

Optimizare Componente Calculator

Mihai Oșan
Inginerie Software
Anul I

Decembrie 2023

Descriere Problemă

Context:

Un student pasionat de gaming și tehnologie, dar limitat de constrângeri bugetare, se angajează într-un demers ambițios de a-și construi un calculator personalizat. Această inițiativă, este ghidată de principiul că fiecare leu investit trebuie să maximizeze valoarea totală. În acest context, obiectivul central al studentului este de a atinge un echilibru optim între performanță și cost, o sarcină complicată de varietatea și complexitatea opțiunilor disponibile pe piața actuală de componente pentru calculatoare.

Cerință:

Să se construiască un calculator care maximizează performanța (potrivit unui scor cumulat) în timp ce menține costurile totale sub un buget prestabilit. Se dorește alegerea componentelor calculatorului: procesor (CPU), placă de bază (MB), memorie (RAM), spațiu de stocare (Storage), placă grafică (GPU), sursă de alimentare (PSU). Fiecare componentă are un cost asociat și contribuie la performanța totală a calculatorului. În procesul de alegere al componentelor se va lua în considerare compatibilitatea dintre acestea.

Rezolvare

Variabile de Decizie:

Variabile Binare

Fiecare componentă disponibilă are o variabilă binară asociată, indicând dacă aceasta este selectată (1) sau nu (0):

$$\begin{aligned}CPU_i &\in \{0, 1\}, \quad \forall i \in CPU_df \\GPU_i &\in \{0, 1\}, \quad \forall i \in GPU_df \\MB_i &\in \{0, 1\}, \quad \forall i \in MB_df \\RAM_i &\in \{0, 1\}, \quad \forall i \in RAM_df \\PSU_i &\in \{0, 1\}, \quad \forall i \in PSU_df \\Storage_i &\in \{0, 1\}, \quad \forall i \in Storage_df\end{aligned}$$

Funcția obiectiv:

Maximizarea scorului total al componentelor

$$\begin{aligned}&Maximize(\sum_{i \in CPU_df} (scoreCPU_i \times CPU_i) \\&+ \sum_{i \in GPU_df} (scoreGPU_i \times GPU_i) \\&+ \sum_{i \in RAM_df} (scoreRAM_i \times RAM_i) \\&+ \sum_{i \in Storage_df} (scoreStorage_i \times Storage_i))\end{aligned}$$

Constrângeri:

1. Bugetul:

Totalul prețurilor componentelor selectate trebuie să fie mai mic sau egal cu bugetul.

$$\begin{aligned}&\sum_{i \in CPU_df} (priceCPU_i \times CPU_i) \\&+ \sum_{i \in GPU_df} (priceGPU_i \times GPU_i) \\&+ \sum_{i \in MB_df} (priceMB_i \times MB_i) \\&+ \sum_{i \in RAM_df} (priceRAM_i \times RAM_i) \\&+ \sum_{i \in PSU_df} (pricePSU_i \times PSU_i) \\&+ \sum_{i \in Storage_df} (priceStorage_i \times Storage_i) \leq budget\end{aligned}$$

2. Compatibilitate CPU și MB:

Asigurarea compatibilității între socket-ul CPU-ului și cel al plăcii de bază.

$$\begin{aligned}CPU_i + MB_j &\leq 1 \iff socketCPU_i \neq socketMB_j, \\ \forall i \in CPU_df, j \in MB_df\end{aligned}$$

3. Compatibilitate Form Factor MB și PSU:

Asigurarea compatibilității form factor-ului între placa de bază și unitatea de alimentare.

$$MB_i + PSU_j \leq 1 \iff form_factor MB_i \neq type PSU_j, \\ \forall i \in MB_df, j \in PSU_df$$

4. Limita TDP pentru CPU și GPU:

Totalul TDP al CPU și GPU nu trebuie să depășească capacitatea PSU.

$$\sum_{i \in CPU_df} (tdpCPU_i \times CPU_i) \\ + \sum_{i \in GPU_df} (tdpGPU_i \times GPU_i) \\ \leq \sum_{i \in PSU_df} (wattagePSU_i \times PSU_i)$$

5. Puterea recomandată pentru GPU acoperită de PSU:

Fiecare GPU selectat trebuie să aibă puterea recomandată acoperită de PSU.

$$\sum_{i \in GPU_df} (recommendedwGPU_i \times GPU_i) \\ \leq \sum_{i \in PSU_df} (wattagePSU_i \times PSU_i)$$

6. Capacitatea maximă a RAM:

Capacitatea totală de RAM nu trebuie să depășească capacitatea maximă a plăcii de bază selectate.

$$\sum_{i \in RAM_df} (gbRAM_i \times modulesRAM_i \times RAM_i) \\ \leq \sum_{i \in MB_df} (max_memoryMB_i \times MB_i)$$

7. Compatibilitatea numărului de module RAM cu sloturile Plăcii de bază:

Numărul de module RAM selectate nu trebuie să depășească numărul de sloturi RAM disponibile pe placa de bază selectată.

$$\sum_{i \in RAM_df} (modulesRAM_i \times RAM_i) \\ \leq \sum_{i \in MB_df} (num_slotsMB_i \times MB_i)$$

8. Alegerea unei Singure Componente din Fiecare Categorie:

Exact o componentă din fiecare categorie (CPU, GPU, MB, PSU, Storage, RAM) trebuie să fie selectată.

$$\begin{aligned}\sum_{i \in CPU_df} (CPU_i) &= 1 \\ \sum_{i \in GPU_df} (GPU_i) &= 1 \\ \sum_{i \in MB_df} (MB_i) &= 1 \\ \sum_{i \in PSU_df} (PSU_i) &= 1 \\ \sum_{i \in Storage_df} (Storage_i) &= 1 \\ \sum_{i \in RAM_df} (RAM_i) &= 1\end{aligned}$$

9. Constrângeri de Buget pentru Fiecare Componentă:

Prețul fiecărei componente (CPU, GPU, MB, RAM, PSU, Storage) nu trebuie să depășească un anumit procent din bugetul total.

$$\begin{aligned}\sum_{i \in CPU_df} (CPU_i \times Pret_CPU_i) &\leq Buget \times \left(\frac{Max_CPU_Pct}{100} \right) \\ \sum_{i \in GPU_df} (GPU_i \times Pret_GPU_i) &\leq Buget \times \left(\frac{Max_GPU_Pct}{100} \right) \\ \sum_{i \in MB_df} (MB_i \times Pret_MB_i) &\leq Buget \times \left(\frac{Max_MB_Pct}{100} \right) \\ \sum_{i \in RAM_df} (RAM_i \times Pret_RAM_i) &\leq Buget \times \left(\frac{Max_RAM_Pct}{100} \right) \\ \sum_{i \in PSU_df} (PSU_i \times Pret_PSU_i) &\leq Buget \times \left(\frac{Max_PSU_Pct}{100} \right) \\ \sum_{i \in Storage_df} (Storage_i \times Pret_Storage_i) &\leq Buget \times \left(\frac{Max_Storage_Pct}{100} \right)\end{aligned}$$