

Troubleshooting Guide: GPU & AI Výpočty

Tento průvodce slouží k diagnostice a řešení nejčastějších technických problémů spojených s provozem [Small Language Models](#), [CUDA](#) a [Dockeru](#).

1. Diagnostické kolečko (První pomoc)

Dříve než začnete měnit konfiguraci, proveďte tyto tři kroky:

- **Ověření HW:** Je karta správně detekována systémem? (`nvidia-smi` nebo Správce zařízení).
- **Kontrola VRAM:** Není paměť zaplněna jinou aplikací (prohlížeč, jiný model)?
- **Restart služeb:** U Dockeru `sudo systemctl restart docker`, u Windows restart aplikace Ollama/LM Studio.

2. Problémy s pamětí (VRAM)

Chyba: "CUDA Out of Memory (OOM)"

K této chybě dochází, když se model (nebo jeho kontext) nevejde do paměti grafické karty.

- **Příčina:** Model je příliš velký nebo máte nastavené příliš dlouhé „Context Window“.
- **Řešení:** * Použijte vyšší stupeň **kvantizace** (např. místo Q8 zvolte Q4_K_M).
 - Snižte parametr `num_ctx` v konfiguraci modelu.
 - Povolte **GPU offloading** (přesun části vrstev modelu do systémové RAM – podporuje např. llama.cpp).

3. Problémy s ovladači a CUDA

Chyba: "NVIDIA-SMI has failed because it couldn't communicate"

- **Příčina:** Verze ovladače v jádře (kernel) neodpovídá nainstalovaným knihovnám. Často po aktualizaci OS bez restartu.
- **Řešení:** Restartujte stroj. Pokud problém přetrvává, přeinstalujte ovladač pomocí [Studio Driveru](#).

Chyba: "RuntimeError: CUDA error: no kernel image is available"

- **Příčina:** Vaše verze PyTorch/TensorFlow byla zkompileována pro novější (nebo starší) architekturu GPU, než kterou máte.
- **Řešení:** Zkontrolujte **Compute Capability** vaší karty a nainstalujte odpovídající verzi knihovny.

4. Problémy v Docker prostředí

Příznak	Možná příčina	Řešení
Kontejner nevidí GPU	Chybí <code>--gpu</code> <code>all</code> nebo <code>runtime</code> v Dockeru.	Instalujte Container Toolkit.
Pomalý výkon v Dockeru	Používá se CPU místo GPU.	Ověřte přítomnost knihovny <code>libnvidia-ml.so</code> v kontejneru.
Chyba „Unknown runtime: nvidia“	Docker daemon není nakonfigurován.	Spusťte <code>nvidia-ctk runtime configure --runtime=docker</code> .

5. Stabilita a přehřívání (Throttling)

Pokud výkon modelu po čase klesá (klesá počet tokenů za sekundu), pravděpodobně dochází k **Thermal Throttlingu**.

- **Diagnostika:** Sledujte teplotu příkazem:

```
watch -n 1 nvidia-smi
```

- **Meze:** Pokud teplota překročí **85°C**, karta sníží takt.
- **Řešení:** * Vyčistěte chlazení.
 - Nastavte agresivnější křivku ventilátorů (Fan Curve).
 - Snižte **Power Limit** (např. `nvidia-smi -pl 200`), což sníží teplotu při minimálním dopadu na výkon AI.

6. Užitečné logy pro hlášení chyb

Při hlášení chyby na support přiložte výstupy z:

- **Linux:** `dmesg | grep -i nvidia` a `journalctl -u docker`
- **Windows:** Prohlížeč událostí (Event Viewer) → Protokoly systému.
- **AI Nástroje:** Logy v `~/.ollama/logs` nebo konzole v LM Studio.

— **Související rozcestník:**

- [Správa ovladačů](#)
- [Verze CUDA a kompatibilita](#)
- [Optimalizace Small Language Models](#)

— **Autor:** @L3_Support_Team **Poslední aktualizace:** 04. 01. 2026

From:
<https://www.serviceit.cz/> - IT ENCYKLOPEDIE

Permanent link:
https://www.serviceit.cz/doku.php?id=it:reseni_potizi_s_grafikou

Last update: **2026/01/04 16:47**



