

# 1 Šprint 2

---

## 1.1 Analýza možnosti paralelizácie výpočtov

### 1.1.1 Využitie školského superpočítača

Spracovanie 3D robotického futbalu je pomerne náročné na výpočtové prostriedky, preto sme sa rozhodli preskúmať možnosti jeho paralelizácie. Pre projekt môže priniesť možnosť paralelného spúšťania a testovania hráča úžitok vo forme rýchleho získavania väčšieho množstva relevantných údajov a skrátenia času pri opätovnom spúšťaní podobných scenárov. Týmto priamo otvára dvere rýchlejšej simulácii veľkého množstva scenárov a strojovému učeniu hráča.

Preto sme sa rozhodli preskúmať ako jedno z riešení paralelizácie využitie superpočítača dostupného na STU FIIT.

#### Školský superpočítač

Na fakulte je dostupný pre náročné paralelné výpočty nižšie popísaný hardware - jedná sa o sieťovo prepojené zariadenia nasledovnej konfigurácie:

- 1 x HP DL160
  - 2 x Intel Xeon L5520 (4x2.26GHz)
  - 24GB DDR RAM
  - 10Gb Ethernet
- 16 x HP BL460c G6
  - 2 x Intel Xeon X5570 (4x2.93GHz)
  - 36GB DDR RAM
  - 1Gb Ethernet
- zdieľané diskové pole HP StorageWorks 2000sa (48TB)

Pre projekt podstatný software:

- Debian GNU/Linux 6.0
- Java 1.6.0\_24

#### Použiteľné technológie

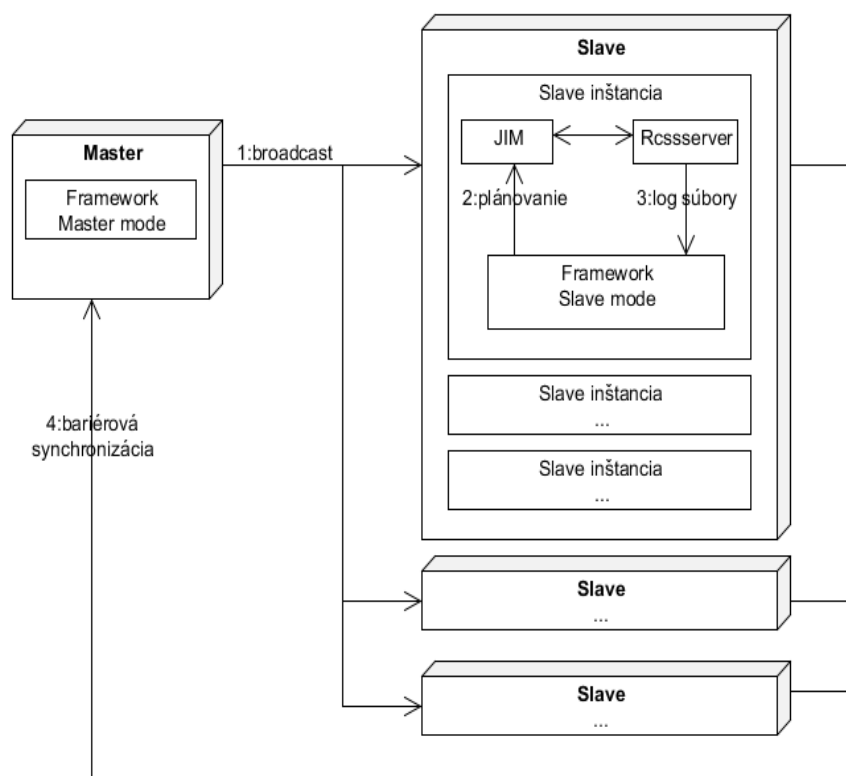
	Výhody	Nevýhody
<b>MPI</b>	-Priamočiare riešenie -Dobrá zrozumiteľnosť -Rýchle pochopenie architektúry začiatočníkom	-Žiadne skúsenosti v tíme s daným riešením na viacerých zariadeniach -Nie natívne riešenie pre Javu
<b>Hadoop</b>	-Dobré skúsenosti jedného člena tímu s daným riešením na školskom superpočítači -Napísaný v Jave	-Nie je stavaný na tento druh úloh -Zbytočné režijné náklady navyše -Ťažšie pochopiteľný pre začiatočníka

Tabuľka 1: Porovnanie použiteľných technológií

## Návrh na paralelizáciu výpočtov

Po konzultácii s Ing. Petrom Lackom, PhD sme dospeli k záveru, že na pre paralelizáciu behov hráča JIM bude vhodnejšie použitie technológie MPI z nasledovných dôvodov. Architektúra daného riešenia bude vyzerat' nasledovne (Obrázok 1):

- Framework bude môcť byť spustený v master a slave móde.
- Jedno zariadenie bude mať spustený master framework, ktorý bude spojený so všetkými slave frameworkami na slave zariadeniach.
- Každý slave framework bude spustený s jemu priradeným rcserverom a hráčom (hráčmi).
- Vhodnosť počtu slave inštancií (framework+hráči+server) bude overená až záťažou na jednotlivých uzloch, teoreticky sa ale ponúka možnosť spustiť ich až 128, pretože máme k dispozícii 16 zariadení x 2 procesory x 4 jadrá. Tento model ale predstavuje potrebu riešiť spustenie viacerých slave inštancií na jednom zariadení.



Obrázok 1: Popis komunikácie distribuovaného spracovania

Jedna slučka evolučného algoritmu môže vyzerat' nasledovne (Obrázok 1):

1. Master framework (pomocou MPI) spôsobom broadcast rozistribuuje údaje jednotlivým slave frameworkom. Obsahom údajov môžu byť príkazy, skripty, alebo zmeny pohybov.
2. Každý slave framework na základe údajov od master frameworku naplánuje hráčovi vykonávanie pohybov.

3. Slave framework bude spracovávať údaje získané zo servera. Na základe nich môže vyhodnotiť úspešnosť správania sa hráča (vypočíta fitness funkciu).
4. Master framework prostredníctvom bariérovej synchronizácie zozbiera všetky výsledky jednotlivých slave frameworkov. Na základe nich potom vygeneruje novú generáciu jedincov, ktorých rozdistribuuje (krok 1).

Identifikované problémy s daným riešením:

- potreba riešiť inicializáciu slave inštancií na diaľku,
- potreba vyriešiť komunikáciu s viacerými slave inštanciami na jednom zariadení,
- schopnosť reagovať na neštandardné správanie rcssservera,
- potreba implementácie master a slave módu frameworku,
- vzhľadom na nenatívnosť MPI v Jave môže vzniknúť neprehľadný kód,
- technológia MPI na superpočítači doposiaľ nie je zavedená,
- otázna je reálna potreba prínosu riešenia pre projekt.