

Author – A.Kishore/Sachin
<http://appsdba.info>

Important Queries to Troubleshoot concurrent request / manager issues

➤ **Query to check the scheduled concurrent Requests/Programs:**

```
SELECT cr.request_id,  
       DECODE (cp.user_concurrent_program_name,  
              'Report Set', 'Report Set:' || cr.description,  
              cp.user_concurrent_program_name  
              ) NAME, argument_text, cr.resubmit_interval,  
       NVL2 (cr.resubmit_interval, 'PERIODICALLY',  
            NVL2 (cr.release_class_id, 'ON SPECIFIC DAYS', 'ONCE')  
            ) schedule_type, DECODE (NVL2 (cr.resubmit_interval,  
            'PERIODICALLY', NVL2 (cr.release_class_id, 'ON SPECIFIC DAYS', 'ONCE')  
            ), 'PERIODICALLY', 'EVERY '  
            || cr.resubmit_interval  
            || ''  
            || cr.resubmit_interval_unit_code  
            || ' FROM '  
            || cr.resubmit_interval_type_code  
            || ' OF PREV RUN',  
            'ONCE', 'AT :'  
            || TO_CHAR (cr.requested_start_date, 'DD-MON-RR HH24:MI'),  
            'EVERY: ' || fcr.class_info  
            ) schedule, fu.user_name, requested_start_date  
FROM apps.fnd_concurrent_programs_tl cp,  
     apps.fnd_concurrent_requests cr,  
     apps.fnd_user fu, apps.fnd_conc_release_classes fcr  
WHERE cp.application_id = cr.program_application_id  
     AND cp.concurrent_program_id = cr.concurrent_program_id  
     AND cr.requested_by = fu.user_id  
     AND cr.phase_code = 'P'  
     AND cr.requested_start_date > SYSDATE  
     AND cp.LANGUAGE = 'US'  
     AND fcr.release_class_id(+) = cr.release_class_id  
     AND fcr.application_id(+) = cr.release_class_app_id;
```

Author – A.Kishore/Sachin
<http://appsdba.info>

- **Query to check the duplicated schedules of the same program with the same arguments**

```
SELECT request_id, NAME, argument_text, user_name
FROM (SELECT cr.request_id,
DECODE (cp.user_concurrent_program_name,
'Report Set', 'Report Set:' || cr.description,
cp.user_concurrent_program_name
) NAME, argument_text, fu.user_name
FROM apps.fnd_concurrent_programs_tl cp,
apps.fnd_concurrent_requests cr,
apps.fnd_user fu
WHERE cp.application_id = cr.program_application_id
AND cp.concurrent_program_id = cr.concurrent_program_id
AND cr.requested_by = fu.user_id
AND cr.phase_code = 'P'
AND cr.requested_start_date > SYSDATE
AND cp.LANGUAGE = 'US'
AND fu.user_name NOT LIKE 'PPG%') t1
WHERE EXISTS ( SELECT 1 FROM (SELECT cr.request_id,
DECODE (cp.user_concurrent_program_name, 'Report Set', 'Report Set:'
|| cr.description, cp.user_concurrent_program_name ) NAME, argument_text,
fu.user_name FROM apps.fnd_concurrent_programs_tl cp,
apps.fnd_concurrent_requests cr,
apps.fnd_user fu WHERE cp.application_id = cr.program_application_id
AND cp.concurrent_program_id = cr.concurrent_program_id
AND cr.requested_by = fu.user_id AND cr.phase_code = 'P'
AND cr.requested_start_date > SYSDATE
AND cp.LANGUAGE = 'US' AND fu.user_name NOT LIKE 'PPG%') t2
WHERE t1.NAME = t2.NAME AND t1.argument_text = t2.argument_text
AND t1.user_name = t2.user_name
GROUP BY NAME, argument_text, user_name
HAVING COUNT (*) > 1)
ORDER BY user_name, NAME
```

Author – A.Kishore/Sachin
<http://appsdba.info>

➤ **Query to check average pending time per request:**

Please Note: Average pending time for a request is calculated like below:

("Highest of Requested_start_date or Date_submitted" - Actual_start_date) /
Total requests

