ERROR: invalid page in block XXXXX of relation base

Last updated by | Lisa Liu | Nov 6, 2020 at 10:34 AM PST

Alert Name: [Region] Server is having invalid page in the block error

Things to check

1. Query Kusto for invalid block logs in the region specified in the alert.

MonRdmsPgSqlSandbox | where text contains "invalid page in block" | summarize count() by LogicalServerName

2. Find the time when most of the errors occurred.

MonRdmsPgSqlSandbox | where text contains "invalid page in block" | project LogicalServerName, TIMESTAMP, text

3. Look up sandbox logs for 30 minutes before and 30 minutes after one of the logs above

4. If you are still seeing the error in the last few days, go to the next step, else it may be a temp table and issue might have been resolved.

5. File an IcM using ASC

Things should be included in the ICM:

1. Add all the information you collected above about the corrupted objects and the event files to the ICM.

2. Ask the customer to Try VACUUM FULL <tablename>, it might fix the corruption

How to fix

- Since the object is corrupted, we need to repair it. We have to work with CSS/Customer to fix the issue. if the object is index, we can drop and recreate the index. For a table we might lose the data, if not, restore a PITR and recover the data from that single table and copy it over to the master.

- If the customer gave the consent to repair the data(which involves some data loss), please File an IcM

Customer Communication

If the call is made to communicate the mitigation steps to the customer and let them handle the recovery of corrupted table you can use the following message template you need the IcM to get the variables in bold below:

We hope that this message finds you well.

Our telemetry has detected logs indicating invalid pages in relation **<RELATION OID>** in database **<DB OID>**.

The details on the objects in question can be obtained by running:

select datname from pg_database where oid = **<DB OID>**;

select * from pg_class where relfilenode=**<RELATION OID>** ;

Based on the type of the relation in question we recommend to take the following steps to mitigate the issue:

a. If the relation is an index, we recommend drop and recreate it.

b. If it is a table, we recommend creating a PITR server(s) and using it to recover the data.`

1. Can create an empty table with a matching schema and INSERT to it rows from the old table using

SELECT * INTO TABLE <new table name> from <old table name> LIMIT X OFFSET Y ORDER BY <unique key>

where X and Y are numbers reflecting row counts to skip and insert. These numbers are to be chosen so that the problematic parts are not included.

2. To recover the remaining parts use their last good version(s) from the PITR server(s).

The table is in a good state if you can successfully execute SELECT * on it. However, even if the table

as a whole is not in a healthy state you might still be able to recover some parts of it using the above

approach with OFFSET and LIMIT.

During recovery please remember that the ordering of tuples is not guaranteed to be consistent.

Considering that the time between when the issue was actually introduced and when it was discovered

is unknown we recommend to start recovery attempts with the PITR copy right before the issue was

detected (<SPECIFY DETECTION TIME>) and moving back in time if the issue is present in the currently

attempted PITR copy. To speed up the process one can perform the search for the last known good

version in binary* fashion with detection time being the start of the search area and the oldest available

restore point being the end.

3. Once the new table is ready drop the old version and rename the newly constructed one accordingly.

We are sorry for any inconvenience it might have caused and are continuously working on understanding and fixing

any possible root causes for any similar occurrences.

*In case of high write activity on the server PITR operations might be time consuming. One can perform more PITR

operations in parallel to speed up the process resulting in ternary / quaternary / ... search.

Email Template - Permission requested for troubleshooting your Azure Database for PostgreSQL instance

Hi [Customer name],

The telemetry generated on one of your Azure Database for PostgreSQL instances below has started alerting us beginning **YYYY-MM-dd HH:mm:ss** UTC. In order to get further understanding of the issue you are experiencing and remedy, we need your permission to access the server node. Could you please let us know whether or not we can diagnose the issue on the node?

Subscription Id: **AAAAAAAA-BBBB-CCCC-DDDD-EEEEEEEEEEEE**

Instance Name: **XXXXXX**

Thank you so much for your help!

Your Signature

How good have you found this content?