New tables:

Apptype:

Merge:1

Remerge:2

Retention:3

Index verification:4

CREATE TABLE [dbo].[IndexLockRequestQueue](

[LockQueueID] [int] IDENTITY(1,1) NOT NULL,

[GroupID] [int] NOT NULL,

[AppType] [tinyint] NOT NULL,

[DateRequested] [datetime] NOT NULL,

[DateExpired] [datetime] NOT NULL,

[Step] [tinyint] NOT NULL,

[DateFinished] [datetime] NULL,

CONSTRAINT [PK\_SearchIndexGlobalLockQueue] PRIMARY KEY CLUSTERED

(

[LockQueueID] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

GO

CREATE TABLE [dbo].[IndexLockRequestQueue\_Item](

[LockQueueIDItem] [int] IDENTITY(1,1) NOT NULL,

[LockQueueID] [int] NOT NULL,

[SearchIndexID] [int] NOT NULL,

[NewSearchIndexID] [int] NULL,

[DateRequested] [datetime] NOT NULL,

[DateFinished] [datetime] NULL,

[IsComplete] [bit] NOT NULL,

CONSTRAINT [PK\_SearchIndexLockQueue] PRIMARY KEY CLUSTERED

(

[LockQueueIDItem] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

GO

ALTER TABLE [dbo].[IndexLockRequestQueue\_Item] WITH CHECK ADD CONSTRAINT [FK\_IndexLockRequestQueue\_Item\_IndexLockRequestQueue] FOREIGN KEY([LockQueueID])

REFERENCES [dbo].[IndexLockRequestQueue] ([LockQueueID])

GO

ALTER TABLE [dbo].[IndexLockRequestQueue\_Item] CHECK CONSTRAINT [FK\_IndexLockRequestQueue\_Item\_IndexLockRequestQueue]

GO

ALTER PROCEDURE [dbo].[stp\_GetLevel3IndexBestCandidateAndLock]

@TotalDocCount INT,

@MaxIndexDocCount INT,

@GroupId INT,

@Type TINYINT,

@AppType TINYINT,

@MaxShards INT=4,

@BatchGuid UNIQUEIDENTIFIER

AS

BEGIN

DECLARE @Tbl TABLE(IndexId INT)

DECLARE @Tbl\_IndexesToExclude TABLE(IndexId INT)

INSERT INTO @Tbl\_IndexesToExclude

SELECT DISTINCT ISNULL(NewSearchIndexID,SearchIndexID)

FROM IndexLockRequestQueue\_Item (NOLOCK) A INNER JOIN

IndexLockRequestQueue (NOLOCK) B ON A.LockQueueID=B.LockQueueID AND A.IsComplete=0

WHERE A.IsComplete=0 AND B.Step < 2 AND

GETDATE()<B.DateExpired

UPDATE searchindex SET IsLocked=1,BatchGuid=@BatchGuid,DateLeased=GETDATE(),AppType=@AppType

OUTPUT inserted.indexid INTO @Tbl

WHERE indexid IN

(

SELECT TOP (@MaxShards) indexid FROM searchindex (NOLOCK)

WHERE groupnum = @GroupId and TYPE=@Type and active=1 and IsLocked=0 and DocCount<=@MaxIndexDocCount-@TotalDocCount

AND IndexId NOT IN

(

SELECT IndexId FROM @Tbl\_IndexesToExclude

)

ORDER BY DocCount DESC

) AND IsLocked=0

SELECT indexid,indexpath,groupnum,[Type],LeaseSeconds,active,DocCount,IsLocked,lastUpdate,SizeInMB,BatchGUID,DateLeased,AppType

FROM searchindex (NOLOCK)

WHERE indexid IN

(

SELECT indexid FROM @Tbl

)

END

GO

ALTER PROC [dbo].[Stp\_SearchIndex\_TurnonSearchIndexes]

@NewIndexId INT,

@NewDocCount INT,

@NewDocSize DECIMAL(10,4),

@Indexes\_Deactivate Tp\_SearchIndexEntityType READONLY,

@GroupID INT,

@BatchGuid UNIQUEIDENTIFIER,

@AppType INT

AS

BEGIN

SET XACT\_ABORT ON

BEGIN TRANSACTION

--activate index

UPDATE SearchIndex

SET active=1,

islocked=0,

DocCount=@NewDocCount,

SizeInMb=@NewDocSize,

lastUpdate=GetDate()

WHERE indexid=@NewIndexId AND

BatchGUID=@BatchGuid AND

AppType=@AppType AND

groupnum=@GroupID

--deactivate indexes

UPDATE SearchIndex

SET active=0,

DocCount=0,

SizeInMB=0,

lastUpdate=GetDate()

WHERE indexid IN (SELECT IndexId FROM @Indexes\_Deactivate) AND

BatchGUID=@BatchGuid AND

AppType=@AppType AND

groupnum=@GroupID

--update reference in IndexLockRequestQueue\_Item table

UPDATE IndexLockRequestQueue\_Item SET NewSearchIndexID=@NewIndexId WHERE SearchIndexID in (SELECT IndexId FROM @Indexes\_Deactivate) AND IsComplete=0

COMMIT TRANSACTION

END

GO

CREATE PROC [dbo].[Stp\_IndexLockRequestQueue\_Add](@GroupID INT, @AppType TINYINT,@DateRequested DATETIME,@DateExpired DATETIME)

AS

BEGIN

DECLARE @LockQueueID INT

DECLARE @Tbl TABLE(IndexId INT)

