Gmail **Ishan Chattopadhyaya <ichattopadhyaya@gmail.com>**

Re: First class support for node roles

29 messages

Gus Heck <gus.heck@gmail.com> Sun, Nov 21, 2021 at 4:48 AM
Reply-To: dev@solr.apache.org
To: dev@solr.apache.org

It is difficult to track everything in this thread for sure. Specifically, I don't feel my email of Wed, Nov 17, 7:43 PM (EST) has been addressed or responded to except for one difference of opinion with Jan on whether or not we should prohibit the notion of any role ever changing at runtime.

Note that I made a specific proposal for an addition to the sip, regarding start up lifecycle and adherence to DRY via zk, in the (probably not clearly expressed) hope that we can move to a general overall application principle that at runtime config info should come from zk. If we follow such a principle, we will always be in a position to add dynamic features without significant rework, and all code could hopefully reuse the same interaction patterns for consulting config values.

Neither item is addressed in the SIP currently.

Also I had put that out there as a single item so it could be focused on (simplifying the discussion I hope), and depending on how discussion proceeds, I may have other follow on items.

-Gus

On Sat, Nov 20, 2021 at 8:45 AM Ishan Chattopadhyaya <ichattopadhyaya@gmail.com> wrote:

> On Fri, 19 Nov, 2021, 7:03 pm Ilan Ginzburg, <ilansolr@gmail.com> wrote:
> > Is the request here for everybody to express again the concerns
> > already expressed in this email thread and not addressed?
>
> I think the expectation now (after we've expressed our intention to reach a lazy consensus) is that if someone wants to block this SIP from adoption, then to put forward the objections. Sort of like a veto.
>
> > I suggest instead the authors review the thread, match expressed
> > concerns with how the concern was addressed (or not addressed) and
> > provide an exhaustive list.
>
> I've summarized most of the discussion in the SIP document. If you feel I could do a better job with it, please help me with areas of improvement.
>
> > This proposal in its current form (data and overseer roles) doesn't
> > offer much that can't be reasonably achieved by other means. I'd find
> > much more value in making sure what is done now is a solid foundation
> > for the future.
>
> Fair enough, I understand your perspective. Thanks for your feedback.
>
> > Ilan
> >
> > On Thu, Nov 18, 2021 at 11:24 PM Noble Paul <noble.paul@gmail.com> wrote:
> > >
> > > After so many back and forth mails, I just can't say who has an outstanding concern and if they are already addressed or not. I think the Google doc would help us get clarity on that. Please take a moment to give your inputs
> > >
> > > On Fri, Nov 19, 2021, 9:18 AM Ishan Chattopadhyaya <ichattopadhyaya@gmail.com> wrote:
> > >>
> > >> Apologies, the vote hasn't passed formally and I was under some confusion on the process.

