Technická univerzita v Košiciach
Fakulta elektrotechniky a informatiky

Redukcia časovej náročnosti spracovania dát nástrojom SLAmeter

Diplomová práca

Príloha B

SYSTÉMOVÁ PRÍRUČKA Evaluatorik

Študijný program:	Informatika
Študijný odbor:	Informatika
Školiace pracovisko:	Katedra počítačov a informatiky (KPI)
Vedúci práce:	Ing. Adrián Pekár, PhD.
Konzultant:	Ing. Ján Juhár

Košice 2015 **Bc. Pavol Beňko**

Obsah

Zoznam obrázkov

1 Funkcia programu

Evaluatorik je reprezentovaný pomocou desktopovej aplikácie a zabezpečuje vyhodnotenie sieťových charakteristík. Tvorí súčasť architektúry meracieho nástroja SLAmeter.

Pre vyhodnotenie sieťových charakteristík využíva záznamy exportované do databázy pomocou zhromažďovacej aplikácie. Záznamy do neho prúdia využitím exportovacej aplikácie. Evaluatorik v sebe obsahuje sadu modulov, ktoré reprezentujú jednotlivé sieťové charakteristiky. Prepojenie medzi touto aplikáciou a Webovou aplikáciou je realizované pomocou databázovej služby Redis.

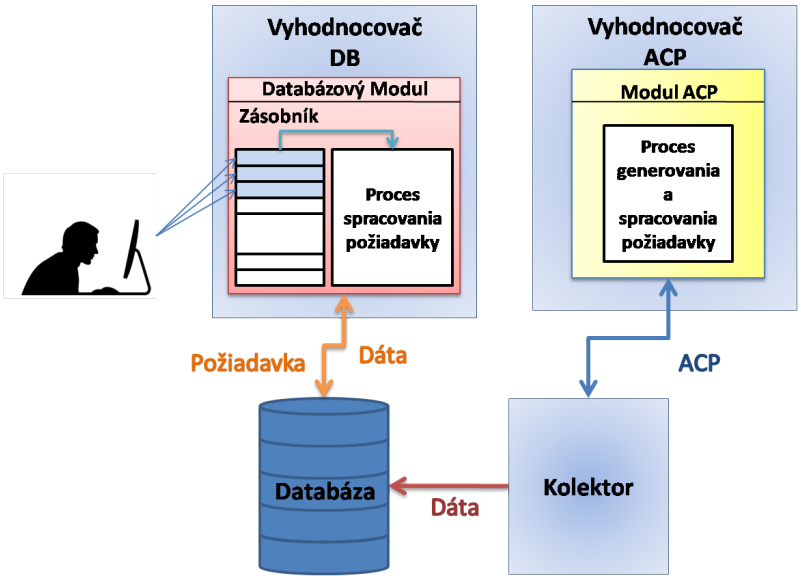
Požiadavky zasielané týmto spojením majú format JSON objektu. Ich bližší popis formátu bude zobrazený v nasledujúcich kapitolách. Pre získanie záznamov z databázy sa používa aplikačné rozhranie databázy Mongo.

1.1 Architektúra Vyhodnocovača

Architektúra vyhodnocovača pozostáva z rámca vyhodnocovača ktorého úlohou je poskytnúť prepojenie modulov s databázou respektíve grafického používateľského rozhrania, webu.

Moduly komunikujú s databázou prostredníctvom BSON správ. Získané dáta sú následne spracované modulmi, pričom modul poskytne výsledok po vyhodnotení. Ten je zaslaný vo forme JSON správy do Redis databázy, z ktorej je následne tento výsledok prebraný webovým rozhraním. Tento výsledok je následne zobrazený vo forme grafu, respektíve vo forme hodnoty.

Architektúra vyhodnocovača je zobrazená na nasledujúcom obrázku, kde došlo k rozdeleniu databázového a ACP vyhodnocovača.



Obr. 1 – 1 Architektúra Vyhodnocovača

2 Opis štruktúry programu

2.1 Projekt

Program Evaluatorik obsahuje tieto adresáre:

- **build** - obsahuje .class súbory tried,
- **dist** - obsahuje spustiteľný súbor (.jar) aplikácie,
- **doc** - priečinok obsahujúci dokumentáciu k programu,
- **lib** - obsahom sú potrebné knižnice,
- **nbproject** - NetBeansom generované súbory,
- **src** - priečinok so zdrojovými textami,
- **.git** - súbory potrebné pre aktualizáciu projektu na GIT.

