

# Consommation électrique d'un data-center

Malo Leroy, Ulysse Tanguy-Bompard

14 mars 2023

# Ancrage au thème et motivation



Ancrage : plus de la moitié des *data center* français sont en zone urbaine

Motivation : écologie, enjeux économiques

# Plan

1. Modélisation d'un ordinateur
2. Grandeurs caractéristiques
3. Puissance consommée
4. Simulation d'un data center
5. Répartition optimale avec l'algorithme du gradient (non fait)

# Modélisation d'un ordinateur

modèles qui n'ont pas été retenus

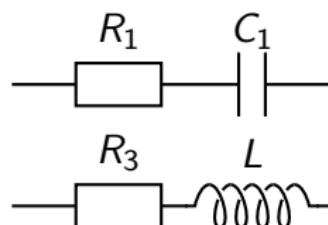
$$U_{\text{eff}} = 230 \text{ V}$$

$$I_{\text{eff}} = 0,20 \text{ A}$$



Ordinateur de bureau complet

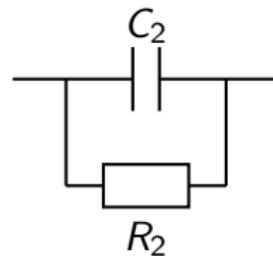
$$R_1 = 1,6 \text{ k}\Omega \quad C_1 = 1,9 \mu\text{F}$$



$$R_3 = 800 \Omega \quad L = 2,6 \text{ H}$$

$$R_2 = 1,6 \text{ k}\Omega$$

$$C_2 = 2,0 \mu\text{F}$$



# Modélisation d'un ordinateur

## Désavantages des modèles non retenus

- ▶ Ordinateur de bureau
  - ▶ Dangereux (230 V)
  - ▶ Grande inertie thermique
- ▶ Circuit équivalent
  - ▶ Grandeur non variables (calculs, température)
  - ▶ Définition de la température difficile

# Modélisation d'un ordinateur

Modèle retenu : Raspberry Pi

## Avantages

- ▶ Peu dangereux
- ▶ Réponse rapide aux perturbations



## Grandeurs caractéristiques

- ▶ Masse volumique  $\rho$
- ▶ Capacité thermique  $C$
- ▶ Conductivité thermique  $\lambda$  (non fait)

## Grandeurs caractéristiques

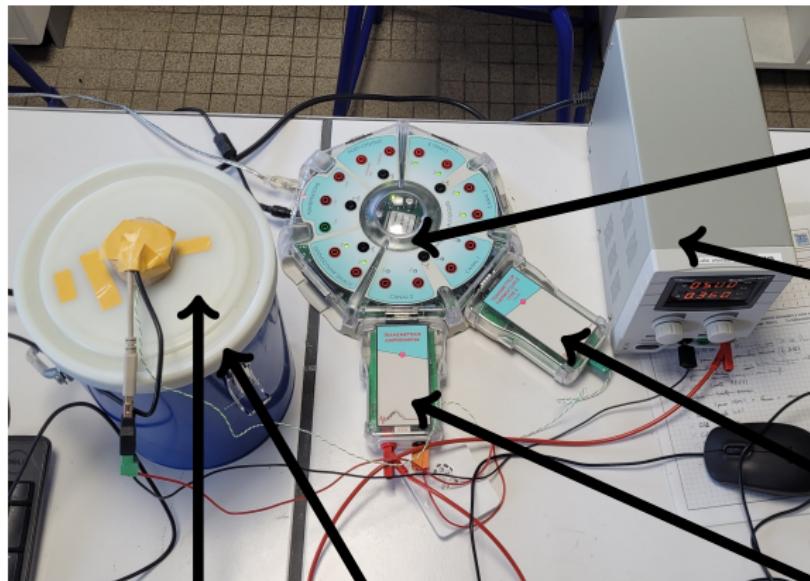
### Masse volumique

Dimensions (au vernier) 85,60 mm × 53,98 mm × 17 mm

Masse (vendeur) 45 g

# Grandeurs caractéristiques

## Capacité thermique



Calorimètre    Raspberry Pi (dans le calorimètre)    Ampèremètre

Carte Sysam

Alimentation

Thermocouple

# Grandeurs caractéristiques

## Masse en eau du calorimètre

### Protocole

1. Le laboratoire est initialement à  $20^\circ\text{C}$
2. On met dans le calorimètre 200 g d'eau chaude à  $52^\circ\text{C}$
3. On mesure après thermalisation (pas trop longue pour éviter les fuites thermiques) la température de l'eau  $T_f = 47^\circ\text{C}$

