

# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

## FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

Dokumentace 3. projektu pro předmět PRL 2015/2016  
Daniel Žůrek – xzurek12

### 1. Rozbor a analýza algoritmu

Carry Look Ahead Parallel Binary Adder (CLAPBA) je paralelní algoritmus pro součet dvou binárních čísel délky  $n$ . Jedná se o algoritmus, který reprezentuje sčítačku s predikcí přenosu. Výhodou algoritmu je, současné provedení výpočtu přenosů mezi jednotlivými řády. Tímto je docíleno snížení celkové časové náročnosti. Topologie procesorů je binární strom, který je tvořen  $(\log n) + 1$  úrovněmi,  $n$  listy a celkový počet procesorů je  $2n - 1$ .

Výpočet začíná od listových uzlů, kdy se provede bitové sčítání, a výsledky jsou určeny dle následující tabulky:

<b>B</b>	<b>0</b>	<b>1</b>
<b>0</b>	s	p
<b>1</b>	g	g

Tabulka 1: Výsledky bitového sčítání

Následně je provedena operace *UpSweep*, tedy výpočet sumy prefixů s časovou složitostí  $\Theta(\log n)$ . Tato operace je podobná operaci *Reduce*, rozdíl je však v tom, že při *UpSweep* si každý uzel pamatuje mezisoučet. Po dosažení kořenového uzlu započne operace *DownSweep*. Kořenovému procesoru se přiřadí neutrální prvek a následně si v každé úrovni směrem k listovým uzlům procesory provádějí výpočet paralelně. Průchod stromem trvá  $\log n$  kroků. Jedná se tedy o algoritmus s logaritmickou časovou složitostí.

$\odot$	<b>s</b>	<b>p</b>	<b>g</b>
<b>s</b>	s	s	s
<b>p</b>	s	p	g
<b>g</b>	g	g	g

Tabulka 2: Výsledky operací pro *UpSweep* a *DownSweep*

Počet procesorů:

$$p(n) = 2n - 1$$

Časová složitost:

$$t(n) = \Theta(\log n)$$

Celková cena:

$$p(n)t(n) = \Theta(n(\log n))$$

## 2. Popis implementace

Začátek programu tvoří kromě inicializace knihovny Open MPI provedení výpočtu pro získání ranku levého a pravého syna k danému procesoru a výpočet ranku prvního listového uzlu. Obě sčítané binární čísla jsou načteny do dvou vektorů. V případě, že jedno z těchto čísel je kratší (menší) než druhé, je toto číslo zleva doplněno nulami.

Procesor s rankem 0 (kořen stromu) si otevře vstupní soubor *numbers* a načte si vstupní čísla. Obě binární čísla jsou uloženy ve strukturách typu *vector*, které poskytuje jazyk C++. Oba vektory jsou zaslány na všechny listové uzly.

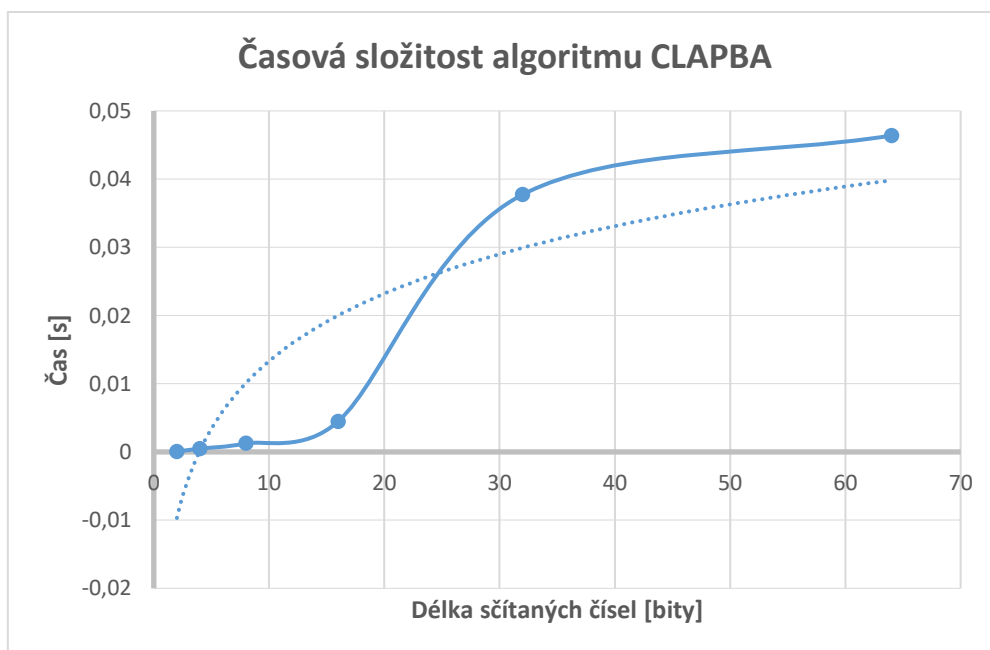
Kořenový uzel dále přijme zprávu od svých dvou synů, provede výpočet z přijatých hodnot, který odpovídá *Tabulce 2*. Odpovídá-li výsledek hodnotě `GENERATE(přetečení)`, vytiskne se na standartní výstup text *overflow*. Pravému synu je zpět zaslána neutrální hodnota `PROPAGATE`, levému synu je zaslána hodnota vypočtena opět pomocí operace *Tabulky 2* z této neutrální hodnoty a hodnoty získané od pravého syna.

Nelistové uzly vypočtou rank svého otce a následně čekají na přijetí hodnot od svých synů (blokuující čekání). Opět se provede výpočet pomocí *Tabulky 2*, výsledek je zaslán otcovi a počká se na jeho odpověď. Pravému synu je zaslána nazpět hodnota od otce, pro levý je opět vypočtena nová hodnota.

