

# Interné fungovanie EDA Servera

EDA Server dokáže obsluhovať viacero klientov naraz vďaka paralelizácii na výpočtovej úrovni aj na úrovni prístupu k databáze. V prípade, že je klient schopný zasielať požiadavky na EDA Server paralelne, tieto požiadavky sú na EDA Serveri taktiež spracované paralelne.

## Session

Postupnosť príkazov, ktoré sú vykonávané serializovane sa v interných štruktúrach EDA Servera nazýva **session**. Session na EDA Serveri vzniká prvou požiadavkou na EDA Server v aktuálne vykonávanej postupnosti príkazov. Na jej identifikáciu sa používa id inštancie prvého ESL skriptu, v reakcii volaní – v prípade, že klient používa ESL runtime, resp. id pridelené objektu *Session* – v prípade, že klientom je J-EDA. Pripojenie na EDA cez Excel, resp. VBA, nie je schopné paralelnej komunikácie s EDA Serverom, a preto používa vždy len jednu session, ktorá má id 0.

Z prostredia ESL, je možné získať id aktuálnej session pomocou externej funkcie [EDA\\_GetSessionId](#).

Na maximálne využitie výpočtového výkonu môžu byť niektoré požiadavky na EDA Server interne rozložené na viacero astí, ktoré taktiež budú vykonané paralelne. Medzi takéto požiadavky patria dávkové operácie (Batch) a paralelné načítanie vektorov v EDA-L cez funkciu **%LoadVektors**.

## Spracovateľské vlákna

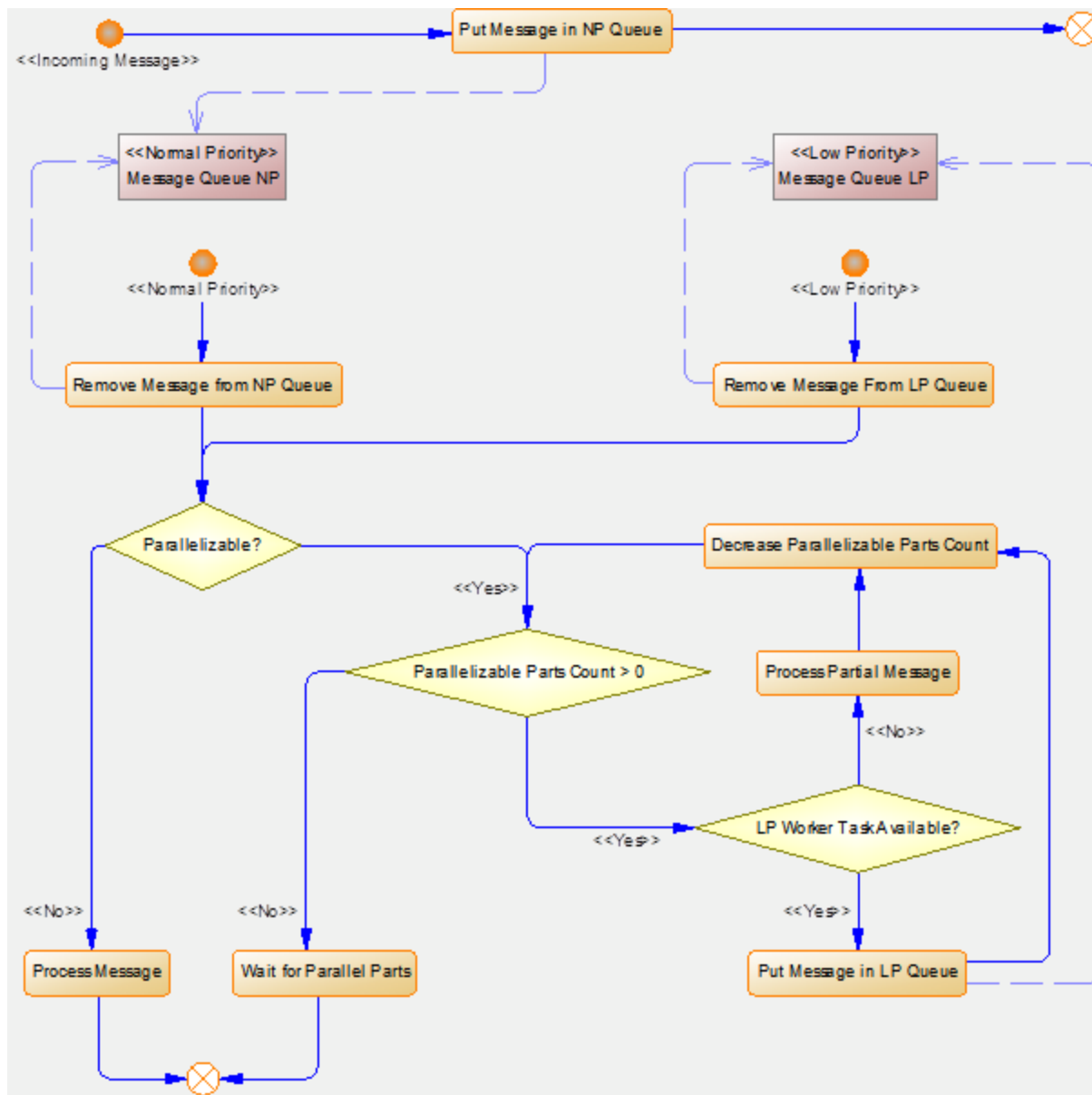
Paralelizáciu na výpočtovej úrovni zabezpečujú spracovateľské (výpočtové) vlákna (Worker Task), ktorých počet je daný konfiguráciou EDA Servera. EDA Server obsahuje tri typy spracovateľských vlákien a to:

