New tables:

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,

[LockAll] [bit] NOT 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)

select indexId

into @Tbl\_IndexesToExclude

from SearchIndexLockQueueRequestItem

where

Step < 2 AND

GETDATE()<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 NOT EXISTS~~

~~(~~

~~SELECT 1 FROM IndexLockRequestQueue (NOLOCK) WHERE GroupID=@GroupId AND Step<>2 AND GETDATE()<DateExpired~~

~~)~~

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 PROCEDURE [dbo].[Stp\_SearchIndex\_GetNewIndexPath]

@Group\_id INT,

@IndexType TINYINT,

@AppType TINYINT,

@LeaseSec INT,

@BatchGuid UNIQUEIDENTIFIER,

@IndexPathEmpty varchar(260)

AS

BEGIN

DECLARE @LockResult INT

DECLARE @MaxLevelNum INT

DECLARE @IndexId INT

DECLARE @IndexPath VARCHAR(260)

SELECT TOP 1 @IndexPath=indexpath,@IndexId=indexid

FROM SearchIndex (NOLOCK)

WHERE groupnum = @Group\_id and

active=0 and

IsLocked=0 and

DocCount=0 and

AppType !=Retention

--if no indexpath found select any

IF (ISNULL(@IndexPath,'')='')

BEGIN

SET @IndexId=NULL

SELECT TOP 1 @MaxLevelNum=CONVERT(INT,REVERSE(SUBSTRING(REVERSE(indexpath),2,CHARINDEX('\', REVERSE(indexpath),2)-2))),

@IndexPath=REVERSE(SUBSTRING(SUBSTRING(REVERSE(indexpath),2,LEN(indexpath)),CHARINDEX('\',SUBSTRING(REVERSE(indexpath),2,LEN(indexpath))),LEN(indexpath)))

FROM SearchIndex (NOLOCK)

WHERE groupnum = @group\_id

~~AND NOT EXISTS~~

~~(~~

~~SELECT 1 FROM IndexLockRequestQueue (NOLOCK) WHERE GroupID=@group\_id AND Step<>2 AND GETDATE()<DateExpired~~

~~)~~

ORDER BY CONVERT(INT,REVERSE(SUBSTRING(REVERSE(indexpath),2,CHARINDEX('\', REVERSE(indexpath),2)-2))) DESC

END

IF (ISNULL(@IndexPath,'')='')

BEGIN

SET @IndexPath=@IndexPathEmpty

SET @MaxLevelNum=0

END

IF (ISNULL(@IndexId,-1)=-1)

BEGIN

SET @MaxLevelNum=@MaxLevelNum+1

--because we have indexpath unique index, i don't use locking for some it will fail if both will receive same indexpath

IF @MaxLevelNum<10

SET @IndexPath=@IndexPath+'0'+CONVERT(VARCHAR(200),@MaxLevelNum)+'\'

ELSE

SET @IndexPath=@IndexPath+CONVERT(VARCHAR(200),@MaxLevelNum)+'\'

INSERT INTO SearchIndex(indexpath,groupnum,Type,LeaseSeconds,active,DocCount,IsLocked,lastUpdate,SizeInMB,BatchGUID,DateLeased,AppType)

VALUES (@IndexPath,@group\_id,@IndexType,@LeaseSec,0,0,1,GETDATE(),0,@BatchGuid,GETDATE(),@AppType)

SET @IndexId = SCOPE\_IDENTITY()

IF (@IndexId<=0)

SET @IndexPath=''

END

ELSE

BEGIN

--lock found levelnum record

UPDATE SearchIndex

SET IsLocked=1,

DateLeased=GETDATE(),

BatchGUID=@BatchGuid,

AppType=@AppType,

LeaseSeconds=@LeaseSec

WHERE IndexId=@IndexId and IsLocked=0

IF @@ROWCOUNT = 0

SET @IndexPath=''

END

IF ISNULL(@IndexPath,'')<>''

BEGIN

SELECT \*

FROM SearchIndex (NOLOCK)

WHERE indexid=@IndexId

END

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,@LockAll BIT)

AS

BEGIN

DECLARE @LockQueueID INT

DECLARE @Tbl TABLE(IndexId INT)

SET XACT\_ABORT ON

BEGIN TRANSACTION

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

OUTPUT inserted.indexid INTO @Tbl

WHERE groupnum = @GroupId AND (IsLocked=0 AND active=1) OR (IsLocked=1 AND AppType=Retention AND @AppType <> Retention)

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

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

SET @LockQueueID = SCOPE\_IDENTITY()

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

COMMIT TRANSACTION

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

FROM IndexLockRequestQueue

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

WHERE LockQueueIDItem=@LockQueueIDItem

END

GO

CREATE PROC [dbo].[Stp\_IndexLockRequestQueue\_Item\_SetComplete](@LockQueueIDItem INT,@DateFinished DATETIME)

AS

BEGIN

DECLARE @LockQueueID INT

UPDATE IndexLockRequestQueue\_Item

SET DateFinished=@DateFinished, IsComplete=1

WHERE LockQueueIDItem=@LockQueueIDItem

DECLARE @DateExpired DATETIME

SELECT @LockQueueID=LockQueueID, @DateExpired = DateExpired FROM IndexLockRequestQueue WHERE LockQueueID=

(

SELECT TOP 1 LockQueueID FROM IndexLockRequestQueue\_Item WHERE LockQueueIDItem=@LockQueueIDItem

)

--prolong dateexpired in IndexLockRequestQueue tabl for one hour, if it's less than 3 hours, because you are still alive

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

BEGIN

UPDATE IndexLockRequestQueue

SET DateExpired=DATEADD(hour,1,DateExpired)

WHERE LockQueueID=@LockQueueID

END

END

GO

CREATE PROC [dbo].[Stp\_IndexLockRequestQueue\_Update](@LockQueueID INT,@DateFinished DATETIME=NULL,@Step TINYINT)

AS

BEGIN

UPDATE IndexLockRequestQueue

SET DateFinished=@DateFinished, Step=@Step

WHERE LockQueueID=@LockQueueID

END

GO

CREATE PROC [dbo].[Stp\_SearchIndex\_ReadyLock](@LockQueueID INT,@IndexID INT, @AppType TINYINT)

AS

BEGIN

DECLARE @Tbl TABLE(IndexId INT)

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

OUTPUT inserted.indexid INTO @Tbl

WHERE indexid=@IndexID AND IsLocked=0 AND active=1

DECLARE @DateExpired DATETIME

SELECT @DateExpired = DateExpired FROM IndexLockRequestQueue 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

CREATE PROC [dbo].[Stp\_IndexLockRequestQueue\_Item\_GetInComplete](@LockQueueID INT)

AS

BEGIN

SELECT LockQueueIDItem, SearchIndexID,NewSearchIndexID

FROM IndexLockRequestQueue\_Item

WHERE LockQueueID = @LockQueueID AND IsComplete = 0

END

GO