SET XACT\_ABORT ON

BEGIN TRANSACTION

--lock all available indexes

UPDATE searchindex SET IsLocked=1,DateLeased=GETDATE(),AppType=@AppType

OUTPUT inserted.indexid INTO @Tbl

WHERE groupnum = @GroupId AND IsLocked=0 AND active=1

--create parent record

INSERT INTO IndexLockRequestQueue(GroupID,AppType,DateRequested,DateExpired,Step,DateFinished)

VALUES (@GroupID,@AppType,@DateRequested,@DateExpired,1,NULL)

SET @LockQueueID = SCOPE\_IDENTITY()

--insert locked indexes by other apps into child table

INSERT INTO IndexLockRequestQueue\_Item(LockQueueID,SearchIndexID,DateRequested,IsComplete)

SELECT @LockQueueID,indexid,@DateRequested,0

FROM searchindex (NOLOCK) WHERE groupnum = @GroupId

AND

(

(indexid NOT IN

(

SELECT indexid FROM @Tbl

) AND active=1 and IsLocked=1)

OR (IsLocked=1 AND active=0 AND AppType=3 AND @Apptype=4)

)

COMMIT TRANSACTION

SELECT LockQueueID,GroupID,AppType,DateRequested,DateExpired,Step,DateFinished

FROM IndexLockRequestQueue (NOLOCK)

WHERE LockQueueID=@LockQueueID

END

GO

CREATE PROC [dbo].[Stp\_IndexLockRequestQueue\_Item\_Add](@LockQueueID INT,@SearchIndexID INT,@DateRequested DATETIME)

AS

BEGIN

DECLARE @LockQueueIDItem INT

INSERT INTO IndexLockRequestQueue\_Item(LockQueueID,SearchIndexID,DateRequested,DateFinished,IsComplete)

VALUES (@LockQueueID,@SearchIndexID,@DateRequested,NULL,0)

SET @LockQueueIDItem = SCOPE\_IDENTITY()

SELECT LockQueueIDItem,LockQueueID,SearchIndexID,NewSearchIndexID,DateRequested,DateFinished,IsComplete

FROM IndexLockRequestQueue\_Item (NOLOCK)

WHERE LockQueueIDItem=@LockQueueIDItem

END

GO

CREATE PROC [dbo].[Stp\_IndexLockQueueItem\_GetIndexesToProcess](@LockQueueID INT,@GroupID INT, @IndexID INT=NULL, @AppType TINYINT)

AS

BEGIN

--if no indexid passing it returns any availble index ready to process

DECLARE @Tbl TABLE(IndexId INT)

UPDATE TOP (1) searchindex SET IsLocked=1,DateLeased=GETDATE(),AppType=@AppType

OUTPUT inserted.indexid INTO @Tbl

WHERE groupnum=@GroupID AND indexid=ISNULL(@IndexID,indexid)

AND IsLocked=0 AND active=1 AND

EXISTS

(

SELECT 1 FROM IndexLockRequestQueue\_Item (NOLOCK) WHERE LockQueueID=@LockQueueID AND IsComplete=0 AND ISNULL(NewSearchIndexID,SearchIndexID)=indexid

)

DECLARE @DateExpired DATETIME

SELECT @DateExpired = DateExpired FROM IndexLockRequestQueue (NOLOCK) WHERE LockQueueID=@LockQueueID

--prolong dateexpired in IndexLockRequestQueue table for two hours, if it's less than 3 hours, because you are going to work on it. Hopefully you will finish for that time

IF (DATEDIFF(hour,GETDATE(),@DateExpired)<=3)

BEGIN

UPDATE IndexLockRequestQueue

SET DateExpired=DATEADD(hour,2,DateExpired)

WHERE LockQueueID=@LockQueueID

END

SELECT indexid,indexpath,groupnum,[Type],LeaseSeconds,active,DocCount,IsLocked,lastUpdate,SizeInMB,BatchGUID,DateLeased,AppType

FROM searchindex (NOLOCK)

WHERE indexid IN

(

SELECT indexid FROM @Tbl

)

END

GO

(There are no case to be expired for sp below, because in previous sp Stp\_IndexLockQueueItem\_GetIndexesToProcess we prolong dateexpired.)

/\*it should return 0, if there are still children

it should return 1 and update parent step to 2 (complete), when no children left and unlock unused indexes

\*/

CREATE PROC [dbo].[Stp\_IndexLockRequestQueue\_Item\_AllDone](@LockQueueID INT,@AppType TINYINT,@GroupID INT)

AS

BEGIN

IF (NOT EXISTS(SELECT 1 FROM IndexLockRequestQueue\_Item (NOLOCK) WHERE LockQueueID=@LockQueueID AND IsComplete=0))

BEGIN

SET XACT\_ABORT ON

BEGIN TRANSACTION

UPDATE IndexLockRequestQueue SET Step=2 WHERE LockQueueID=@LockQueueID

--unlock all unused indexes

UPDATE SearchIndex SET IsLocked=0,DateLeased=NULL WHERE groupnum=@GroupID AND Apptype=@AppType

COMMIT TRANSACTION

RETURN 1

END

RETURN 0

END

GO

CREATE PROC [dbo].[Stp\_IndexLockRequestQueue\_Item\_SetCompleteIndex](@LockQueueID INT,@IndexID INT,@AppType TINYINT)

AS

BEGIN

SET XACT\_ABORT ON

BEGIN TRANSACTION

UPDATE IndexLockRequestQueue\_Item SET IsComplete=1,DateFinished=GETDATE() WHERE LockQueueID=@LockQueueID AND ISNULL(NewSearchIndexID,SearchIndexID)=@IndexID

--unlock index in searchindex table

UPDATE SearchIndex SET IsLocked=0,DateLeased=NULL WHERE indexid=@IndexID AND Apptype=@AppType

COMMIT TRANSACTION

END

GO