1. vlákna určené na spracovanie primárnych požiadaviek klientov (normálna priorita),
2. vlákna na spracovanie paralelizovaných požiadaviek klientov (nižšia priorita),
3. vlákno na spracovanie interných požiadaviek EDA Servera (normálna priorita).

Každý typ spracovateľského vlákna má pridelenú vlastnú frontu správ, z ktorej správy vyberá. Rozklad paralelizovaných požiadaviek je implementovaný tak, aby nedochádzalo k zbytočnému akaniu na spracovanie vo fronte.

Spracovanie každej požiadavky od klienta ale vyžaduje jedno spracovateľské vlákno, a teda akonáhle je naraz spracovávaných toľko požiadaviek ako je počet spracovateľských vlákien, ostatné požiadavky klientov čakajú vo fronte. V prípade, že veľkosť fronty dlhodobo rastie, znamená to, že systém nestíha spracovávať požiadavky klientov. Toto môže byť spôsobené viacerými faktormi, ako napríklad vysokým zaťažením servera, kde beží EDA Server, inými procesmi, nedostatkom potom spracovateľských vlákien, pomalou sieťou medzi EDA Serverom a databázovým systémom alebo vysokým zaťažením databázového systému.

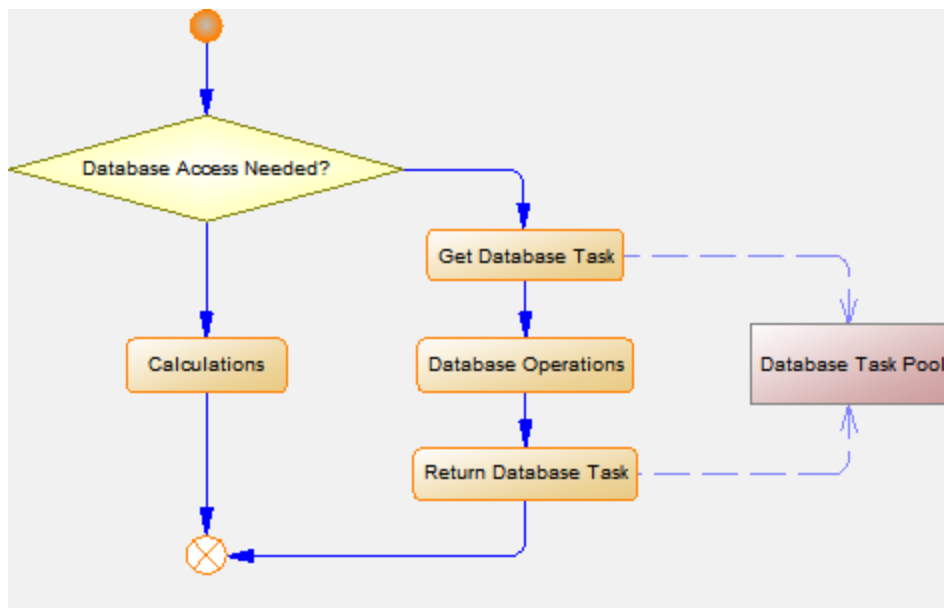
Základnú logiku spracovania požiadaviek klientov znázorňuje nasledovný diagram.



Monitorovanie výkonu EDA Servera, dŕky front, poet spracovaných správ a as ich spracovania, ako aj rozklad tohto asu na as strávený v databáze a rôzne druhy výpotov, je možné cez systémovú štruktúrovanú premennú `SV_System_EDAServerPerformance2`. Na zistenie aktuálneho stavu spracovateľských vlákien, ako sú práve spracovávané požiadavky a ich pôvod, slúži TELL príkaz `SHOW_WORKER_INFO`. Štatistiky rozdelené podľa typu požiadavky sú dostupné TELL príkazom `SHOW_WORKER_ACTIONS_STATS`.