$$(1^{\text{er}} \text{ principe}) \Delta U = C \Delta T$$

### Résultats

- ▶ Masse en eau  $m_{\text{calo}} = 37 \text{ g}$
- ▶ Capacité thermique  $C_{\text{calo}} = 167 \text{ J/K}$

# Grandeurs caractéristiques

## Capacité thermique du Raspberry Pi

### Protocole

1. Avant calculs,  
 $\langle T \rangle_i = 18,98^\circ\text{C}$  et intensité moyenne de 288 mA
2. Lancement des calculs à  $t = 1 \text{ min}$
3. Fin des calculs à  $t = 381,6 \text{ s}$ .  
Pendant les calculs, on a
  - ▶ la tension 5 V
  - ▶ intensité moyenne 381 mA
  - ▶ le travail électrique  
 $W = 612 \text{ J}$
4. Thermalisation : sur les 100 dernières secondes,  $20,63^\circ\text{C}$

$$(1^{\text{er}} \text{ principe}) \Delta U = W = C \Delta T$$

### Résultats

- ▶ Capacité thermique  
 $C = 204 \text{ J/K}$
- ▶ Capacité thermique massique  
 $c = 4,5 \text{ kJ} \cdot \text{K}^{-1} \cdot \text{kg}^{-1}$

### Commentaires

- ▶ valeur élevée (ex. Silicium )
- ▶ masse de la carcasse
- ▶ pertes thermiques (10 min)

## Puissance consommée

1. Définition de la quantité de calcul
2. Mesure de la consommation du Raspberry Pi
3. Régression linéaire

# Puissance consommée

## Définition de la quantité de calcul

- ▶ Défini à une constante près (au repos,  $K = 0$ )
- ▶ Unité standard : *floating-point operation*
- ▶ Un calcul est une opération sur des flottants

```
def calculs(n):
    for _ in range(n):
        a, b = 60986.5150141834, 2831540.2372984355
        c = a ** (-b)
```

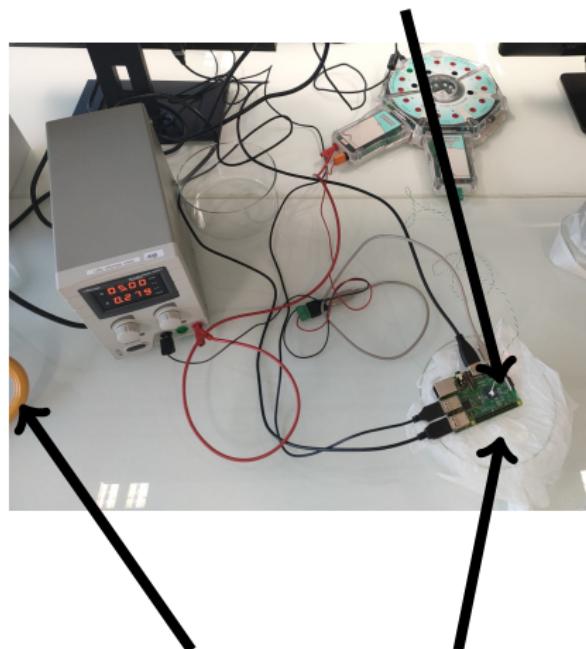
# Puissance consommée

## Mesure de l'intensité

Raspberry Pi

### Protocole

1. On influe sur la température (glace, lampe de chantier)
2. On impose la quantité de calculs
3. On mesure l'intensité consommée