Listové uzly si alokují dva buffery pro přijetí binárních čísel od nultého procesoru. Velikost polí je  $(numproc + 1) / 2$ , kde *numproc* je celkový počet procesorů. Po obdržení obou vektorů je nutné, aby každý listový procesor indexoval správnou pozici. Index se vypočte na základě ranku procesoru mínus rank prvního listového procesoru. Výsledek sčítání odpovídá *Tabulce 1*. Po odeslání výsledku otcovskému procesoru se počká na jeho odpověď. Je-li přijatá hodnota rovna `GENERATE`, je do pomocné proměnné *fixingValue* přiřazena 1. Celkový výsledek se spočítá jako  $(b1 + b2 + fixingValue) \bmod 2$ .

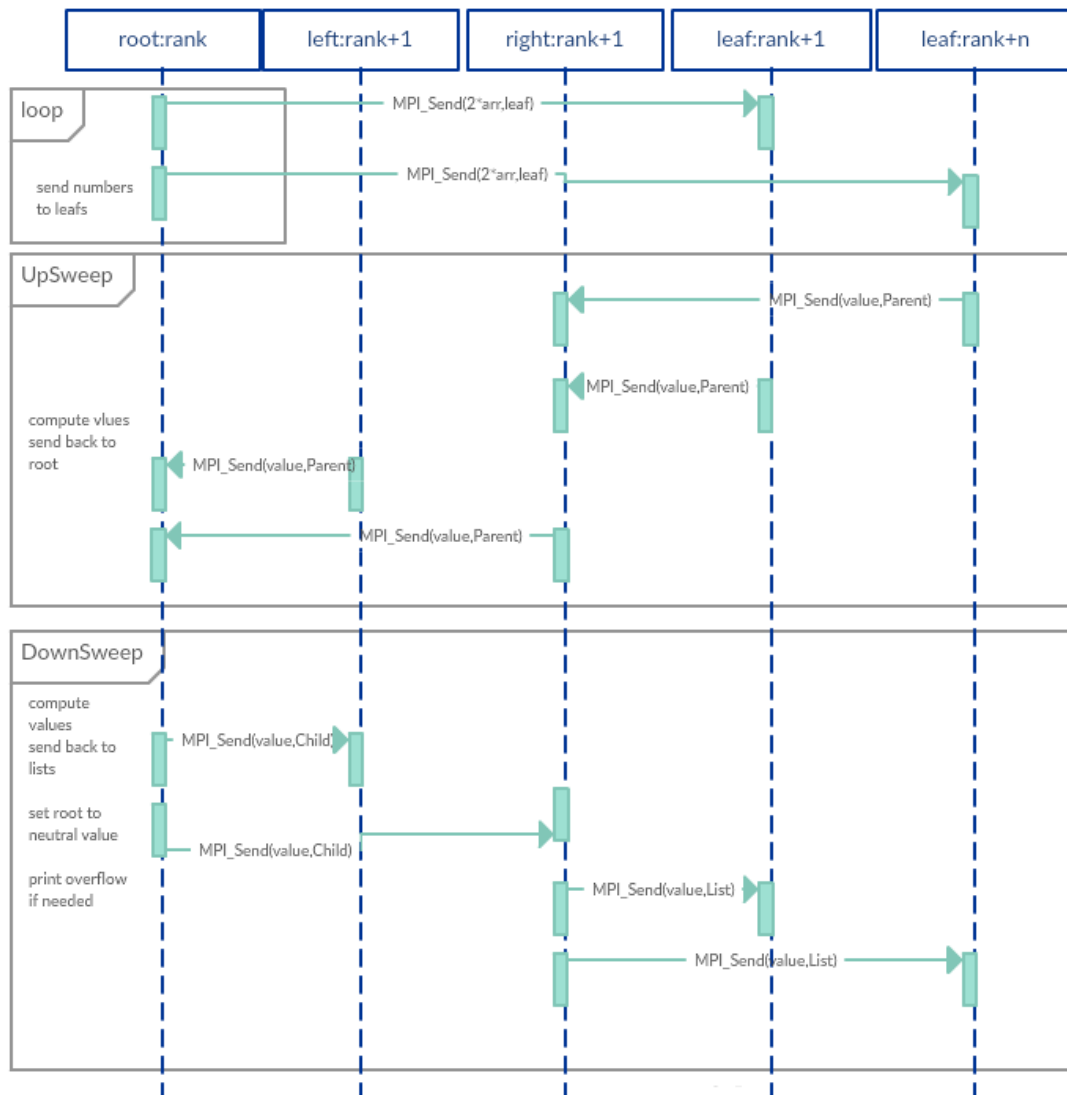
## 3. Experimenty

Experimenty byly prováděny na virtuálním stroji se systémem Manjaro 4.1.15-1, kde každé měření bylo provedeno 10x a výsledek zprůměrován.



Cílem experimentů je dosáhnout logaritmické časové složitosti. Pro dosažení co nejlepších výsledků, není do měření zahrnuto, rozesílání hodnot na listové procesory, inicializace MPI, nebo výpis na terminál.

## 4. Komunikační protokol



## 5. Závěr

V projektu byl implementován paralelní algoritmus Carry Look Ahead Parallel Binary Adder pro součet dvou binárních čísel. Dále byla s tímto algoritmem provedena série testů, která měla potvrdit předpokládanou logaritmickou časovou složitost. Výsledky testů se oproti předpokladu mírně liší. To je nejspíše zapříčiněno experimentováním na virtuálním stroji a režii při přepínání kontextu mezi jednotlivými procesy. Funkčnost projektu byla ověřena na serveru Merlin.