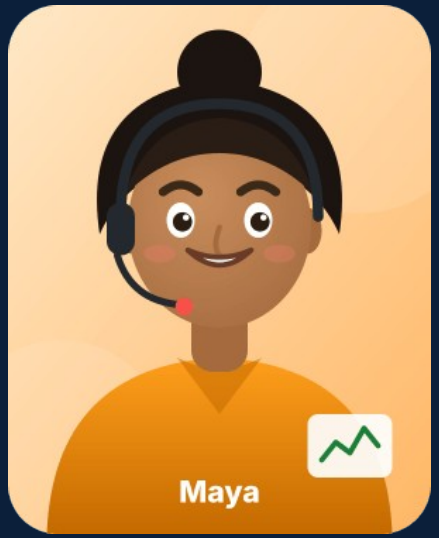
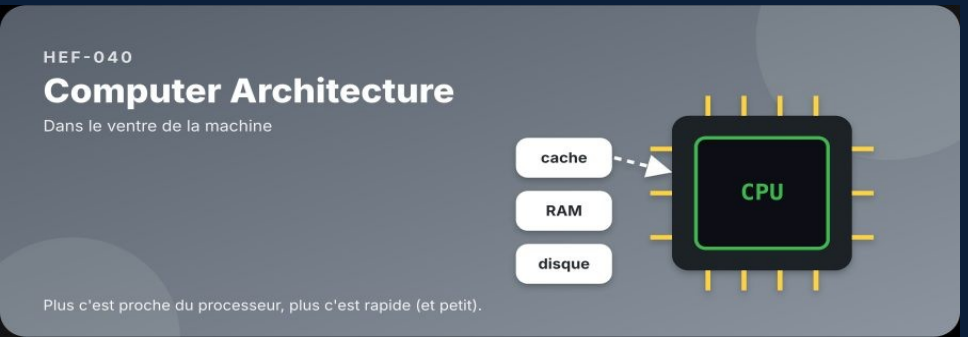
HEF-040 · MISSION 10 / 10

Étude d'une configuration réelle

Lire une fiche technique, c'est lire une histoire de compromis. À toi de décrypter.





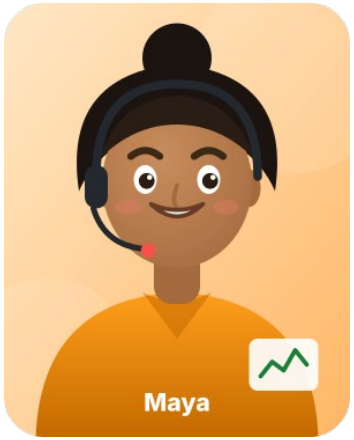
Badge : Architecte matériel

80 min · débutant

L'histoire

HashBox doit acheter une machine pour héberger sa nouvelle plateforme de cours vidéo. Deux fournisseurs envoient leurs fiches techniques. Maya les pose sur la table : « Laquelle prend-on ? »

> *« Pas à l'intuition, dit Sam. On traduit chaque ligne en capacité réelle, on identifie les goulots, et on compare à notre charge. »*



Dans cette dernière mission, tu mobilises tout le module sur un cas concret.

Ta mission

À la fin, je serai capable de…

1. décoder les principales lignes d'une fiche technique (CPU, cache, RAM, stockage, réseau)
2. calculer des capacités théoriques (calcul, bande passante)
3. identifier le goulot d'une configuration pour une charge donnée
4. comparer deux configurations avec des critères chiffrés
5. rédiger une recommandation argumentée

Défi d'échauffement

Défi d'échauffement

Tu choisis entre deux ordinateurs portables. A : 8 Go de RAM, SSD 512 Go, processeur 4 cœurs à 4 GHz. B : 32 Go de RAM, SSD 256 Go, processeur 8 cœurs à 3 GHz. Tu veux faire du montage vidéo. Quelles informations te manquent avant de choisir ? Quelle ligne du tableau te semble la plus décisive pour le montage vidéo ?