2.2 Zdrojové texty

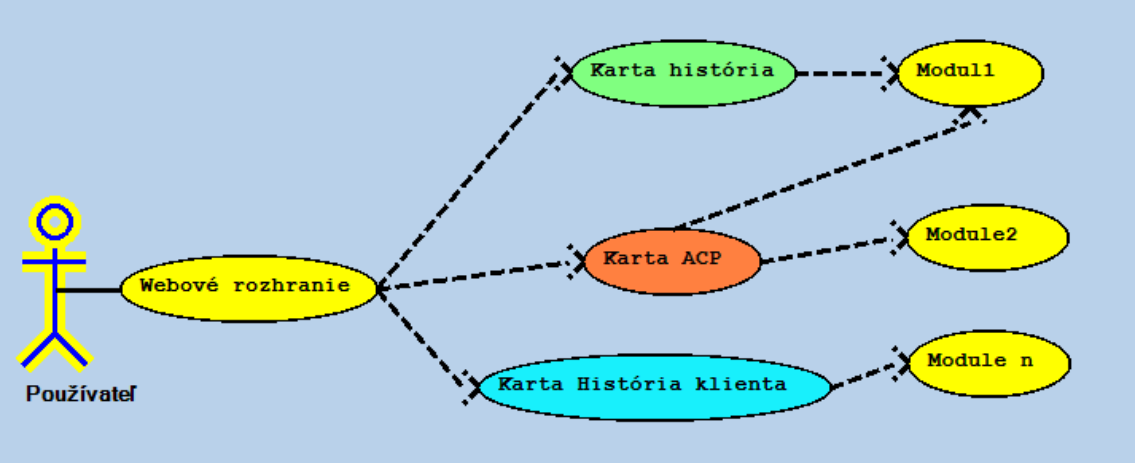
Zdrojové kódy sú delené do týchto balíkov

- **Evaluatorik**
  - **connections** triedy umožňujúce pripojenie na databázu a službu Redis
  - **evaluatorik** triedy umožňujúce spustenie aplikácie, spracovanie konfiguračného súboru
  - **module** triedy reprezentujúce abstraktný modul
  - **module.calculation** triedy reprezentujúce výpočtové algoritmy modulov, ktoré je možné použiť na viacero miestach.
  - **module.modules** singleton triedy reprezentujúce moduly

2.3 Diagram prípadov použitia

Na nasledujúcom diagrame 2 – 1 je zobrazený prípad použitia modulov vyhodnocovacej aplikácie. Používateľ prichádza na adresu webového grafického rozhrania, pričom to v okamihu prístupu zašle požiadavky, ktoré sú uložené v databázovej službe Redis.

Požiadavky sú zaslané na všetky moduly, ktoré sú umiestnené na danej stránke. V prípade **karty ACP** v diagrame použitia, budú požiadavky zaslané na **Modul1** a **Modul2**. Po ich vyhodnotení je výsledok zaslaný naspäť do Redis databázy, z ktorej sú prebrané webovým rozhraním a následne zobrazené.

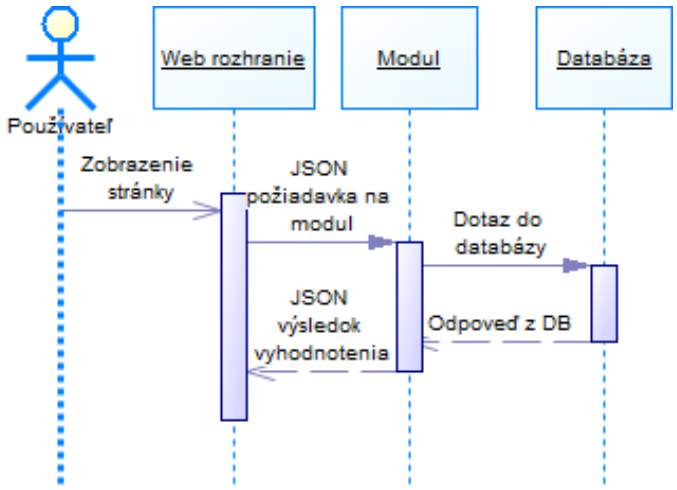


Obr. 2 – 1 Diagram prípadov použitia

2.4 Sekvenčný diagram

Mechanizmus komunikácie užívateľa s webovým rozhraním a tým aj medzi webovým rozhraním a vyhodnocovačom je možné vyobraziť aj pomocou sekvenčného diagramu 2 – 2. Popis uvedeného diagramu je nasledujúci. Používateľ zobrazuje webovú stránku s konkrétnymi modulom. Web generuje požiadavku a tú zasiela do Redis služby. Vyhodnocovač prijíma požiadavku z Redis a spúšťa konkrétny modul. Modul po

vyhodnotení odosiela výsledok do Redis služby. Web reaguje na prijatú hodnotu a zobrazuje ju vo forme grafu alebo štandardnej hodnoty.