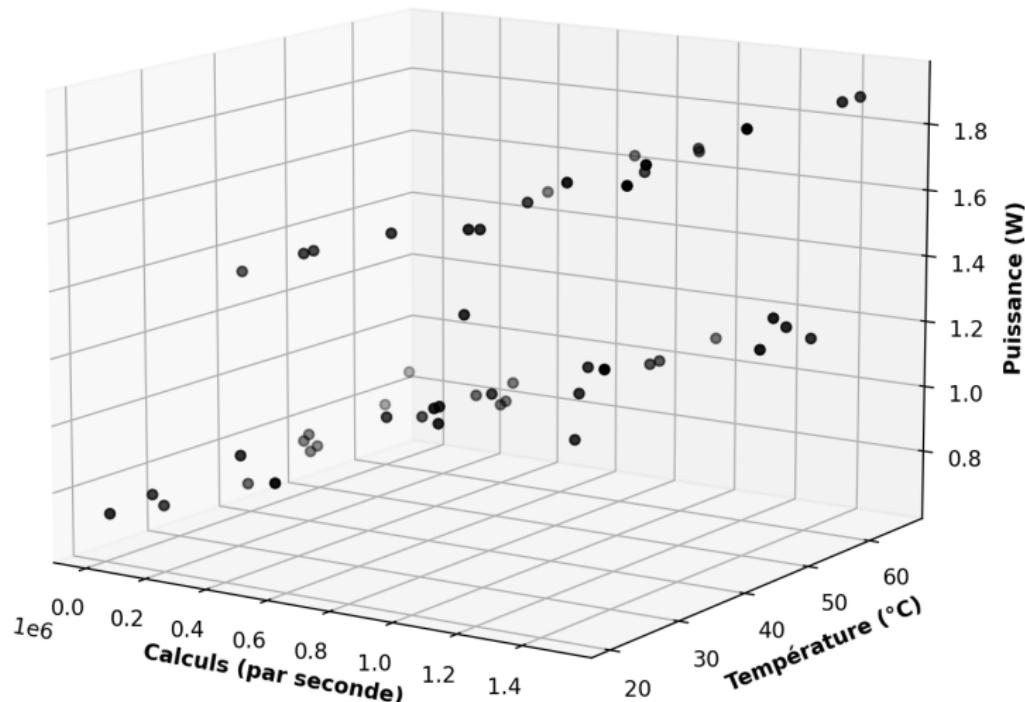


Lampe de chantier

Glace

# Puissance consommée

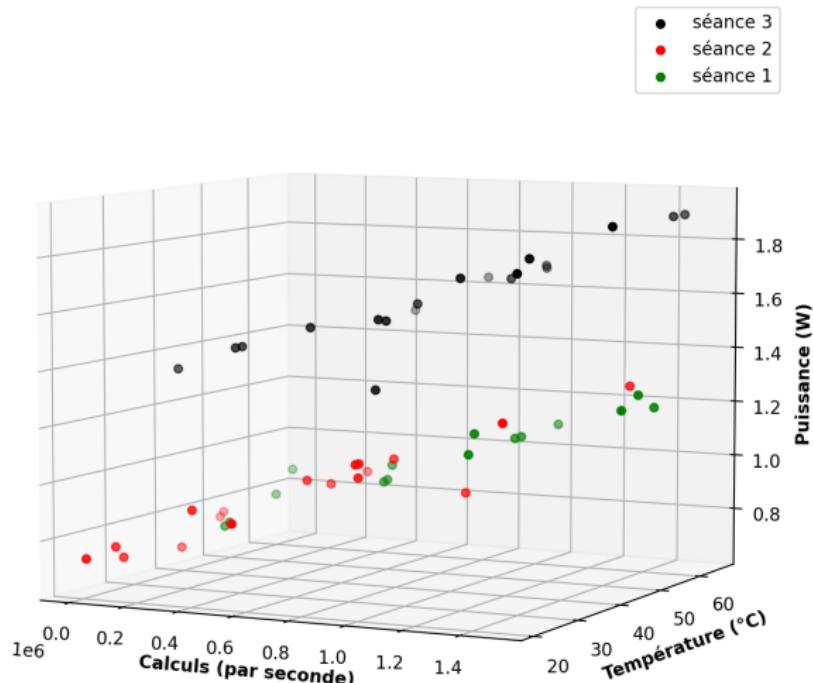
## Résultats expérimentaux



Problème : on a « deux plans »

# Puissance consommée

## Résultats expérimentaux



Sans doute une erreur systématique (par exemple : placement du thermomètre, calculs au repos différents)

