BangDB – *Configuration*

www.iqlect.com

Copyright © 2014 IQLECT, All rights reserved

Configuration -Bangdb Embedded

Bangdb provides range of configurable parameters and knobs to adjust them to suit the application in best possible manner. Following are few important ones listed here;

1. Server Dir(SERVER_DIR): This to let db know where do you want db to keep db related files. Default is the app directory

2. Database Name(BANGDB_DATABASE_NAME): Applicable only for server as for embedded version, user opens the db with the approapriate db name.

3. *persist type: (`BANGDB_PERSIST_TYPE`) user can set the following db types using this switch;*

I. INMEM_ONLY – This is for using db as Embedded in proc mode. The data will always be written and read from the cache. The db will never go to disk for any operation. The Application will allocate enough buffer space to be able to work out of the cache comfortably. The db can be accessed concurrently by multiple threads, but only one process can edit the db at a time. There is no overflow to disk supported in this case. While closing the application may request db to flush all the data to disk for later use as required or application can simply discard the data

II. INMEM_PERSIST – This is similar to the above type except that the data will overflow to disk when buffer pool is full. The data in the cache is backed by the file system hence regular flushing, fetching etc will be managed by the DB Page Cache

III. PERSIST_ONLY – This mode if for writing directly to file on the disk. In this mode there is no buffer pool and data is read and written to/from disk. This mode is solely for application that wants maximum durability and is not concerned about performance. In this mode multiple processes can work on the same db simultaneously but they can't be multi-threaded

4. index type: (`BANGDB_INDEX_TYPE`) The supported index types at the moment are;

I. EXTHASH – This is for extend-able hash based index support. This doesn't maintain the order of the keys in the index files but uses hash value of the key to read and write the data. The constant look up time for keys in this case provides the maximum performance for the read operation. Please see the perf doc for stats

II. BTREE – This is for B+ link Tree based index support. This maintains the order of the keys while write and hence the range scan is useful in this case. This performs equally good in both read and write. Please see the perf doc for stats

5. buffer pool size(BUFF_POOL_SIZE_HINT): This sets the buffer pool size. This is again hint to the db and bangdb then computes the best possible value for buffer pool less than but as close to this hint as possible. Providing higher value is always good but it depends on the machine configuration as well. Using BUFF_POOL_SIZE_HINT once can set the value

6. log (BANGDB_LOG): Bangdb implements write ahead log for data recovery and durability. But user has an option to enable or disable the log depending upon need. For example for INMEM_ONLY db type user may want to disable log as application may not be interested in the persisting the data on the disk. Disabling log further improves the performance significantly.

7. log buffer size(LOG_BUF_SIZE): When log is enabled, this size will be used for circular log buffer. The default 256MB is optimal for many scenarios but can be changed as per need. Depending upon the RAM size, the log buffer size can be increased or decreased. Using LOG_BUF_SIZE, one can set the value

Copyright © 2014 IQLECT, All rights reserved

8. Application log(BANGDB_APP_LOG): user can use the bangdb logging mechanism for app logging need. The BangDB uses this log for internal error log as well. Using this config variable, user can switch to BangDB app logging mechanism or to syslog(for linux) or to standard input output

9. App log buffer size(DB_APP_LOG_SIZE_MB): For app logging for BangDB is asynchronous for performance reasons and hence it maintains in memory buffer for the same. Using this param user can set the log size appropriately

10. data size(DAT_SIZE): This denotes the maximum size of data supported by the db. Default set value is 64 KB but the user can change it to higher or lower value as suited. Note that this value has some impact on the available buffer pool size and RAM size of the machine. If we have higher RAM on the machine, then we can commit more memory to the buffer pool hence typically can handle higher sizes of the data. For moderate 256MB buffer pool. We can live with 1-8MB of the maximum data sizes, and if we have 1GB for buffer pool we can go for 1-16MB as well. Note that there is no theoretical limit on this but for practical purposes we should not over commit this value. For data beyond the set maximum size, user will have to split and write and the get and combine accordingly

11. key size(KEY_SIZE): Default size for index is 24 bytes (the key size). This again has no theoretical limit but one should be careful in setting this. Too high value would lead to reduced performance for both read and write. The key size has to be less than the page size, typically user should provide the space for at-least 16 keys per page, but more is better. In all probabilities the key sizes would remain limited and in few bytes only hence set this according to the requirement without bloating index size

12. Max Resultset Size: This is to limit the maximum amout of data to be returned in a single scan call. For example if limited by size of data, default value is 2MB or user can set differen value through config changes or at run time through table_env type. If limited by number of rows, then this indicates number of rows to be returned in a single scan call