Obr. 2 – 2 Sekvenčný diagram

3 Pridávanie modulov do nástroja Evaluatorik

1. Vytvoríme triedu v priečinku *module.modules*
2. Trieda modulu dedí vlastnosti z triedy *AbstractModule* a zároveň triedu implementujeme ako návrhový vzor jedináčik.
3. Implementujeme zdedenú abstraktnú metódu *run* do ktorej umiestnime samotnú logiku výpočtu
4. Implementujeme zdedenú abstraktnú metódu *getModuleName*, ktorá vráti meno modulu
5. Následne podľa mnou implementovaných modulov vykonáme pridanie modulu do načúvača Redis (trieda JedisListener). Pre potrebu spustenia modulov nachádzajúcich sa v konfiguračnom súbore realizujeme pridanie modulu do konfiguračného súboru aplikácie a taktiež upravíme párser konfiguračného súboru *Config*.

4 Popis tried, metód a premenných

4.1 Balík connections

Trieda **DatabaseConnection**

Globálne premenné

- MongoClient **mongo** - manažér spojenia na databázu ,
- DB **database** - prístup k databáze,
- boolean **isConnected** - príznak pripojenia na databázu.

Metódy

boolean ***connect****(String host, String port, String name, String username, String password)*

metóda realizuje pripojenie na databázu:

- String host- adresa databázy,
- String port- port databázy,
- String name- názov databázy,
- String username- používateľské meno pre prístup k databáze,
- String password- používateľské heslo pre prístup k databáze

void ***disconnect()***

metóda realizuje odpojenie aplikácie od databázy

boolean ***isConnected()***

metóda vráti príznak pripojenosti k databáze

DB ***getDatabase()***

metóda vráti objekt k databáze

Trieda **JedisConnection**

Globálne premenné

- JedisConnection **connection**- získanie spojenia s redis službou,
- JedisPool **pool**- zásobník pre spojenia na Redis službu

Konštruktor

JedisConnection()

Metódy

boolean ***getConnection()***

metóda vytvárajúca singleton inštanciu triedy

void ***returnConnection(Jedis jedis)***

vrátenie Redis spojenia späť do zásobníka

- Jedis jedis- jedis spojenie

Trieda **JedisListener extends JedisPubSub**

Globálne premenné

- log- premenná určená pre loggovanie v triede

Metódy

void ***onMessage****(String channel, String message)*

metóda reaguje na prijatú správu

- String **channel**- názov kanála ,
- String **message**- správa určená pre kanál

void ***onSubscribe****(String channel, int subscribedChannels)*

metóda reaguje na zaregistrovanie kanála

- String **channel**- názov kanála ,
- String **message**- správa určená pre kanál

4.2 Balík evaluatorik

4.2.1 Trieda Config

Globálne premenné

- String **dbHost**- host pripojenie databázy
- String **dbPort**- port databázy
- String **dbName**- názov databázy
- String **dbLogin**- meno užívateľa pre prihlásenie do databázy
- String **dbPassword**- heslo pre databázu
- String **redisHost**- host pre pripojenie na Redis
- String **redisPort**- port pre pripojenie na Redis
- String **redisPoolSize**- veľkosť zásobníka spojení
- boolean **AmountOfTransferredData**- príznak spustenia modulu
- boolean **BandwidthHistoryTrend**- príznak spustenia modulu
- boolean **AverageOfTransferredDataPacket**- príznak spustenia modulu
- boolean **MaximumDownloadUpload**- príznak spustenia modulu
- boolean **AmountOfTransferredDataPacket**- príznak spustenia modulu
- boolean **BandwidthHistoryTrendPacket**- príznak spustenia modulu
- boolean **AverageMiniTable**- príznak spustenia modulu

- boolean **HistoryTrendFlows**- príznak spustenia modulu
- boolean **AverageOfTransferredData**- príznak spustenia modulu
- boolean **HistoryTable**- príznak spustenia modulu
- boolean **AmountMiniTable**- príznak spustenia modulu
- boolean **NumberOfFlows**- príznak spustenia modulu
- boolean **PingTime**- príznak spustenia modulu
- boolean **TopUploader**- príznak spustenia modulu
- boolean **TopDownloader**- príznak spustenia modulu