>>
>> I'd like to proceed with a lazy consensus and proceed to the implementation phase now.
>>
>> However, I would appreciate it if someone wants to bring out any outstanding concerns about the SIP document.
>>
>> To facilitate in-line comments, here's a temporary Google Docs version of this document.
>> https://docs.google.com/document/d/1hijvM1WX9u2TOUdLEkFYVCofLeJlv2MRZqe-ncobJVw/edit?usp=sharing
>> (I shall copy changes back to confluence eventually)
>>
>> Thanks and apologies again regarding the confusion with the voting,
>> Regards,
>> Ishan
>>
>> On Thu, Nov 18, 2021 at 9:50 PM Ishan Chattopadhyaya <ichattopadhyaya@gmail.com> wrote:
>>>
>>> The SIP passed the voting phase. Thanks for all for the feedback and insights.
>>> Looking forward to your collaboration and reviews as we implement this.
>>>
>>> On Thu, Nov 18, 2021 at 9:42 PM Ishan Chattopadhyaya <ichattopadhyaya@gmail.com> wrote:
>>>>
>>>> > It's fine if we don't provide any ability for runtime modification of roles at this time but I'm leery of precluding it in the future.
>>>>
>>>> In future, the necessity for such a facility can dictate our course of action. We cannot lay down rules cast in stone for functionality that we can't foresee yet.
>>>>
>>>> On Thu, Nov 18, 2021 at 9:40 PM Ishan Chattopadhyaya <ichattopadhyaya@gmail.com> wrote:
>>>>>
>>>>> Thanks Jan, I added both those points to the SIP document in the Notes section.
>>>>>
>>>>> On Thu, Nov 18, 2021 at 7:18 PM Jan Høydahl <jan.asf@cominvent.com> wrote:
>>>>>>
>>>>>> 18. nov. 2021 kl. 01:43 skrev Gus Heck <gus.heck@gmail.com>:
>>>>>>
>>>>>>
>>>>>> 2) Roles will not be checked by loading config from disk or caching disk config in memory. (zk ONLY source of truth)
>>>>>>
>>>>>>
>>>>>> It sounds a bit backward for a local node to first parse solr.node.roles, determine its local set of roles, then publish them to Zookeeper, and then read back its own roles from ZK.
>>>>>> Code that only needs to determine "Do I have the XXX role?" or find out "What roles do I have" should be able to fetch the (static) roles from some roles utility class without consulting ZK.
>>>>>> Code that needs to check what nodes have a certain role (such as placement) would obviously need ot consult ZK.
>>>>>>
>>>>>> Perhaps the SIP should also state some Non-goals or assertions such as
>>>>>> * Roles are static and immutable (also in zk) for the entire life cycle of a node
>>>>>>
>>>>>> I also think we should state that the bar for adding new roles should be high so it is not abused as any other tag or label for any tiny feature. It should be reserved for functionality that may benefit from a dedicated set of nodes. That may be clear already, but you never know...
>>>>>>
>>>>>> Jan

To unsubscribe, e-mail: dev-unsubscribe@solr.apache.org
For additional commands, e-mail: dev-help@solr.apache.org

--
http://www.needhamsoftware.com (work)
http://www.the111shift.com (play)

Ishan Chattopadhyaya <ichattopadhyaya@gmail.com> Sun, Nov 21, 2021 at 9:08 AM
To: dev@solr.apache.org

Hi Gus,
Thanks for bringing it up again. Initially, I wasn't clear what you meant and mistook them for gish gallop.

These were the things I am/was confused about (Noble explained it to me what you mean):
- adherence to DRY via zk
- runtime config info should come from zk
- add dynamic features
- same interaction patterns for consulting config values.

Now, after Noble's help, I think I understand your motivations.
- DRY = Don't Repeat Yourself ;-)
- You're suggesting that we have a standard way to have role specific configurations (in ZK), so that a new role added later can access the configurations in a standard way

This is a very good suggestion, and makes complete sense now. This was definitely a missing piece!

Here's a proposal to address that (adding it to Google Docs: https://docs.google.com/document/d/1hijvM1WX9u2TOUdLEkFYVCofLeJlv2MRZqe-ncobJVw/edit):
- Per Role ZNode
- Every role's ZNode to hold configuration data (this is the key addition that you're asking for, IIUC)
- Children of role ZNodes to be list of ephemeral solr nodes

We can iterate over this on the Google Docs (internal representation ZK section) with inline commenting if we need to.

Please let us know how that sounds.
Regards,
Ishan

[Quoted text hidden]

Ishan Chattopadhyaya <ichattopadhyaya@gmail.com> Sun, Nov 21, 2021 at 10:28 AM
To: dev@solr.apache.org

I'll add the lifecycle details to the document as well.
[Quoted text hidden]

Ishan Chattopadhyaya <ichattopadhyaya@gmail.com> Sun, Nov 21, 2021 at 10:46 AM
To: dev@solr.apache.org

Added both your suggestions to Google Docs for inline commenting and the SIP document at confluence. Thanks for the feedback.
[Quoted text hidden]

Gus Heck <gus.heck@gmail.com> Sun, Nov 21, 2021 at 2:23 PM
Reply-To: dev@solr.apache.org
To: dev@solr.apache.org

Thanks for adding that :) and Thanks to Nobel for translating my ravings :). Sorry about the acronym. Amusingly I had to look up gish gallop to understand what you meant :) I can assure you that I will never gish gallop you or anyone on the list, and I find such techniques repugnant. If I'm not making sense, I'm likely covering ground too fast, skipping things that are in my head but not yours, expressing too many tangents (when I start nesting parentheses this is (usually) a sign I'm wandering too much), or I am simply mistaken about something. Always feel free to ask me to explain!

