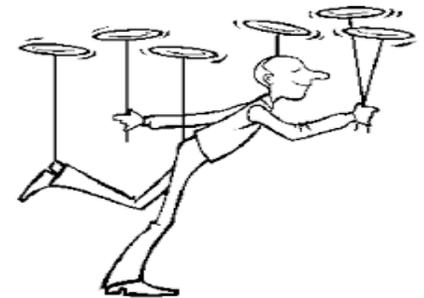
- **Description**

- Après avoir introduit **les bases de l'algorithmique répartie**, nous étudions comment la **technique d'auto-stabilisation** est utilisée **pour rendre robustes les systèmes répartis actuels**.



- **Pré-requis**

- notions de base en : réseaux, systèmes, algorithmique classique et algorithmique de graphes



- **MCC/évaluation**

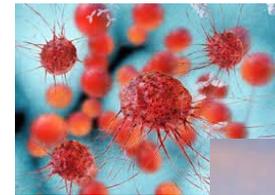
- Contrôle Continu \* 40% [lecture et présentation d'articles] + Examen Écrit \* 60%

- **Responsables** : Janna Burman et Sylvie Delaët

# Algorithmes de la nature (M2)

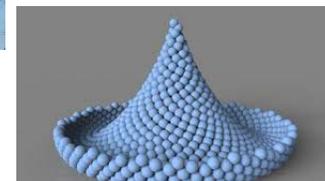
- **Description**

- **La nature a développé des algorithmes répartis** (sans contrôle centralisé), efficaces et peu gourmands en ressources et en énergie. Les réseaux traditionnels s'en sont parfois inspirés.
- Ce module est consacré à **l'étude d'algorithmes liés, d'une façon ou d'une autre, à des phénomènes naturels**. Ils reposent sur des modèles répartis se basant sur des processeurs très limités en ressources et en capacités de calcul et de communication.



- **Pré-requis**

- notions de base en : réseaux, systèmes, algorithmique classique et algorithmique de graphes, algorithmique distribuée (avantage)



- **MCC/évaluation**

- Examen Oral \* 100% [lecture et présentations d'articles]

- **Responsables** : Thomas Nowak et Janna Burman

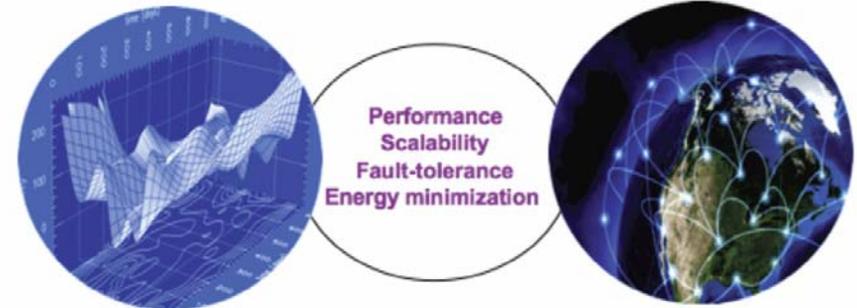
# Frontières du calcul parallèle et distribué (M2)

- **Description**

Ce module se propose de décrire **les avancées les plus récentes en matière de calculs distribué et parallèle**. Son contenu précis est donc à même d'évoluer au cours du temps.

Exascale solvers  
HPC for data analytics  
Quantum algorithms

Distributed algorithms  
Natural algorithms  
Population protocols



- **Pré-requis**

- notions de base en : réseaux, systèmes, algorithmique classique et algorithmique de graphes ; notions de base en algorithmique répartie et parallèle sera un plus

- **MCC/évaluation**

- Examen Oral \* 100% [lecture et présentations d'articles]

- **Responsables :** Thomas Nowak et Laércio Lima Pilla

- **Intervenants:** membres d'équipe ParSys (LRI) et d'autres intervenants extérieurs selon les sujets proposés.

# Ordonnancement et systèmes d'exécution (M2)

- **Description**
  - Ce module s'intéresse à la **gestion efficace des ressources** des systèmes utilisées **pour le calcul scientifique et le traitement de données massives à travers des systèmes d'exécution** et d'autres composants logiciels.
  - Nous discuterons des sujets liés à l'ordonnancement et l'équilibrage de charge en systèmes parallèles, hétérogènes et pour le Big Data.
- **Pré-requis** : Notions de base en systèmes, en programmation parallèle et en algorithmique
- **Évaluations** : Compte-rendu + assiduité, lecture et présentation d'articles
- **Responsable** : Laércio LIMA PILLA – [pilla@lri.fr](mailto:pilla@lri.fr)

