[DOC] Kolejki Mp

Kolejki MP

1. Opis metryk **MP** - Utrzymanie/MPMETRICS.
2. Procedury zwizane z **MPCore** - Utrzymanie/MpCoreService

accountingQueue

Nazwa	Priorytet	Lokalizacja
accountingQueue	krytyczny	JMS (!MySQL@india)

Opis

Kolejka do której wrzucane s wszystkie wiadomoci wysyane przez usugi. Jest to bufor kolejkujcy komunikaty przed operacj ich zapisu do schematu MP na SIGMIE.

Reakcja (Procedura restartu kolejek ACCOUTING)

1. wchodzimy na beana https://thunder:8443/jmx-console/HtmlAdaptor?action=inspectMBean&name=jboss.mq.destination%3Aservice%3DQueue%2Cname%3DaccountingQueue
2. sprawdzamy jak warto ma InProcessMessageCount i weryfikujemy czy cos tam sie zmienia i nie ma wartoci 0(parokrotnie odwieajc)
3. jesli sie nic nie zmienia to logujemy sie na accounting na mpdevelopa
4. przechodzimy do /usr/local/mpservices/system/avantis-accounting-node* (*zastepujemy numerem wezla)
5. otwieramy do edycji plik: vi classes/spring-config-bitronix.xml
6. szukamy linijki: property name="serverId" value="200096"
7. i teraz bedzie magic: dla n1 ZMNIEJSZAMY o 1, dla n2 ZWIKSZAMY o 1 !!
8. a pozniej normalnie restartujemy accounting jak kad inn mpusluge: ./avantis-accounting-node2.sh stop;./avantis-accounting-node2.sh start; tail -f /usr/local/mplogs2/avantis-accounting-node2.log
9. jesli widzimy w logach e usluga wstala przechodzimy do beana z poczatku instrukcji i sprawdzamy czy zaczelo przetwarza jeli tak to analogicznie postepujemy z drugim nodem

10. jeli to nie pomaga lub accounting si co chwila wywala to trzeba sie uda do Wojtka

zostawiam stary opis, tak eby byo w razie co ;)

1. Sprawdzi logi MP /usr/local/mplogs/mp.log
2. Sprawdzi dostpno bazy SIGMA
3. Sprawdzi dostpno bazy JMS (india)
4. Sprawdzi czas accountowania wiadomoci na podstawie logów:

```
$ tail -f /usr/local/mplogs/mp.log |grep 'ACCOUNT_COST ='
```

5. Jeeli nie ma problemów z baz (bd byy przejciowe), a MP ma problemy z poczeniem, mona spróbowa zrestartowa pule:
   - opis w https://wiki.avantis.pl/bin/view/Utrzymanie/MpCoreService#pula
6. Jeeli nie ma problemów z baz (bd byy przejciowe), mona spróbowa zrestartowa accounting.
   - Opis w MpCoreService.

Checki ACC_AND_TIME_N1 oraz ACC_AND_TIME_N2

Powysze checki monitoruj ilo wiadomoci przetwarzanych przez accounting w cigu ostatniej minuty, a take czas ich przetwarzania. Checki sprawdzaj logi /usr/local/logs/mpservices/system/avantis-accounting-node1.log oraz /usr/local/logs/mpservices/system/avantis-accounting-node2.log na accounting.

Progi checków:

Warning	Critical	
Czas:	>300 ms	>500 ms
Ilo:	<100 (noc), <600 (dzie)	<50 (noc), <200 (dzie)

noc - 24:00-6:00

dzie - 6:00-24:00

monitoringQueue

Nazwa	Priorytet	Lokalizacja
!monitoringQueue	krytyczny	JMS (!MySQL@india)

Opis

Kolejka w której s kolejkowane wszystkie alerty generowane przez usugi i platform. Nastpnie s one:

1. Zapisywane do bazy - MP.MPEVENT
2. Przesyane do ARH+

Reakcja

1. Sprawdzi dziaanie ARH+
2. Sprawdzi dostpno bazy JMS (india)
3. Sprawdzi dziaanie MPEvent-Notifier
4. Wej na JMX mp.core.MonitorManagerService i wykona metod **listChainElements()**. Jeeli bdzie wicej ni 3 wpisy - wywali nadmiarowe (powinien by jeden **PersistenceEventHandler** i dwa **MPEventNotifierHandler**).
5. Czyszczenie kolejki:
   - wchodzimy na JMX org.jboss.mq.server.jmx.Queue, wykonujemy metod **removeAllMessages()**. **Uwaga! MP w czasie usuwania wiadomoci (kilka minut dla 50 000 eventów) jest praktycznie zablokowane!**

undeliveredQueue

Nazwa	Priorytet	Lokalizacja
!undeliveredQueue	pilny	JMS (!MySQL@india)

Opis

Wiadomoci opónione (!DelayedDelivery), które czekaj na zapisanie do bazy mp_undelivered (czyli do **waiting to send**).