Metódy

void ***parseXmlFile()***

metóda pársujúca XML konfiguračný súbor

4.2.2 Trieda Evaluatorik

Metódy

void **main**(String[] args)

hlavná metóda spúšťajúca celú aplikáciu

4.2.3 Trieda RunModules

Globálne premenné

- ArrayList<String >**listenedModules**- premenná uchováva mená spustených modulov
- **log**- premenná určená pre loggovanie v triede

Metódy

void **getAllModulesViaConfig()**

metóda načíta do premennej listu všetky moduly s príznakom true z konfiguračného súboru

void **getAllModulesViaReflection()**

metóda načíta do premennej listu všetky moduly z balíka module.modules

void **sub(ArrayList list)**

metóda realizuje spustenie listenera pre načítané moduly

- ArrayList **list**- list obsahujúci spustené moduly

4.3 Balík module

4.3.1 Trieda AbstractModule

Globálne premenné

- PoolRequest **requests**- zásobník pre požiadavky
- Jedis **jedis**- inštancia pre spojenie s Redis databázou
- PrepareStatement **prepareStatement**- inštancia pre potreby transformácie požiadavky na databázový dotaz.
- ArrayList<String >**requiredAttribute**- list povinných atribútov modulu
- ArrayList<String >**optionalAttribute**- list voliteľných atribútov modulu
- Logger **log**- premenná určená pre loggovanie v triede

Gettery a settery

ArrayList<String >**getRequiredAttribute()**

void **setRequiredAttribute(String... s)**

ArrayList<String>**getOptionalAttribute()**

void **setOptionalAttribute(String... s)**

void **returnJedisResources(Jedis jedis)**

PoolRequest **getPoolRequest()**

void **setPoolRequest(PoolRequest requests)**

Jedis **getJedis()**

void **setJedis(Jedis jedis)**

void **setPrepareStatement(PrepareStatement ps)**

PrepareStatement **getPrepareStatement()**

Metódy

abstract String **getModuleName()**

metóda pre vrátenie mena modulu

@Override

void **run()**

metóda vlákna v ktorej sa realizuje samotný výpočet modulu

4.3.2 Trieda MaxValues

Globálne premenné

- ArrayList<User>**values**- kolekcia užívateľov
- int **maxRecords**- maximálny počet záznamov v kolekcii

Konštruktor

MaxValues()

konštruktor bez parametrov vytvori kolekciu o veľkosti maximálne 5

MaxValues(int max)

konštruktor určujúci počet maximálnych hodnôt v kolekcii

Metódy

void **push**(User value) JSONArray **toArray()**

Gettery a settery

int **getMaxRecords()**

void **setMaxRecords**(int maxRecords)

4.3.3 Trieda MaxValues.User

Globálne premenné

- double **octetCount**- počet oktetov používateľa
- String **ipcka**- ip adresa používateľa

Konštruktor

User(double octetCount,String ipcka)

konštruktor realizuje vytvorenie objektu používateľa s parametrami počet oktetov a ip adresa

Metódy

int **compareTo**(User o)

- User **o**- porovnanie užívateľa o

Metóda realizujúca porovnávanie používateľov podľa octetCount

4.3.4 Trieda ModuleResponse

Globálne premenné

- Logger **log**- premenná určená pre loggovanie v triede
- SimpleDateFormat **sdf**- premenná pre inicializovanie formátu dátumu

Metódy

JSONObject **sentData**(String name, String status, String res)

- String **name**- meno modulu
- String **status**- status odpovede
- String **res**- odpoveď vo formáte string

metóda vráti JSONObjekt odpovede

JSONObject **sentDataDouble**(String name, String status, Double res)

- String **name**- meno modulu
- String **status**- status odpovede
- String **res**- odpoveď vo formáte Double

metóda vráti JSONObjekt odpovede

JSONObject **sentDataArray**(String name, String status, long[] res)

- String **name**- meno modulu
- String **status**- status odpovede
- String **res**- odpoveď vo formáte long[]

metóda vráti JSONObjekt odpovede

4.3.5 Trieda PoolRequest

Globálne premenné

- Queue<Request>**queue**- zásobník na prijaté požiadavky

Konštruktor

public **PoolRequest()**

konštruktor na vytvorenie zásobníka pre požiadavky

Metódy

void **add**(Request r)