# Modélisation et Optimisation des systèmes Discrets (M1)

- Description
  - Analyser et modéliser des systèmes discrets issus de problèmes combinatoires sous forme de programme linéaire ou d'un graphe.
  - Appliquer différentes méthodes pour la résolution de problèmes d'optimisation combinatoire.
- Pré-requis
  - **Conseillés** : Introduction à la programmation linéaire.
- MCC/évaluation
  - Contrôle continu.
- Responsable
  - Abdel Lisser
  - Chargé de TD : Shangyuan Zhang

# Jeux, Apprentissage et Optimisation des Systèmes complexes (M1)

- Description
  - Présenter les liens importants entre l'optimisation, la théorie des jeux et la théorie de l'apprentissage,
  - Présenter quelques sujets fondamentaux dans chaque domaine et de la façon dont les idées de chaque domaine peuvent éclairer les autres.
- Pré-requis
  - **Conseillés** : Bases de la théorie des jeux, bases en optimisation convexe,
- MCC/évaluation
  - Soutenance orale d'articles de recherche liés au domaine du cours.
- Responsable
  - Abdel Lisser
  - Chargé de TD : Shangyuan Zhang

# Optimisation stochastique (M2)

- Description
  - Présenter les problèmes d'optimisation où les décisions sont prises en **présence d'incertitude**.
  - Présenter les fondements théoriques de l'optimisation stochastique, les différentes modélisations de l'aléa et du risque et les méthodes de résolution associées.
- Pré-requis
  - **Conseillés** : Introduction à la théorie des probabilités
- MCC/évaluation
  - Soutenance orale d'articles de recherche
- Responsable
  - Abdel Lisser

# Programmation Orientée Objet (M1)

## Concepts avancés de C++ :

- Surcharge des opérateurs, references
- Généricité : fonction & class templates, itérateurs, STL
- Héritage multiple

## Projet de programmation d'un jeu video utilisant :

- Interface graphique 3D (OpenGL)
- Système multi-agents (joueurs automatiques)
- Jeu en réseau (architecture client/serveur)

- **Pré-requis** : algorithmique, programmation
- **MCC/évaluation** : controle continu et projet final
- **Responsable** : Patrick AMAR



# Programmation C++ Avancée

- **Description**
  - **C++ est un langage qui évolue rapidement et dont la portée dépasse la simple programmation orientée objet.** Il est à la base de 95 % des logiciels qui font que votre vie numérique ne s'effondre pas sous son propre poids.
  - Ce module a pour objectif de vous former à des **techniques de développement puissante permettant d'allier performance, maintenance et abstraction** comme la programmation générique ou la méta-programmation.
- **Pré-requis**
  - [PDCS] Prog. Orientée Objet
  - Avoir des bases dans un langage de programmation arbitraires
- **MCC/évaluation** : TP notés
- **Responsable** : Joel FALCOU (joel.falcou@gmail.com)

# Big Data (M2)

- **Description**
  - Paradigme Map-Reduce : algorithmique et TP en Spark sur cluster de PC
  - Problématique et métriques du passage à l'échelle
  - Mécanismes sous-jacents d'HDFS/Hadoop et de Spark
  - BdD SQL et NoSQL : TP de MongoDB & principes de Spark-SQL, problématique du Join en NoSQL.
- **Pré-requis conseillés** : connaissance et pratique de Python
- **MCC/évaluation** : 50% : 2 CR de TP (en binôme)  
50% : 1 examen écrit individuel à la fin
- **Responsables** : Stéphane Vialle & Gianluca Quercini

# Questions ?

# Règles pour valider PDCS

## Blocs de cours

# Blocs de cours

UEs  
disciplinaires

- **Bloc 1 : Paralle and Distributed Computer Science** – cours obligatoires de spécialité PDCS
  - Tout cours intitulé [PDCS]
  - Programation MPI [ANO] (pas obligatoire pour étudiants M2)
- **Bloc 2 : Informatique générale** - cours de toute spécialité Master Informatique Paris-Saclay
- **Bloc 3 : Soft Skills** – cours parcours [SOFT] (UEs d'ouverture) de compétences complémentaires
  - e.g., langues, communication, formation à la vie de l'entreprise, formation à la recherche, cycle de conférences, etc.
- **Bloc 4 :**
  - En M1 : **Ecole thématique/stage et TER**
  - En M2 : **Stage** dans un laboratoire de recherche ou à l'entreprise

# Notes et compensation

- Les crédits d'une UE sont acquis si l'étudiant obtient à l'UE **une note  $\geq 10/20$**
- Les 60 ECTS de l'année sont acquis lorsque **la moyenne obtenue à chacun des blocs est  $\geq 10/20$**

## **Compensation :**

- Le seuil de compensation des UE compensables est fixé à 07/20.
- **Les cours dans le même bloc** (pour tout bloc) sont compensable entre eux, mais il n'y pas de compensation entre les cours des blocs différents.