-Gus
[Quoted text hidden]

Ishan Chattopadhyaya <ichattopadhyaya@gmail.com> Tue, Nov 23, 2021 at 5:39 PM

To: dev@solr.apache.org

This proposal has passed with lazy consensus. We can proceed to the implementation phase.
Thanks to everyone for feedback, esp. Gus for the patience.
[Quoted text hidden]

Jan Høydahl <jan.asf@cominvent.com> Tue, Nov 23, 2021 at 7:03 PM
Reply-To: dev@solr.apache.org
To: dev@solr.apache.org

There is still discussion and disagreement in the Google doc.

The Google doc proposes that zk role structure is 1:1 for all roles -- except the "data" role.
I assume the reason for that choice, even if not spelled out in the doc, is to optimize for large clusters with 1000 nodes, and thus avoid listing 1000 nodes under "data" if a node (if many nodes are started without explicit roles).
I believe this thread had clear arguments for ALL roles being a simple 1:1 mapping, and no role be treated special. Also that all roles are defalt "on" if no roles specified (and perhaps make a generic concept of roles choosing whether to be default=off as an exception).

I see that zk is impl detail and that the public HTTP and Java API for roles will have the complete list, amended by some logic in that finds nodes without roles and adds "data" for those.
While the public APIs are the most important, I'm still questioning whether this "optimization" is justified by real performace concerns of ZK not scaling with this many ephemeral znodes?
The concern needs to be major in order to deviate from the obvious simple 1:1 structure everywhere, and a design where "data" role is hardcoded to be the only "default" role must also be justified vs treating all roles the same.

Where can we find the "latest" SIP proposal now? Is Confluence the official latest version?

Jan
[Quoted text hidden]

Mike Drob <mdrob@mdrob.com> Tue, Nov 23, 2021 at 8:09 PM
Reply-To: dev@solr.apache.org
To: dev@solr.apache.org

-1, I would like to see a proposal on list where it will be permanently available in the archives instead of being directed to a Google doc which can be edited at any point in time.

This has been an incredibly difficult proposal to keep track of.

Mike
[Quoted text hidden]

Gus Heck <gus.heck@gmail.com> Tue, Nov 23, 2021 at 8:19 PM
Reply-To: dev@solr.apache.org
To: dev@solr.apache.org

IMHO two things must happen before lazy consensus wait period can begin:
1) The wiki must be updated to reflect the results of discussion in the google doc.
2) The fact that the wiki has been updated, and that you think the discussion is at an end must be posted here, preferably with a summary on the list.

If it didn't happen on the list it didn't happen.

Also the google doc has not been notifying me of changes/responses. So a note here when you've responded and some time to respond before the above steps are taken would be appreciated.

-Gus
[Quoted text hidden]

Gus Heck <gus.heck@gmail.com> Tue, Nov 23, 2021 at 8:24 PM
Reply-To: dev@solr.apache.org
To: dev@solr.apache.org

Also, I don't know that it's defined anywhere, but I feel that declaring lazy consensus on a SIP in less than a week is rushing things. Some folks won't have the flexibility to address things daily and I've been uncomfortably pulled away from paid work trying to keep up with this as is. Lazy consensus should include at least one weekend (IMHO)
[Quoted text hidden]

Ishan Chattopadhyaya <ichattopadhyaya@gmail.com> Tue, Nov 23, 2021 at 9:07 PM
To: dev@solr.apache.org

Okay, sure, I'll rescind my assertion about having reached a lazy consensus. I'll work through all the recent comments and feedback.
I think at the last moment, there was an edit by Noble to the Google Doc that hasn't made it into the SIP document, and I'll review that change, the recent comments and get back here tomorrow. Other than that last moment change, the SIP document is up to date.
[Quoted text hidden]

Ishan Chattopadhyaya <ichattopadhyaya@gmail.com> Tue, Nov 23, 2021 at 9:09 PM
To: dev@solr.apache.org

> -1, I would like to see a proposal on list where it will be permanently available
> in the archives instead of being directed to a Google doc which can be edited at any point in time.
> This has been an incredibly difficult proposal to keep track of.