- Request **r**- požiadavka uložená v objekte Request

Request **get()**

vyber požiadavky z kolekcie a súčasné zmazanie

boolean **isEmpty()**

overenie či je kolekcia prázdna

int **size()**

získanie veľkosti kolekcie

4.3.6 Trieda PoolRequest.Request

Globálne premenné

- String **id**- id modulu reprezentované menom
- String **request**- samotná prijatá požiadavka

Konštruktor

Request(String id,String request)

konštruktor pre vytvorenie požiadavky s menom modulu a samotnou požiadavkou

4.3.7 Trieda PrepareStatement

Globálne premenné

- QueryBuilder **query**- query použité na transformovanie požiadavky do databázového dotazu

Metódy

QueryBuilder **getStatement**(JSONObject object, AbstractModule m)

- JSONObject **object**- objekt požiadavky pre modul
- AbstractModule **m**- objekt samotného modulu

návratová hodnota QueryBuilder ktorý je použitý ako dotaz do databázy
ArrayList<byte[]>**JArrayToArrayB**(JSONArray array)

- JSONArray **array**- pole obsahujúce byte[] hodnoty

metóda transformuje hodnoty z JSONArray do ArrayList<byte[]>
Integer[] **JArraytoArrayI**(JSONArray array)

- JSONArray **array**- pole obsahujúce int hodnoty

metóda transformuje JSONArray do Integer[] boolean **chcekIPaddress**(String ip)

- String **ip**- ip adresa

overí správnosť zadanej ip adresy boolean **checkPort**(int i)

- int **i**- port

overí správnosť zadaného portu boolean **containRequiredAttribute**(JSONObject jo,AbstractModule m)

- JSONObject **jo**- objekt požiadavky pre modul
- AbstractModule **m**- objekt samotného modulu

overenie či jo obsahuje všetky povinné atribúty modulu m
JSONObject **removeNullValue**(JSONObject jo, AbstractModule m)

- JSONObject **jo**- objekt požiadavky pre modul
- AbstractModule **m**- objekt samotného modulu

zmazanie null hodnôt z jo objektu

4.4 Balík module.calculation

4.4.1 Trieda CalculateTransferedData

Globálne premenné

- Long **startInterval**- počiatočná hodnota intervalu v ktorom bude realizovaný výpočet
- Long **endInterval**- konečná hodnota intervalu v ktorom bude realizovaný výpočet
- Double **resutValue**- výsledná hodnota výpočtu

Metódy

Long **getTimeIntervalInSec()**

vráti počet sekúnd v zadanom intervale

void **processData**(Iterator<DBObject>result)

- Iterator<DBObject>**result**- výsledný iterátor získaný dopytom do databázy

metóda realizujúca výpočet počtu prenesených dát v oktetoch intervale

Gettery a settery

Long **getStartInterval**()

void **setStartInterval**(Long startInterval)

Long **getEndInterval**()

void **setEndInterval**(Long endInterval)

Double **getResutValue**()

void **setResutValue**(Double resutValue)

4.4.2 Trieda CalculateTransferedDataPacket

- Long **startInterval**- počiatočná hodnota intervalu v ktorom bude realizovaný výpočet
- Long **endInterval**- konečná hodnota intervalu v ktorom bude realizovaný výpočet
- Double **resutValue**- výsledná hodnota výpočtu

Metódy

Long **getTimeIntervalInSec()**

vráti počet sekúnd v zadanom intervale

void **processData**(Iterator<DBObject>result)

- Iterator<DBObject>**result**- výsledný iterátor získaný dopytom do databázy

metóda realizujúca výpočet počtu prenesených dát v paketoch intervale

Gettery a settery

Long **getStartInterval**()

void **setStartInterval**(Long startInterval)

Long **getEndInterval**()

void **setEndInterval**(Long endInterval)

Double **getResutValue**()

void **setResutValue**(Double resutValue)

4.5 Balík module.modules

Všetky moduly obsahujú rovnaké metódy a z toho dôvodu tieto opakujúce metódy uvediem jedenkrát. V moduloch uvediem ich globálne premenné a konštruktory. V prípade, že niektorý z modulov obsahuje pomocnú metódu uvediem ju podkapitole

daného modulu.