13. Key Comp Func Id(KEY_COMP_FUNCTION_ID): This is to select the appropriate key comparison method provided by the db. Value 0 means select the lexicographically sorting method and value 1 means quasi lexicographically sorting method

14. Autocommit(BANGDB_AUTOCOMMIT): When transaction is enabled and autocommit is off the user will have to explicity wrap all the ops in begin and commit/abort transaction boundary. This means that without explicitly having transaction, user will not be able to run even a single op. When off, user can continue using the db in usual manner

15. Bangdb Transaction Cache Size(BANGDB_TRANSACTION_CACHE_SIZE): This indicates the cache size for transaction house keeping. Transaction cache size in terms of number of concurrent transactions. Increasing this would decrease the probability of transaction getting aborted due to forced reclaim of cache nodes. But default works well in most of the situation

16. Max Tables(MAXTABLE): Will be depricated. The max num of tables user can create for a db. Default value is 64

17. Max Conn(MAXCONN): For embedded db only. Max num of connections application can create. Deafault is 256

18. Max Threads(MAX_THREADS): This is just an indication to the db in embedded mode. This defines max num of threads user can use concurrently. Leave this value at default as 256

19. page size(PAGE_SIZE_BANGDB): Default page size used by db is 8192 bytes. But user can change it to any size that he/she thinks fit. Note that having too high or too low page size has it's own pros and cons, but maintaining a balance is important for getting optimal performance. The various factors that could lead to determination of page size could be, (a) avg size of keys (b) avg size of data (c) single

Copyright © 2014 IQLECT, All rights reserved

threaded vs multi-threaded etc...

20. page split factor(PAGE_SPILT_FACTOR): This indicates in what proportion a page would be split in case of BTree. If operation is sequential then high value for the config helps in performance otherwise for random operations the 50 value is good. The range for the paramerter is between 50 – 100.

21. *master log buf size(MASTER_LOG_BUF_SIZE):* Bangdb maintains a master log for tracking purpose for the various log files and also some of the db stats when log is ON. The default size of the file is 64KB which could take care of several hundreds of millions of records

22. sync tran(BANGDB_SYNC_TRAN): The transaction writes log to the disk during commit, but this default behaviour can be changed with this flag

23. signal handler(BANGDB_SIGNAL_HANDLER_STATE): BangDB registers various signal handler to deal with any crash, illegal memory access etc... in order to ensure DB sanity. User can enable or disable the signal handling by the DB using this flag

24. log flush frequency(LOG_FLUSH_FREQ): The frequency at which the log buffer should be flushed to the disk. Note that the high value here may mean increased performance but higher loss of data in case of process/app/machine crash. The default value is 50 ms which is very good from the data recovery perspective

25. Checkpoint Enabled(CHKPNT_ENABLED): To switch on/off the checkpoint of log in the db

26. checkpoint frequncy(CHKPNT_FREQ): The frequency at which the db will do the log check-pointing. This is good from the data recovery speed perspective in case of crash or issues. The default value is 3.37 sec which is a good balance between performance and check pointing

27. log split check frequency(LOG_SPLIT_CHECK_FREQ): The frequency for worker to check if log needs to split. Default value works good for most of the time

28. buffer reclaim frequency(BUF_FLUSH_RECLAIM_FREQ): This frequency value is used by the buffer pool background workers to flush the dirty pages to the disk and also re claim the memory in case of need. This frequency means that the worker will at least check regularly on what action to take and depending upon the situation of the pool they will decide on what to do. The default value is 60ms

29. Growth Buffer Size: Not used currently. In future when enabled, db will add this much of extra memory when in presussure (until feasible)

30. Page Read Factor: This indicates speed at which data is read. This is an opaque value and doesn't correspond to number of pages or blocks

31. Page Write Factor: This indicates speed at which data is written. This is an opaque value and doesn't correspond to number of pages or blocks

32. buffer growth and shrink, incremental: In case of dire need of memory or free pages in high pressure scenario, buffer pool will use this value to augment the size of the pool. When pressure eases, buffer pool will free the added buffer size accordingly. In this version this is not enabled, it will be available in next version

33. combine pages for write: While writing the pages tot the disk, buffer pool gathers the adjacent pages to combine writes into single write to increase performance. User can define the value or leave to the db to use the OS supported value. 0 means using the system supported one. Bangdb takes various pages to write, sorts them in non decreasing order (for block number), and the writes them together in chunks to improve peroformance

Copyright © 2014 IQLECT, All rights reserved

34. minium scan for dirty pages: This is again an hint to the db on how much to scan at a time to find the dirt pages. The db doesn't follow this in total but uses this as hint to decide various other parameters