This entire SIP process has been incredibly difficult to coordinate. The idea of doing this on a mail thread was a terrible one. There's no archive available publicly for this list that doesn't break the threading functionality (I guess someone among us is using some funky mail client that's messing with Pony Mail). JIRA for discussion would've been much better. I suppose JIRA used to have threaded discussions once upon a time, that would've been very handy here.
[Quoted text hidden]

Ishan Chattopadhyaya <ichattopadhyaya@gmail.com> Tue, Nov 23, 2021 at 9:12 PM
To: dev@solr.apache.org

> -1, I would like to see a proposal on list where it will be permanently available

It will be available in the confluence document (as it currently is). The google docs is just for collecting temporary feedback, thanks to easy inline commenting capability. We promised to consolidate feedback in the gdoc and sync back to the SIP document in confluence. This is reasonable enough expectations, and shouldn't be a worry enough for a -1.

As of right now, while trying to address some concerns, some edits have been made in gdocs by Noble that we'll sync back into confluence; so that's the only divergence.
[Quoted text hidden]

Mike Drob <mdrob@mdrob.com> Tue, Nov 23, 2021 at 9:37 PM
Reply-To: dev@solr.apache.org
To: dev@solr.apache.org

I should clarify, that my -1 was specifically about reaching lazy consensus and not about that proposal itself.
[Quoted text hidden]

Ishan Chattopadhyaya <ichattopadhyaya@gmail.com> Wed, Nov 24, 2021 at 9:03 AM
To: dev@solr.apache.org

I've synced back all changes to confluence. Both represent the same document as of now.
Please review and suggest if there are any outstanding concerns. Thanks for all the feedback.

I wish to proceed with the implementation based on lazy consensus again.
[Quoted text hidden]

Gus Heck <gus.heck@gmail.com> Wed, Nov 24, 2021 at 8:18 PM

Reply-To: dev@solr.apache.org
To: dev@solr.apache.org

Left some additional comments in the google doc, extracting the key point for others here so they can know if they want to read the google doc commentary:

There is presently disagreement with my proposal that the sip specify both how we track configuration (which I have been referring to as "capability") and runtime state ("providing"). I see the primary value of having a role concept as one of organizing the code and configuration in a way that is easy to understand, which is why I'm in favor of this SiP overall.

If you don't want to tackle implementing the runtime state bit I'm ok with that as long as we have a clear plan in the SIP of how this type of information should be organized in the future.

The structure proposed in the SIP for information in zk seems to map more clearly to runtime state than configuration, and seems like it would inhibit a natural representation of such.

Finally the structure in the SIP for zk information doesn't specify if some of the nodes are ephemeral, and reliable for a "current" list or if they are persistent over time. I feel like we may not yet have a plan for how much information about nodes (n general) should be retained while a node is down, which maybe needs to underlie this design.

-Gus

[Quoted text hidden]

Ishan Chattopadhyaya <ichattopadhyaya@gmail.com> Wed, Nov 24, 2021 at 9:05 PM
To: dev@solr.apache.org

On Wed, Nov 24, 2021 at 8:19 PM Gus Heck <gus.heck@gmail.com> wrote:

> Left some additional comments in the google doc, extracting the key point for others here so they can know if they want to read the google doc commentary:

I'm addressing them here, inline.

> There is presently disagreement with my proposal that the sip specify both how we track configuration (which I have been referring to as "capability") and runtime state ("providing"). I see the primary value of having a role concept as one of organizing the code and configuration in a way that is easy to understand, which is why I'm in favor of this SiP overall.