Le concept

Décoder une fiche technique

Ligne	Ce qu'elle dit	Ce qu'elle ne dit pas
Cœurs / threads	parallélisme disponible	si ton programme sait l'utiliser
Fréquence (GHz)	cycles par seconde	l'IPC, la fréquence soutenue sous charge
Caches L1/L2/L3	taille des mémoires rapides	leur latence, leur partage
RAM : capacité	combien on peut garder en mémoire	la vitesse
RAM : type et fréquence	bande passante potentielle	le nombre de canaux utilisés
Stockage	capacité, type (SATA, NVMe)	l'endurance, le débit soutenu
Réseau	débit nominal (1 Gb/s, 10 Gb/s)	la latence
TDP	puissance thermique indicative	la consommation réelle

Le concept

Calculs de capacité

- **Calcul de pointe (ordre de grandeur)** = cœurs × fréquence (GHz) × instructions par cycle.
- **Bande passante mémoire** = canaux × 8 octets × transferts par seconde.
  - DDR4-3200 : 3 200 MT/s × 8 o = 25,6 Go/s **par canal**.
  - DDR5-4800 : 4 800 MT/s × 8 o = 38,4 Go/s par canal.
- **Débit réseau** : 1 Gb/s ≈ 125 Mo/s ; 10 Gb/s ≈ 1,25 Go/s.
- **Débit disque** : SATA SSD ≈ 500 Mo/s ; NVMe ≈ 2 à 7 Go/s.

Identifier le goulot

Pour une charge donnée, le **goulot** est la ressource la plus saturée :

Le concept

Identifier le goulot (suite)

Charge	Ressource probable
Streaming vidéo (beaucoup d'utilisateurs)	réseau, puis disque
Calcul scientifique	cœurs, bande passante mémoire
Base de données	RAM, disque (aléatoire)
Compilation	cœurs, disque

Mesurer sa propre machine

```
import os, platform, shutil

print("Système :", platform.system(), platform.machine())
print("Cœurs logiques :", os.cpu_count())
total, utilise, libre = shutil.disk_usage("/")
print("Disque :", round(total / 1e9), "Go,", round(libre / 1e9), "Go libres")
```

Le concept

Mesurer sa propre machine (suite)

🧠 ***Astuce*** *: transforme chaque ligne de la fiche en un chiffre comparable (Go/s, ops/s, Mo/s). Les chiffres s'additionnent et se comparent ; les adjectifs ne le font pas.*

Exemple pas à pas

Problème : comparer deux serveurs pour un service de vidéo à la demande.

	Serveur X	Serveur Y
CPU	8 cœurs, 3,5 GHz	24 cœurs, 2,6 GHz
RAM	64 Go DDR4-3200, 2 canaux	128 Go DDR5-4800, 4 canaux
Stockage	2 × SSD SATA (500 Mo/s chacun)	2 × NVMe (3 Go/s chacun)
Réseau	1 Gb/s	10 Gb/s

```
def bande_passante_mem(canaux, mt_s):
    return canaux * 8 * mt_s / 1e9           # Go/s

serveurs = {
    "X": dict(coeurs=8, ghz=3.5, mem=bande_passante_mem(2, 3200), disque_mo_s=2 *
   500, reseau_mo_s=125),
    "Y": dict(coeurs=24, ghz=2.6, mem=bande_passante_mem(4, 4800), disque_mo_s=2 *
```

Exemple pas à pas

Exemple pas à pas (suite)

```
    3000, reseau_mo_s=1250),
}

debit_par_flux_mo_s = 0.6          # un flux vidéo HD ≈ 5 Mb/s ≈ 0,6 Mo/s
for nom, s in serveurs.items():
    par_reseau = s["reseau_mo_s"] / debit_par_flux_mo_s
    par_disque = s["disque_mo_s"] / debit_par_flux_mo_s
    capacite = min(par_reseau, par_disque)
    goulot = "réseau" if par_reseau < par_disque else "disque"
    print(f"{nom}: mémoire {s['mem']:.1f} Go/s ; flux simultanés max ≈ {capacite:.
   0f} (goulot : {goulot})")
```

Lecture : pour X, le réseau limite à environ 200 flux. Pour Y, environ 2 000. Le meilleur CPU n'aurait servi à rien pour X : **le goulot est ailleurs**. Pour une charge de calcul pur, le verdict pourrait changer.

À toi, avec aide

Un service de **compilation en intégration continue** (beaucoup de petites tâches parallèles qui lisent et écrivent des fichiers) hésite entre les serveurs X et Y de l'exemple.

1. Quelle ressource est probablement le goulot ?
2. Quelle machine recommandes-tu ?
3. Quelle mesure te donnerait confiance avant d'acheter ?

À toi, avec aide · réponse

Indices progressifs

Indice 1

Compilation = beaucoup de cœurs + beaucoup d'accès disque.

Indice 2

Compare le nombre de cœurs et le débit disque.

Indice 3

Mesurer sur une machine d'essai la durée d'une vraie compilation.

