

AI Core vs Multi-Core Systems: Detaljna Analiza

Autor: O.H.

Repo: GitHub AI Core vs Multi-Core Analysis

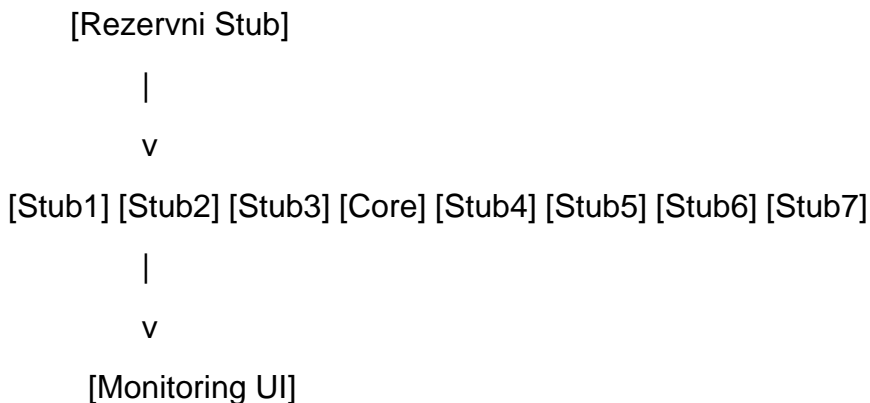
1. Uvod

- Definicija AIOS Core, AI stubova i Rezervnog stuba.
- Cilj: poredenje AIOS Core sistema sa tradicionalnim multi-core sistemima.

2. Arhitektura AIOS kernela

- Centralni Kernel Core: koordinacija stubova, globalni nadzor, scheduling, prikupljanje telemetrije.
- 7 modularnih AI stubova: predikcija gresaka, optimizacija resursa, paralelna obrada.
- Rezervni stub: watchdog, self-healing, predikcija padova, rollback.
- Tok podataka: Stubovi -> Core -> Rezervni stub -> Core/Stubovi -> Monitoring UI.

ASCII Dijagram



3. Prednosti AIOS Core vs Quad-Core

- AIOS Core nadmasuje u AI-driven workload-u i self-healingu.
- Quad-core ima prednost u raw CPU paralelizaciji.

4. Mane AIOS Core-a

- Softverski limit CPU cycles.
- Latencija zbog AI modela.
- Hardware crash utjece na sve stubove.
- Ceiling pri ekstremnoj paralelizaciji.

5. Strategije unapredjenja

- Dinamicko upravljanje stubovima.
- Laki AI modeli.
- Asinhrona komunikacija.
- Rezervni stub kao load balancer.
- Inteligentno batch-iranje i throttling.
- Hardware-assisted optimizacija.
- Prediktivna skalabilnost stubova.
- Konkretni primeri implementacije/pseudokod.

6. Simulacija izvršenja zadatka (T1-T12)

- Raspodela zadataka po stubovima.
- Rezervni stub intervenise po potrebi.
- Core rebalansira workload.
- Monitoring UI prikazuje status i self-healing akcije.
- Dodatak: primeri stvarnih AI taskova.

7. Performanse u procentima

- Paralelizacija, Latencija, AI predikcija, Self-healing, Skalabilnost, CPU overhead.
- Objašnjenje metodologije testiranja.

8. Zaključak

- AIOS Core je inovativni, modularni, samopopravljajući sistem.
- Idealno za AI-driven, real-time i samopopravljajuće sisteme.

9. Reference

- GitHub repo.
- Napomena: originalan rad, bez plagijata.