Reakcja

1. Sprawdzi logi MP /usr/local/logs/mpcore.server.log
2. Sprawdzi dostpno bazy SIGMA.
3. Sprawdzi dostpno bazy JMS (india).

businessQueue

Nazwa	Priorytet	Lokalizacja
!businessQueue	pilny	JMS (!MySQL@india)

Opis

W tej kolejce kolejowane s wszystkie rekordy biznesowe tzn. **RBRy** (**Raw Business Record**)
- s one wynikiem zaistniaej transakcji biznesowej na platformie - jak dokonanie patnoci -> odebranie SMS'a Premium przez usuge. **RBRy** wrzuca ne s do kolejki przez wszystkie usugi biznesowe a odbierane przez aplikacj **rbss-rating**.

Reakcja

1. Weryfikacja dziaania usugi **rbss-rating**.
2. Restart usugi **rbss-rating**.

agreementCollectorQueue

Nazwa	Priorytet	Lokalizacja
!agreementCollectorQueue	wysoki	JMS (!MySQL@india)

Opis

Wiadomoci czekajce na przetworzenie przez usug **agreementcollector** lub **n1.mcsm**.

Reakcja

1. Sprawdzi dziaanie usugi **agreementCollector** na **tango**.
2. Sprawdzi dziaanie usugi **rbr-trigger-service** na **whiskey**.
3. Sprawdzi dziaanie usugi **n1.mcsm** na **whiskey**.
4. Sprawdzi dostpno bazy JMS (india).

> *UWAGA: **agreementCollector** jest obecnie na sierra, **rbr-trigger-service** na lima, tylko **n1.mcsm** jest tak jak napisane powyej na whiskey*
>
> *Jeeli powysze nie pomoe warto spróbowa zrestartowa **rbss-rmi-service** na tango*
>
> *Daj w komentarzu bo dokumentacja i tak jest do weryfikacji. MJ*

Jeeli kolejka przekroczy 150 tysicy i zacznie wpywa to negatywnie na dziaanie innych usug, mona tymczasowo wyczy przetwarzanie RBRów w **n 1.mcsm**. Wykonuje si to zapytaniem na bazie, schemat mcsm (pass tutaj).

Wczenie przetwarzania (domylnie):

```
pdate COMPONENT_VALUES set value='McsmRBRRequestDTO' where component_id in (
select c.component_id from filters f, COMPONENTS c, COMPONENT_VALUES cv
where f.filter_id=101 and c.filter_id=f.filter_id and cv.component_id=c.component_id
and c.component_type_id=5);
```

Wyczenie przetwarzania (awaryjnie):

```
update COMPONENT_VALUES set value='disabled_McsmRBRRequestDTO' where component_id in (
select c.component_id from filters f, COMPONENTS c, COMPONENT_VALUES cv
where f.filter_id=101 and c.filter_id=f.filter_id and cv.component_id=c.component_id
and c.component_type_id=5);
```

Trzeba pamita, e nie mona zostawi wyczonego przetwarzania na dugi okres (kilka godzin). Dodatkowo - kad tak rekonfiguracj najlepiej skonsultowa z DEV (Pawe Osiski).

Waiting to send

Nazwa	Priorytet	Lokalizacja
Waiting to send	normalny	Oracle (mp_undelivered)

Opis

Wiadomoci opónione (**DelayedDelivery**) przez **MP**. Wiadomoci trzymane s w schemacie **mp_undelivered**, w tabelce **delayed_delivery**.

Reakcja

1. Weryfikacja alertu.
2. Sprawdzenie czy która z usug nie zaczea tam wrzuca mieci (select na **delayed_delivery**).

Ready to send

Nazwa	Priorytet	Lokalizacja
Ready to send	pilny	Oracle (mp_undelivered)

Opis

Kolejka zawiera wiadomoci z **delayed_delivery**, które powinny zosta wysane przez **MP** (z **DELAYED_DELIVERY_TIME** > **current_timestamp**). W MP dziaa proces **DelayedDelivery**, który cyklicznie wybiera te wiadomoci i wrzuca do MP.