## Databázové vlákna

Paralelizácia na úrovni databázového prístupu je realizovaná pomocou viacerých spojení s databázovým systémom, pričom každému spojeniu prislúcha jedno databázové vlákno. Databázové vlákno sa používa len vtedy, keď je potrebné spojenie s databázou a získava sa z poolu databázových vlákien. Ak EDA Server vykonáva paralelne veľa databázových operácií, môže sa zvýšiť priemerná doba čakania na databázové vlákno. Vysokú dobu čakania na databázové vlákno je možné na strane EDA Servera riešiť zvýšením počtu databázových vlákien.



Aktívne databázové vlákno je vždy pridelené spracovateľskému vláknu. Informácia o tom, či sa používa databázové vlákno je súčasťou informácií o spracovateľských vláknoch. Štatistiky rozdelené podľa typu databázovej akcie sú dostupné TELL príkazom [SHOW\\_DATABASE\\_ACTIONS\\_STATS](#).

## Cache

Na zrýchlenie výpotov a odahlenie databázy EDA využíva rôzne formy cachovania dát.

### Globálna cache

EDA Server voliteľne používa globálnu cache, v ktorej sú automaticky cachované dáta načítaných zdrojových (nie vypoítaných) vektorov. Výnimku tvoria dáta párových vektorov, ktoré cachované nie sú.

Zo štatistik využitia globálnej cache je možné zistiť, či nedochádza k astému vyhadzovaniu dát z cache, čo môže byť dôsledkom astých zápisov dát alebo maximálneho naplnenia globálnej cache. V prípade, že je vekos globálnej cache vzhľadom k množstvu najastejšie používaných dát nízka, a teda dochádza k astému vyhadzovaniu dát z cache, aby mohli byť iné dáta zapísané, môže jej použitie v krajnom prípade spôsobovať aj mierne zníženie výkonu. V tomto prípade by sa mala vekos globálnej cache zväšovať.

Štatistiky využitia globálnej cache sú súčasťou systémovej štruktúrovanej premennej **SV.\_System\_EDAServerPerformance2**. Podrobnejšie štatistiky je možné získať TELL príkazom [SHOW\\_GLOBAL\\_CACHE\\_STATS](#). Manuálne vypnutie a zapnutie, resp. vyprázdnenie globálnej cache je realizované TELL príkazmi [ENABLE\\_GLOBAL\\_CACHE](#), prípadne [INVALIDATE\\_GLOBAL\\_CACHE](#).