Réponse

Goulots probables : les cœurs (tâches parallèles) puis le disque (fichiers). Le serveur Y (24 cœurs, NVMe) est favorable : plus de cœurs, disque beaucoup plus rapide ; la fréquence par cœur plus basse compte moins si le travail est bien parallèle. Mesure avant d'acheter : exécuter la compilation réelle sur une machine de test comparable (ou un service cloud de même taille) et noter la durée et l'utilisation des cœurs/disque.

À toi, en solo

Ta machine. Écris un script qui affiche les caractéristiques de ta machine (système, architecture, cœurs, disque), puis :

1. mesure la vitesse d'écriture séquentielle sur 100 Mo dans un dossier temporaire ;
2. mesure la vitesse de lecture ;
3. estime le temps pour copier 10 Go ;
4. compare avec les valeurs théoriques d'un SSD SATA (500 Mo/s) et d'un NVMe (3 Go/s).

À toi, en solo · réponse

Pistes de correction

```
import os, platform, shutil, tempfile, time

print(platform.system(), platform.machine(), os.cpu_count(), "cœurs logiques")
total, _, libre = shutil.disk_usage(tempfile.gettempdir())
print(round(libre / 1e9), "Go libres")

taille = 100 * 1024 * 1024
bloc = os.urandom(1024 * 1024)
with tempfile.TemporaryDirectory() as d:
    chemin = os.path.join(d, "test.bin")
    t = time.perf_counter()
    with open(chemin, "wb") as f:
        for _ in range(100):
            f.write(bloc)
        f.flush(); os.fsync(f.fileno())
    ecriture = taille / (time.perf_counter() - t) / 1e6

    t = time.perf_counter()
    with open(chemin, "rb") as f:
```

Pistes de correction (suite)

```
        while f.read(1024 * 1024):
            pass
    lecture = taille / (time.perf_counter() - t) / 1e6

print(f"écriture ≈ {ecriture:.0f} Mo/s ; lecture ≈ {lecture:.0f} Mo/s (peut être s
   ervie par le cache de pages)")
print(f"copie de 10 Go ≈ {10_000 / ecriture:.0f} s à la vitesse d'écriture mesurée
   ")
```

La lecture est souvent faussée par le cache de pages du système (le fichier vient d'être écrit). Pour une vraie mesure, on lit un fichier plus gros que la RAM ou on vide le cache.

Pièges classiques

Piège	Pourquoi c'est un problème	Comment l'éviter
Comparer sur un seul chiffre	GHz seuls ne disent rien	Traduire en capacités par ressource
Oublier le goulot	Un super CPU avec un petit réseau reste limité	Trouver la ressource la plus saturée
Croire les débits théoriques	Les débits réels sont plus bas	Mesurer sur la charge réelle
Négliger l'évolutivité	La machine saturera dans un an	Estimer la croissance

Mini-boss

Mini-boss

Dossier d'achat final du module HEF-040. HashBox héberge une plateforme avec trois services : (1) vidéos à la demande, (2) base de données de progression, (3) tâches de calcul nocturnes (rapports). Deux configurations candidates (invente des fiches réalistes : cœurs, GHz, RAM, canaux, stockage, réseau, prix).

1. Écris un programme qui calcule, pour chaque configuration et chaque service, un indicateur de capacité et le goulot.
2. Compare les configurations sur capacité **et** prix par unité de capacité.
3. Décris l'impact du cache et de la localité sur le service (2).
4. Propose une architecture finale (une machine, ou deux machines spécialisées).
5. Rédige une recommandation d'une page : hypothèses, calculs, risques, et mesures à faire avant achat.

Mini-boss · réponse

Pistes de correction

Un bon dossier : chaque service associe un goulot (réseau/disque pour la vidéo ; RAM/disque aléatoire pour la base ; cœurs/bande passante mémoire pour les rapports) ; les calculs sont montrés (Go/s, flux simultanés, temps de rapport) ; la comparaison inclut un coût ; l'architecture peut séparer les services (stockage NVMe pour la base, réseau rapide pour la vidéo) ; la recommandation liste les incertitudes (débits réels, croissance) et propose des mesures de validation (test de charge, benchmark de la base).

Quiz éclair

Quiz éclair

1. Pour identifier le goulot d'une configuration, on cherche :

A. la ressource la plus saturée par la charge
B. la ressource la moins chère
C. le composant le plus récent

2. Un réseau de 10 Gb/s offre environ :

A. 10 Go/s
B. 1,25 Go/s
C. 100 Mo/s

3. La bande passante mémoire dépend notamment :

A. de la couleur des barrettes

Quiz éclair

Quiz éclair (suite)

B. de la taille du disque

C. du nombre de canaux et de la fréquence de transfert

4. 🔁 *Révision mission 09* : La loi d'Amdahl dit que l'accélération par parallélisme est limitée par :

A. la partie non parallélisable

B. la fréquence

C. la taille de la RAM

5. 🔁 *Révision mission 08* : Un cadre de pile contient notamment :

A. le code source

B. les variables locales et l'adresse de retour

C. les fichiers ouverts

Quiz éclair · réponse

Réponses

1. **A**.
2. **B**.
3. **C**.
4. **A**.
5. **B**.