Przez kolejk opónionych przechodz wszystkie opóniane WapPushe - czyli te z platformy contentowej (*smsps/comproxy) i bulków contentowych.*

Reakcja

1. Weryfikacja czy nie ma problemów z **sigm** (schemat **mp_undelivered**).
2. Sprawdzenie czasów wybierania wiadomoci:

```
$ tail -f /usr/local/logs/mpcore/server.log |grep DelayedDeliveryService
```

3. Jeeli wiadomoci przetwarzane s prawidowo i nie wida adnych bdów - mona spróbowa zatrzyma bulki dwu-elementowe (z opónionymi wappushami). Bulki takie namierzamy przy pomocy bulki.php, a zatrzymujemy za pomoc pbm-ui.
4. Wiadomoci, które oczekuj do wysania moemy wybra zapytaniem:
    - ```
      SELECT  dd.sender_service_id,  dd.recipient_service_id, count(*) FROM mp_undelivered.delayed_delivery dd WHERE dd.delayed_delivery_time < sysdate group by dd.sender_service_id, dd.recipient_service_id
      ```
    - naley zwróci uwag przede wszystkim na kolumny **RECIPIENT_SERVICE_ID** i **SENDER_SERVICE_ID**. Jeli np. wszystkie rekordy maj tylko jedno **id** w **recipient_service_id** to znaczy e najprawdopodobniej jest problem z usug. Jeli za s tam róne wartoci i ich nie ubywa - prawdopodobnie jest problem z samym mechanizmem wybierania tych wiadomoci przez **MP**.
5. Ilo jednoczenie wybierany wiadomoci mona zmieni w JMX mp.core.DelayedDeliveryService. Rozsdne wartoci oscyluj midzy **500** a **3000**.
6. Jeeli nie ma problemów z baz (bd byy przejciowe), a MP ma problemy z poczeniem, mona spróbowa zrestartowa pule: opis w MpCoreService#pula.

BUSINESS DELAYED DELIVERY WAITING TO SEND oraz BUSINESS DELAYED DELIVERY READY TO SEND

Nazwa	Priorytet	Lokalizacja
BUSINESS DELAYED DELIVERY WAITING TO SEND	normalny	Oracle (mp.delayed_business_object)
BUSINESS DELAYED DELIVERY READY TO SEND	normalny	Oracle (mp.delayed_business_object)

Opis

1. To w zasadzie nie jest typowa kolejka, tylko tabelka **mp.delayed_business_object**. Trafiaj do niej rekordy **RBR** nie przetworzone (odrzucone) przez **RBSS** (**rbss-rating** na **tango**) albo przez **Agreement Collector**.
2. Tabelka **mp.delayed_business_object** zawiera kolumn **subscriber_id** która definiuje z którego mechanizmu trafiy do niej RBRy:
    - 0 dla **rbss-rating**
    - 1 dla **Agreement Collector**
3. Tabelka zawiera te pole z dat która definiuje kiedy RBR ma znowu zosta przetworzony.
4. Kolejka **READY TO SEND** to liczba rekordów w tej tabelce które maj ustawiony ten czas na teraz albo przeszo.

Reakcja

1. Poinformowanie Józka Andrzejewskiego.

Queue with no service

Opis

Monitoring weryfikuje czy wystpuje kolejka na usugach ju wyczonych/nigdy nie istniejcych.

Reakcja

Naley sprawdzi dlaczego wiadomoci pojawiy si na kolejce ju nie istniejcej usugi oraz doprowadzi do sytuacji aby nie trafiay tam dalej. Kolejnym krokiem jest wydobycie jakie wiadomoci trafiy na kolejk (Logi MP po frazie "dispatched to queue: QUEUE.IDKOLEJKI"), nastpnie wydoby je na podstawie TransationID aby po zrzuceniu z nieistniejcej kolejki dokona ponownego przetworzenia.
      ```

Kolejki MP Secondary

W zasadzie to s takie same kolejki jak dla gównego MP, tylko s na platformie backupowej i maj troch inne progi.

Nazwa	Pilno	Próg	Reakcja
Secondary accountingQueue	rednia	100	Weryfikacja dziania usugi mp-config-pump, polaczenia interconnect
Secondary agreementCollectorQueue	rednia	100	Weryfikacja dziania usugi mp-config-pump, polaczenia interconnect
Secondary businessQueue	rednia	100	Weryfikacja dziania usugi mp-config-pump, polaczenia interconnect
Secondary dispatchQueue	rednia	100	Weryfikacja dziania usugi mp-config-pump, polaczenia interconnect
Secondary monitoringQueue	rednia	100	Weryfikacja dziania usugi mp-config-pump, polaczenia interconnect
Secondary undeliveredQueue	rednia	100	Weryfikacja dziania usugi mp-config-pump, polaczenia interconnect
accountingDLQ	rednia	100000	Eskalacja do developera MP.

– Main.MikolajKlimek - 2011-02-04