# Puissance consommée

## Régression linéaire

- ▶ Allure de plan

$$P = a \times K + b \times T + c$$

- ▶ Régression linéaire avec `np.linalg.lstsq`

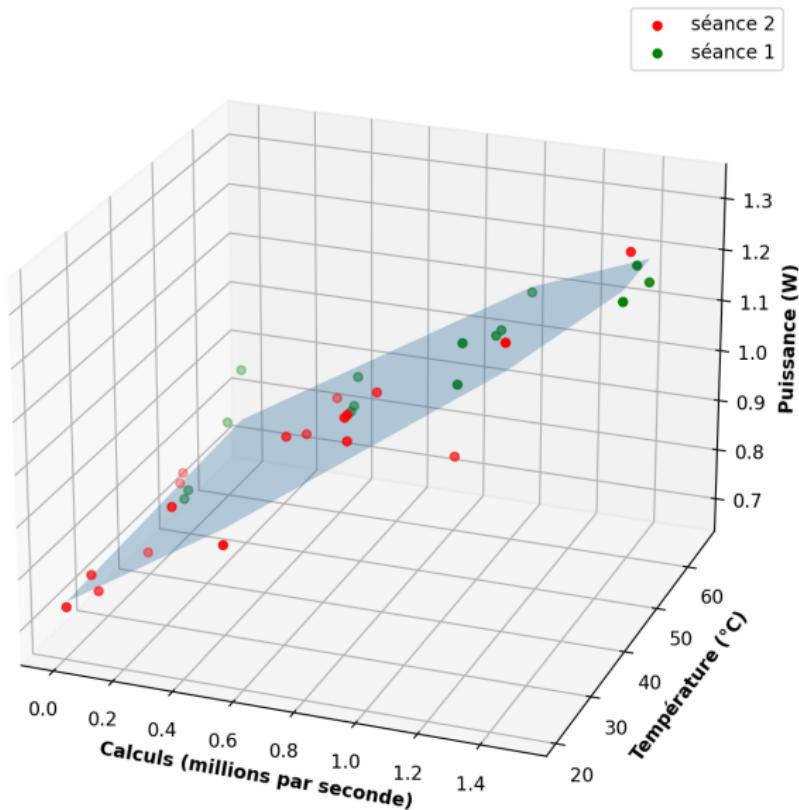
- ▶  $a = 5,1 \times 10^{-4} \text{ W/K}$

- ▶  $b = 3,4 \times 10^{-7} \text{ W/calcul par seconde}$

- ▶  $c = 0,73 \text{ W}$

# Puissance consommée

## Régression linéaire



# Simulation d'un data center

## Hypothèses

- ▶ Carcasse de l'ordinateur : pavé uniforme
- ▶ Source thermique : pavé centré sur la carcasse
  - ▶ Puissance volumique uniforme
- ▶ Air extérieur
  - ▶ Pas de convection (seulement la conduction)
  - ▶ Conductivité dépendant de la température
  - ▶ Masse volumique dépendant de la température

# Simulation d'un data center

## Principe de l'algorithme

- ▶ Temps et espace discrétisés
- ▶ Température : tableau numpy
  - ▶ 1D  $T[x, t]$  (cylindre)
  - ▶ 2D  $T[x, y, t]$  (plan épais)
- ▶ Équation différentielle (1D)

$$\frac{\partial T}{\partial t} = D(x) \frac{\partial^2 T}{\partial x^2} + P_c(x) \quad \text{où} \quad P_c(x) = \frac{P_v}{\rho c}$$