Je réfléchis

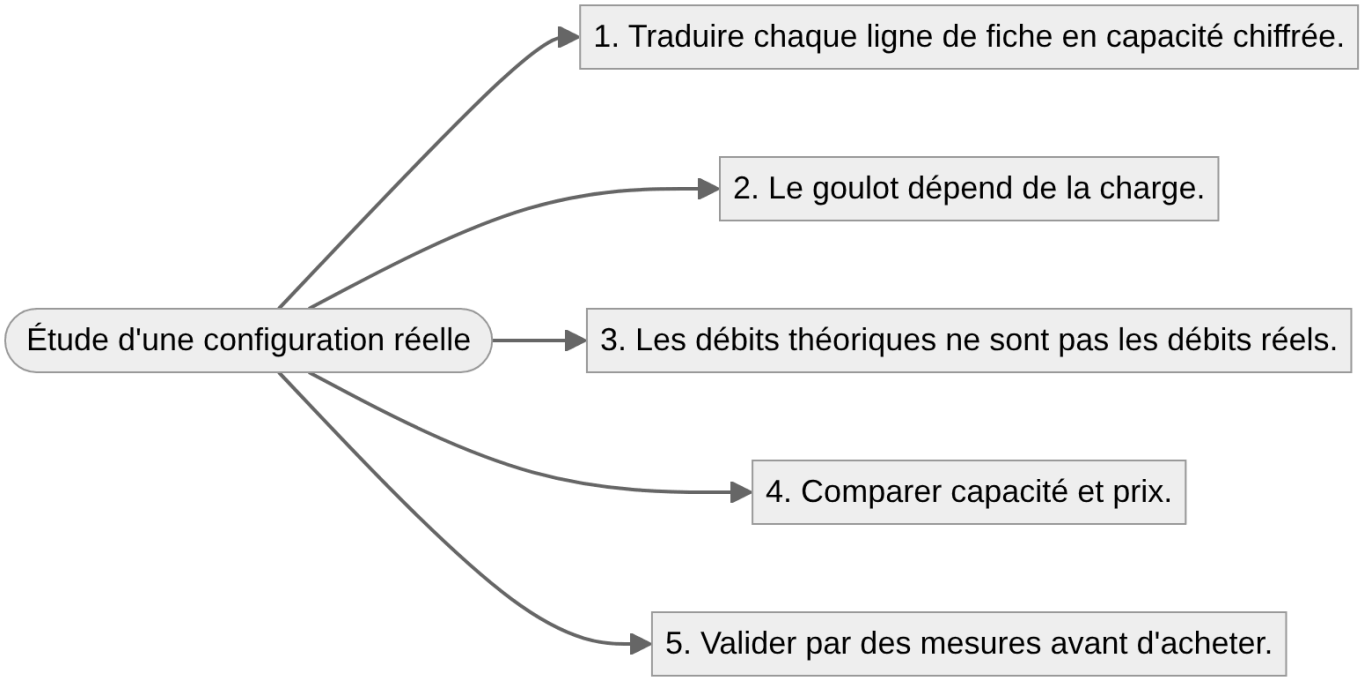
1. Quelle ligne de fiche technique te semblait la plus importante avant ce module, et laquelle te paraît la plus décisive maintenant ?
2. Qu'est-ce qui rendrait ta recommandation fausse dans un an ?

À retenir

1. Traduire chaque ligne de fiche en capacité chiffrée.
2. Le goulot dépend de la charge.
3. Les débits théoriques ne sont pas les débits réels.
4. Comparer capacité et prix.
5. Valider par des mesures avant d'acheter.

À retenir

Le schéma de la mission



Félicitations

Félicitations

Tu as terminé **HEF-040 — Architecture des ordinateurs**. Tu sais lire les bits, suivre le cycle du processeur, raisonner sur la hiérarchie mémoire, le cache, les entrées/sorties, les interruptions et la pile, et décoder une fiche technique pour trouver le goulot.
→ Prochain module : **HEF-050 — Systèmes d'exploitation**

MISSION ACCOMPLIE

+120 XP

Badge débloqué : Architecte matériel

Fin du module HEF-040. Bravo !