### Klientska cache

Klientska cache je vytváraná a rušená aplikáciou, no pre jej maximálne využitie platia rovnaké pravidlá ako pre globálnu cache. V klientskej cache sú okrem dát zdrojových vektorov uchovávané aj predpisy a výsledné dáta vypoítaných vektorov. Štatistiky využitia klientskej cache je ideálne vypisovať pred jej zatvorením. Z nich je možné zistiť, či bola vhodne zvolená jej veľkosť a ako bola využitá. Na výpis štatistik klientskej cache slúži funkcia [EDA\\_WriteCacheInfo](#).

V EDA Serveri je maximálna veľkosť klientskych cache všetkých pripojených klientov daná konfiguračným parametrom a v tejto veľkosti je kvôli zvýšeniu výkonu predalokovaná. Toto maximum je opäť potrebné zvoliť tak, aby pokrylo požiadavky všetkých súčasne pripojených klientov, inak klient nedostane požadované množstvo cache.

Monolitická knižnica EDA a EDA Workbook toto obmedzenie nemajú, pretože nepoužívajú predalokovanú klientsku cache a pamäť postupne alokujú. Sumárne štatistiky využitia klientskych cache sú taktiež súčasťou systémovej štruktúrovanej premennej **SV.\_System\_EDAServerPerformance2**.

## Archívne vektory

EDA umožňuje transparentné mapovanie archivovaných dát na zdrojové vektory a taktiež načítanie archivovaných dát cez EDA-L. Ak sa používa veľké množstvo takýchto vektorov, tak sa záťaž vo vašej miere presúva na proces **Archív** a EDA len aká na **Archív**. Zo štatistik potu a trvaní transakcií s archívom (SV.\_System\_EDAServerPerformance2) je možné takúto situáciu odhaliť.

## Spätný kanál na EDA Server

Spätný kanál na EDA Server slúži na korekciu [globálnej cache](#) po odhalení modifikácie dát EDA vykonanej mimo EDA Servera. V takom prípade databázový server nadviaže spojenie s EDA Serverom, ktorému oznámi, o bolo modifikované. Ten túto informáciu posunie D2000 Serveru, ktorý ju rozošle ostatným EDA Serverom v rovnakej skupine. Každý EDA Server v skupine na takúto informáciu reaguje vyhodnotením modifikovaných dát z globálnej cache, označením dát ako necachovateľné a potvrdením o spracovaní tejto informácie. Primárne kontaktovaný EDA Server potom taktiež odošle do databázového servera potvrdenie o spracovaní informácie.

Celý tento mechanizmus prebieha synchronne, aby zabránil vzniku nekonzistencie medzi globálnou cache a dátami v databáze. Po skončení databázovej transakcie, v rámci ktorej boli takto dáta modifikované, každý EDA Server opätovne povolí cachovanie zakázaných dát.

V prípade veľkého množstva takýchto modifikácií sa, v relatívne veľkej miere, spomaľuje práca s databázou, znehodnocuje prínos globálnej cache a celkovo znižuje výkon EDA.

Na identifikáciu takýchto stavov slúžia štatistiky o pote pripojených spätných kanálov a o množstve modifikovaných dát vektorov (SV.

\_System\_EDAServerPerformance2 a TELL príkaz [SHOW\\_TRIGGER\\_INFO](#)). V prípade dlho otvorenej databázovej transakcie modifikujúcej dáta EDA je možné na EDA Serveri TELL príkazom [SHOW\\_BYPASSED\\_ENTITIES](#) zistiť zoznam entít, ktorých cachovanie bolo z dôvodu ich modifikácie zakázané.



#### Súvisiace stránky:

[EDA server](#)