## Bloc 2 (options)

En plus de règles générales, dans le bloc 2 (Informatique générale), **au moins 1 cours choisi dans chaque de 3 ensembles de cours suivants :**

- **Ensemble 1 :** Big Data [PDCS, Polytech] / Machine learning [AI] / Mathematics for data science [AI] / Distributed systems for massive data management [DS]
- **Ensemble 2 :** Algorithmique de graphe / Algorithmique probabiliste et jeux / Complexité, décidabilité, modèles de calculs / Langages de programmation et compilation [MPRI]
- **Ensemble 3 :** Réseaux sans fil / Réseaux mobiles / Internet of Things / Blockchain / Programmation système et réseaux [ANO]
- **Exception pour M2 :**
  - 2 cours dans ensemble différents, ou remplacement par cours [PDCS] M1

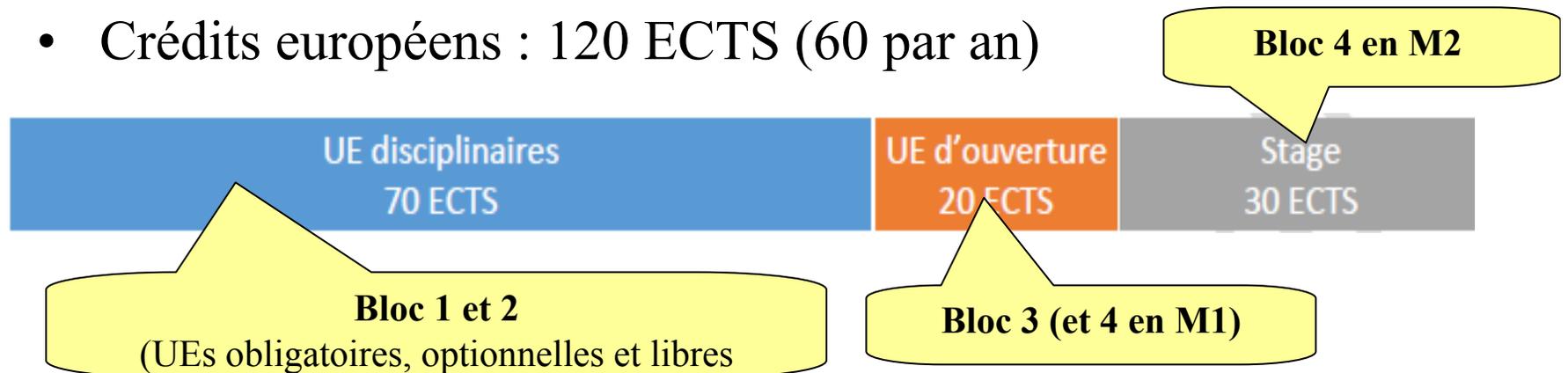
## Bloc 2 (options) - conseils

- **Les étudiants en M2** choisir **Algorithmes distribués robustes** (et/ou Auto-stabilisation) de M1 si jamais suivis des cours théorique en algorithmique distribuée.
- De pas choisir des UEs des parcours en apprentissage (alternance)
- Vérifier les pré-requis de chaque cours choisi
  - les étudiants M1 sont conseillés de choisir plutôt des cours M1 que M2, et en tout cas suivant les pré-requis annoncés
- Vérifier l'EDT
  - que les cours choisis et obligatoires ne coïncident pas

**Équilibrez bien vos trimestres !**

# Rappel et conseil

- Crédits européens : 120 ECTS (60 par an)



- En Master 1 : 4 périodes
  - à chaque période 4 UE disciplinaires + 1 UE d'ouverture
- En Master 2 : 3 périodes
  - à chaque période 3 UE disciplinaires + 1 UE d'ouverture
  - 1 stage

# Rappel et conseil

## M1 (60 ECTS)

S1 (30 ECTS)		S2 (30 ECTS)	
T1 (15 ECTS)	T2 (15 ECTS)	T3 (15 ECTS)	T4 (15 ECTS)
4 UE disciplinaires (au moins 2 obligatoires)			
1 UE d'ouverture	1 UE d'ouverture	1 UE d'ouverture	1 UE d'ouverture
Ecole thématique / Stage (5 ECTS)		TER (5 ECTS)	

## M2 (60 ECTS)

S1 (30 ECTS)			S2 (30 ECTS)
T1 (10 ECTS)	T2 (10 ECTS)	T3 (10 ECTS)	Stage
3 UE disciplinaires (au moins 2 obligatoires)	3 UE disciplinaires (au moins 2 obligatoires)	3 UE disciplinaires (au moins 2 obligatoires)	
1 UE d'ouverture	1 UE d'ouverture	1 UE d'ouverture	

**Bon courage !**

**Questions ?**