35. *minimum scan for updated pages:* This is the length of the lru list to scan to find a updated page that can be reused for other block. Default is 32 but can have any theoretical number. Note that high value of this constant would impact the performance

36. *flush constratints:* The flush constraints for index and data or dir file defines the way background flushing of dirty pages would be considered in adaptive manner. Higher value would distribute the flushing to larger operation time, but this would affect the performance and might have more IO, whereas lower value would only allow flushing when in high memory pressure, hence does not affect performance when flushing is not required but when faced with low memory situation, it makes the operations to slow down momentarily hence more checkered behaviour. The default value has been set after doing proper tuning and testing but user can change the value and check the performance

37. *reclaim constraints:* The reclaim constraints for index and data or dir file again is to control the reclaiming of pages to clear memory for future operations. Higher value means even recliming throughout the operation time but would result in more cache misses. Whereas lower value would allow higher cache hit but when under pressure, operation may take longer time in wait till memory is cleared up or may be first flushed and then cleared up. The default again is set after the tuning hence should work in most of the use case scenarios, however users are free to change and set according to their need

38. index and data page normalization factor: This value is to set the relative importance of index pages over data pages in the memory. In most cases we would like to keep index pages in memeory as much and as long as possible for quicker access. Higher the value, we give higher importance to index pages and vice versa. The deault value 2 works fine in most of the cases

Server Related Config Parameters

39. Server Type(SERVER_TYPE): Applicable only for server. This denotes the type of server, master or slave. 0 means master and 1 means slave

40. Enable Replication(ENABLE_REPLICATION): Applicable only for server. For switching on/off the replication for the server. When off no slaves can be attached to the server. And when off slaves can be attached during startup of during the course of the server run

41. Server ID(SERVER_ID): Applicable only for server (not for embedded version). The ip address of the server, the current instance for which user is editing the config file. Note that both master and slave use the same config file (each server keeps one copy of bangdb.config, and change the config values as they require). The server ID is for the server's own IP address (could be master or slave). In other words it defines the IP address of the server which is configuring the bangdb.config

42. Server Port(SERV_PORT): Applicable only for server. The port at which server listens, the current instance for which user is editing the config file

43. Master Server ID(MASTER_SERVER_ID): Applicable only for server. The ip address of the master server. Note that for master, the SERVER_ID and MASTER_SERVER_ID would be same. But for slave, SERVER_ID would be it's own ip address and MASTER_SERVER_ID would be the ip address of the master in th cluster

44. Master Server Port(MASTER_SERV_PORT): Applicable only for server. The port at which the master server listens

Copyright © 2014 IQLECT, All rights reserved

45. ListenQ(LISTENQ): Applicable only for server. The queue size for the listen API

46. Max Slaves(MAX_SLAVES): Applicable only for server. The maximum number of slaves a master can have. This is defined before running the master and only applicable for master. If the max slaves is 4, user can not attach more than 4 slaves at any time during the run of the server. Howerver can be changed after stopping and the server

47. Ops Rec Buf Size(OPS_REC_BUF_SIZE): Applicable only for server. This for non-log based replicvation. This defines the buffer size of the current ops rec when replication is on. Typically 512 MB is more than for 100s millions operations. Hence user can set it accordingly. The default value is 256 MB as it's assumed that replication would be over before this buffer gets full. As long as there is space in the buffer the master continue serving the requests (both read and write) during the replication process, once full master will deny the write ops but continue with reads until the replication is done

48. PING Freq(PING_FREQ): Applicable only for server. This is to check the health of other servers in the cluster. Default value is 10 sec

49. PING Threshold(PING_THRESHOLD): Applicable only for server. The limit after which the server will decide other node is dead or unreachable

50. Client Time Out(CLIENT_TIME_OUT): Applicable only for server. This is the expiry time for clients. This doesn't apply to the nodes in the cluster, only for connecting clients. The default value is 120 sec. 0 means no timeout

51. Max Clients Events(MAX_CLIENT_EVENTS): This specifies how many concurrent connections from clients the db server can handle. Applicable only for server

52. Server Stage Option: Applicable only for server. The server can be run in different stages. It can be run in two stage mode where one stage handles the client connections and events and other takes care of all other operations. The other mode involves four stages, one for client connections and events, one for read, one for ops, and last one for write. The default is two stage mode (1)

53. Server Ops Workers: Applicable only for server. User can specify the number of workers for the ops job. Default ins num of cores on the machine (for >= 4 cores) else ncores – 2

54. BangDB Group(BANGDB_GROUP): Default group for the bangdb users

Copyright © 2014 IQLECT, All rights reserved