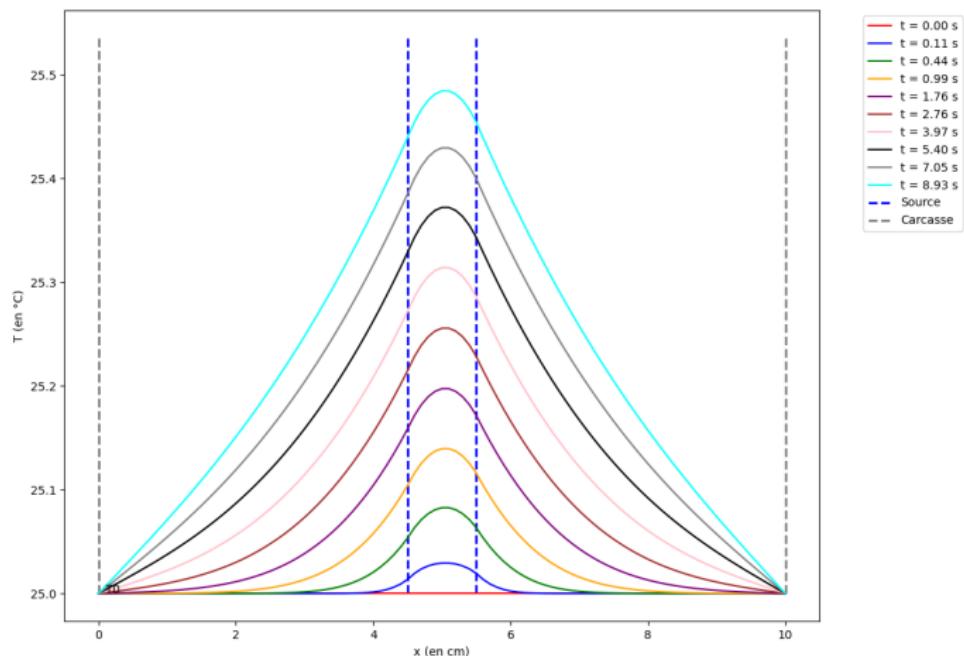
- ▶ Traduction informatique (1D)

```
T[xi, ti+1] = T[xi, ti] + (t_e / (x_e**2)) * D(xi)
               (T[xi+1, ti] - T[xi, ti] + T[xi-1, ti])
               + P_c(xi) * t_e
```

- ▶ Deux hypothèses
  - A Air à température constante
  - B Enceinte calorifugée (avec air)

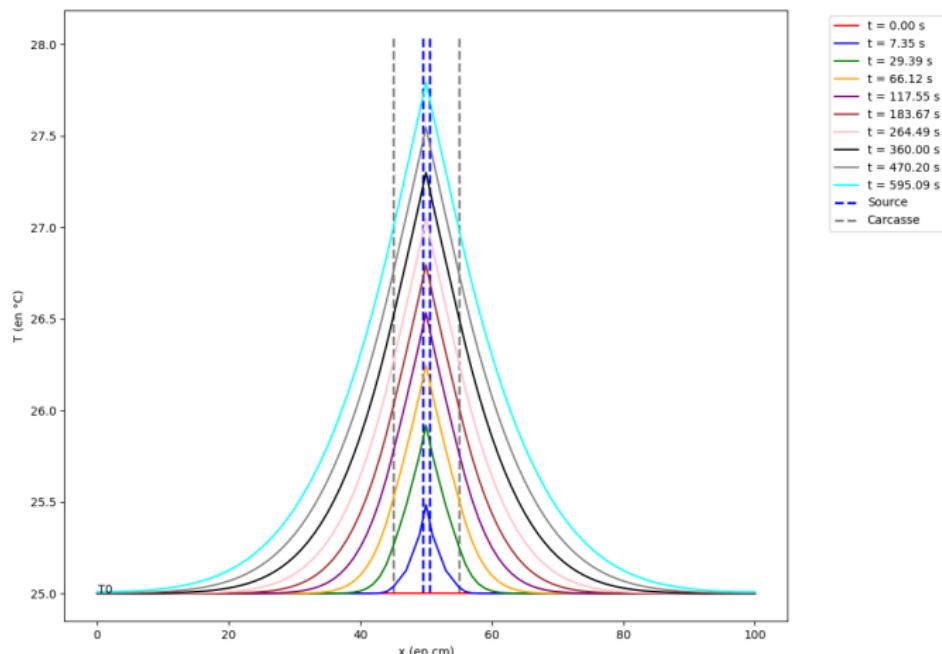
# Simulation d'un data center

## Résultats pour l'hypothèse A



# Simulation d'un data center

## Résultats pour l'hypothèse B



## Ce qui reste à faire

- ▶ Simulation à 2D puis à 3D
- ▶ Détermination de la conductivité thermique du Raspberry Pi
- ▶ Répartition optimale avec l'algorithme du gradient
  - ▶ Répartition des calculs entre les ordinateurs
  - ▶ Répartition spatiale des ordinateurs