A Request can be in **Pending** state for variety of reasons like conflict with other requests, improperly tuned managers (sleep seconds / cache size / number of managers etc)

We can schedule this script to gather data regularly for historical analysis as we normally purge the concurrent requests regularly.

```
SELECT TO_CHAR (actual_start_date, 'DD-MON-YYYY') DAY,  
concurrent_queue_name,  
(SUM ( ( actual_start_date  
- (CASE  
WHEN requested_start_date > request_date  
THEN requested_start_date  
ELSE request_date  
END  
)  
)  
)  
* 24  
* 60  
* 60  
)  
)  
/ COUNT (*) "Wait_Time_per_Req_in_Secs"  
FROM apps.fnd_concurrent_requests cr,  
apps.fnd_concurrent_processes fcp,  
apps.fnd_concurrent_queues fcq  
WHERE cr.phase_code = 'C'  
AND cr.actual_start_date IS NOT NULL  
AND cr.requested_start_date IS NOT NULL  
AND cr.controlling_manager = fcp.concurrent_process_id  
AND fcp.queue_application_id = fcq.application_id  
AND fcp.concurrent_queue_id = fcq.concurrent_queue_id  
GROUP BY TO_CHAR (actual_start_date, 'DD-MON-YYYY'),  
concurrent_queue_name  
ORDER BY 2
```

Author – A.Kishore/Sachin
<http://appsdba.info>

➤ **Query to checking which manager is going to execute a program:**

```
SELECT user_concurrent_program_name, user_concurrent_queue_name
FROM apps.fnd_concurrent_programs_tl cp,
apps.fnd_concurrent_queue_content cqc,
apps.fnd_concurrent_queues_tl cq
WHERE cqc.type_application_id(+) = cp.application_id
AND cqc.type_id(+) = cp.concurrent_program_id
AND cqc.type_code(+) = 'P'
AND cqc.include_flag(+) = 'I'
AND cp.LANGUAGE = 'US'
AND cp.user_concurrent_program_name =
'&USER_CONCURRENT_PROGRAM_NAME' AND NVL (cqc.concurrent_queue_id, 0)
= cq.concurrent_queue_id
AND NVL (cqc.queue_application_id, 0) = cq.application_id
AND cq.LANGUAGE = 'US'
```

➤ **Query to see all the pending / Running requests per each manager wise :**

```
SELECT request_id, phase_code, status_code, user_name,
user_concurrent_queue_name
FROM apps.fnd_concurrent_worker_requests cwr,
apps.fnd_concurrent_queues_tl cq,
apps.fnd_user fu
WHERE (cwr.phase_code = 'P' OR cwr.phase_code = 'R')
AND cwr.hold_flag != 'Y'
AND cwr.requested_start_date <= SYSDATE
AND cwr.concurrent_queue_id = cq.concurrent_queue_id
AND cwr.queue_application_id = cq.application_id
AND cq.LANGUAGE = 'US'
AND cwr.requested_by = fu.user_id
ORDER BY 5
```

Author – A.Kishore/Sachin
<http://appsdba.info>

➤ **Query to checking the incompatibilities between the programs:**

```
SELECT a2.application_name, a1.user_concurrent_program_name,
       DECODE (running_type,
              'P', 'Program', 'S', 'Request set', 'UNKNOWN') "Type",
       b2.application_name "Incompatible App",
       b1.user_concurrent_program_name "Incompatible_Prog",
       DECODE (to_run_type, 'P', 'Program', 'S', 'Request set', 'UNKNOWN')
incompatible_type FROM apps.fnd_concurrent_program_serial cps,
apps.fnd_concurrent_programs_tl a1, apps.fnd_concurrent_programs_tl b1,
apps.fnd_application_tl a2, apps.fnd_application_tl b2
WHERE a1.application_id = cps.running_application_id
AND a1.concurrent_program_id = cps.running_concurrent_program_id
AND a2.application_id = cps.running_application_id
AND b1.application_id = cps.to_run_application_id
AND b1.concurrent_program_id = cps.to_run_concurrent_program_id
AND b2.application_id = cps.to_run_application_id
AND a1.language = 'US' AND a2.language = 'US'
AND b1.language = 'US' AND b2.language = 'US'
```