Metódy

AverageMiniTable getInstance()

metóda vráti inštanciu modulu

void run()

metóda vykonávajúca samotne spracovanie prijatých požiadaviek

String getModuleName()

metóda vráti meno modulu

4.5.1 Trieda CalculateTransferedData

Globálne premenné

- CalculateTransferedData **calculate**
- AmountMiniTable **AmountMiniTableInstance**

Konštruktor

CalculateTransferedData()

konštruktor na vytvorenie objektu modulu

4.5.2 Trieda AmountOfTransferredData

Globálne premenné

- CalculateTransferedData **calculate**
- AmountOfTransferredData **AmountOfTransferredData**

Konštruktor

AmountOfTransferredData()

konštruktor na vytvorenie objektu modulu

4.5.3 Trieda AmountOfTransferredDataPacket

Globálne premenné

- CalculateTransferedDataPacket **calculate**
- AmountOfTransferredDataPacket **AmountOfTransferredDataPacket**

Konštruktor

AmountOfTransferredDataPacket()

konštruktor na vytvorenie objektu modulu

4.5.4 Trieda AverageDownloadUpload

Globálne premenné

- CalculateTransferedData **calculate**
- AverageDownloadUpload **AverageDownloadUploadInstance**

Konštruktor

AverageDownloadUpload()

konštruktor na vytvorenie objektu modulu

4.5.5 Trieda AverageDownloadUploadPacket

Globálne premenné

- CalculateTransferedDataPacket **calculate**
- AverageDownloadUploadPacket **AverageDownloadUploadPacketInstance**

Konštruktor

AverageDownloadUploadPacket()

konštruktor na vytvorenie objektu modulu

4.5.6 Trieda AverageMiniTable

Globálne premenné

- AverageMiniTable **AverageMiniTableInstance**
- CalculateTransferedData **calculate**

Konštruktor

AverageMiniTable()

konštruktor na vytvorenie objektu modulu

4.5.7 Trieda BandwidthHistoryTrend

Globálne premenné

- CalculateTransferedData **calculate**
- BandwidthHistoryTrend **BandwidthHistoryTrendInstance**

Konštruktor

BandwidthHistoryTrend()

konštruktor na vytvorenie objektu modulu

Metódy

4.5.8 Trieda BandwidthHistoryTrendPacket

Globálne premenné

- CalculateTransferedDataPacket **calculate**
- BandwidthHistoryTrendPacket **BandwidthHistoryTrendPacketInstance**

Konštruktor

BandwidthHistoryTrendPacket()

konštruktor na vytvorenie objektu modulu

Metódy

Hashtable<Long[],Double>**getIntervals**(long timeS,long timeE)

metóda generuje heštabuľku intervalu času a Double hodnoty pre potreby uchovania poctu prenesených dát v danom intervale

4.5.9 Trieda HistoryTable

Globálne premenné

- HistoryTable **HistoryTableInstance**
- CalculateTransferedData **calculate**

Konštruktor

HistoryTable()

konštruktor na vytvorenie objektu modulu

Metódy

Hashtable<Long[],Double>**getIntervals**(long timeS,long timeE)

metóda generuje heštabuľku intervalu času a Double hodnoty pre potreby uchovania poctu prenesených dát v danom intervale

4.5.10 Trieda HistoryTrendFlows

Globálne premenné

- HistoryTrendFlows **HistoryTrendFlowsInstance**

Konštruktor

HistoryTrendFlows()

konštruktor na vytvorenie objektu modulu

Metódy

long **getNumberOfFlows**(AggregationOutput output, long time) metóda generuje pre určený čas počet tokov

4.5.11 Trieda MaximumDownloadUpload

Globálne premenné

- MaximumDownloadUpload **MaximumDownloadUploadInstance**
- CalculateTransferedData **calculate**

Konštruktor

MaximumDownloadUpload()

konštruktor na vytvorenie objektu modulu

4.5.12 Trieda NumberOfFlows

Globálne premenné

- NumberOfFlows **NumberOfFlowsInstance**

Konštruktor

NumberOfFlows()

konštruktor na vytvorenie objektu modulu

4.5.13 Trieda PingTime

Globálne premenné

- PingTime **PingTimeInstance**

Konštruktor

PingTime()

konštruktor na vytvorenie objektu modulu

4.5.14 Trieda TopDownloader

Globálne premenné

- CalculateTransferedData **calculate**
- TopUploader **topUploaderInstance**
- MaxValues **array**

Konštruktor

TopDownloader()

konštruktor na vytvorenie objektu modulu

4.5.15 Trieda TopUploader

Globálne premenné

- CalculateTransferedData **calculate**
- TopUploader **topUploaderInstance**
- MaxValues **array**

Konštruktor

TopUploader()

konštruktor na vytvorenie objektu modulu