

Re: Full Text – Index Population – HDD Access

Source:

<http://www.tech-archive.net/Archive/SQL-Server/microsoft.public.sqlserver.fulltext/2004-09/0101.html>

From: Mike Davies (*michaeld_at_wangled.com*)

Date: 09/13/04

Date: Mon, 13 Sep 2004 19:24:22 +0100

Hi Rich,

It's running on a single P4 Prescott at 3Ghz with 4GB of RAM. SQL Server currently has access to all system RAM but is sat at approx 1.7Gb during the indexing process.

Both the FullText index and the SQL database are on separate physical SATA drives, however because of an apparent incompatibility issue with Windows and running the two SATA drives on their own bus they are both running through the standard IDE bus, I imagine this is not doing the performance any favours.

Thanks for the 2005 tip, I'll give that a try.

Can you offer any other suggestions to the setup?

Thanks

Mike

"Richard Yeo" <RichardYeo@discussions.microsoft.com> wrote in message news:692389A4-ED05-42C7-B645-982E7E9CE989@microsoft.com...

> *Mike*

>

> *What hardware is it running on?*

>

> *Is the Full-Text Index on the same physical drive/lun as the database? It*

> *should be possible to increase IO throughput by placing the Full-Text*

Index

> *on its own drive.*

>

> *Are you using SCSI disks? If so are you using the faster 15k RPM disks as*

> *apposed to 10k RPM disks.*

>

> *How much RAM does the machine have?*

>

> *How many processors and what spec are they?*

>
> *Is SQL Server configured to leave some RAM free for the Operating System
and
> other Applications, i.e. if you have 2GB RAM is SQL configured to use max
> 1.5GB?*
>
> *You might want to try SQL Server 2005 Beta 2. I understand that Full-Text
> Search is much faster at indexing.*
>
> <http://msdn.microsoft.com/library/en-us/dnsq190/html/yukonftsearch.asp?frame=true>
>
> *Rich*
>
>
> *"Hilary Cotter" wrote:*
>
>> *No.*
>>
>> *There is the resource usage setting which may increase performance
during
>> the indexing process (sp_fulltext_service
'catalogname','resource_usgae',5)*
>>
>> *The more resources you can spend to improve the indexing process the
better,
>> normally you get the best bang for your buck by investing in the disk
>> subsystem. For instance placing your catalogs on a RAID 10, 1, or 01
array.
>> 01 is not considered to be a good choice as it does not offer the the
>> protection 10 does.*
>>
>> *Also your database should be raid 5 depending on the level of write
activity
>> on it.*
>>
>> *What type of content are you indexing. You will get best performance by
>> indexing text content as opposed to indexing documents in their native
>> format (doc, xls, pdf, etc).*
>>
>> *Please review the full text white paper on SQL FTS performance at:*
>>
>>
>> http://support.microsoft.com/support/sql/content/2000papers/FTS_White%20Paper.asp?GSSNB=1
>>
>> *Save this document as a zip and extract it.*
>>
>> ---
>> *Hilary Cotter*
>> *Looking for a SQL Server replication book?*
>> <http://www.nwsu.com/0974973602.html>
>>

>>
>> *"Mike Davies" <michael.davies@synsoft.co.uk> wrote in message*
>> *news:ezLfiHZmEHA.3824@TK2MSFTNGP12.phx.gbl...*
>>> *Hi,*
>>>
>>> *I'm indexing a single field on a table with 10 million rows in it, its*
>>> *currently taking about 48 hours to build.*
>>>
>>> *I've been looking into performance bottlenecks on the system and see*
that
>>> *this delay is being mainly caused by MSSearch writing massive amounts*
of
>>> *data to the FT Index location. Since the finished index is only about*
>> *400mb*
>>> *I can only imagine that the population process is rewriting the data*
again
>>> *and again to the disk as it is updated.*
>>>
>>> *Is this a fair assumption? If so is there anyway around it, maybe to*
tell
>>> *MSSearch to use ram until the population is build then write the index*
to
>>> *disk?*
>>>
>>> *Thanks*
>>>
>>> *Michael*
>>>
>>>
>>
>>
>>