I purposely don't want to complicate this design by allowing for "capability" vs "currently providing" to be represented as first class citizens. Besides the OVERSEER role (preferred overseer), I can't think of concrete scenarios where such a concept could be applicable, and hence I don't want to get ahead of myself in trying to address how to solve that. Since the current proposal for dealing with roles is flexible and generic enough, I don't think adding such extensions would be a problem. However, I don't want to go down that rabbithole at this point in time.

> If you don't want to tackle implementing the runtime state bit I'm ok with that as long as we have a clear plan in the SIP of how this type of information should be organized in the future.
> The structure proposed in the SIP for information in zk seems to map more clearly to runtime state than configuration, and seems like it would inhibit a natural representation of such.

The structure proposed allows for role specific configurations (if needed). It doesn't provide for node specific configuration in ZK, since all node properties should come from either solr.xml or sysprops. At present, I don't think we ever have node specific info in ZK in any part of Solr, and I think we shouldn't go that route here.

> Finally the structure in the SIP for zk information doesn't specify if some of the nodes are ephemeral, and reliable for a "current" list or if they are persistent over time. I feel like we may not yet have a plan for how much information about nodes (n general) should be retained while a node is down, which maybe needs to underlie this design.

All the node names under the roles znodes are ephemeral. Added this to the document (confluence and gdoc both).
[Quoted text hidden]

Ishan Chattopadhyaya <ichattopadhyaya@gmail.com> Wed, Nov 24, 2021 at 9:07 PM
To: dev@solr.apache.org

On Wed, Nov 24, 2021 at 9:05 PM Ishan Chattopadhyaya <ichattopadhyaya@gmail.com> wrote:

> On Wed, Nov 24, 2021 at 8:19 PM Gus Heck <gus.heck@gmail.com> wrote:
> > Left some additional comments in the google doc, extracting the key point for others here so they can know if they want to read the google doc commentary:
>
> I'm addressing them here, inline.
>
> > There is presently disagreement with my proposal that the sip specify both how we track configuration (which I have been referring to as "capability") and runtime state ("providing"). I see the primary value of having a role concept as one of organizing the code and configuration in a way that is easy to understand, which is why I'm in favor of this SiP overall.
>
> I purposely don't want to complicate this design by allowing for "capability" vs "currently providing" to be represented as first class citizens. Besides the OVERSEER role (preferred overseer), I can't think of concrete scenarios where such a concept could be applicable, and hence I don't want to get ahead of myself in trying to address how to solve that. Since the current proposal for dealing with roles is flexible and generic enough, I don't think adding such extensions would be a problem. However, I don't want to go down that rabbithole at this point in time.
>
> > If you don't want to tackle implementing the runtime state bit I'm ok with that as long as we have a clear plan in the SIP of how this type of information should be organized in the future.
> > The structure proposed in the SIP for information in zk seems to map more clearly to runtime state than configuration, and seems like it would inhibit a natural representation of such.
>
> The structure proposed allows for role specific configurations (if needed). It doesn't provide for node specific configuration in ZK, since all node properties should come from either solr.xml or sysprops. At present, I don't think we ever have node specific ~~info~~ **configuration** in ZK in any part of Solr, and I think we shouldn't go that route here.

Correction: "At present, I don't think we ever have node specific **configuration** in ZK in any part of Solr, and I think we shouldn't go that route here."
[Quoted text hidden]

Gus Heck <gus.heck@gmail.com> Wed, Nov 24, 2021 at 9:40 PM
Reply-To: dev@solr.apache.org
To: dev@solr.apache.org

So we are potentially talking past each other. Let me re-clarify since you've interpreted "configuration" differently (I think). What I'm referring to in ZK is the "reflection of configuration" not the actual configuration. This is why I'm asking the question of how much info should be retained about a node after it goes down. Case in point zookeeper embedded: The code might take different actions if it can see that a node that had a running zookeeper did exist (and thus may return) vs a situation where there's only 2 nodes playing zookeeper roles right now and no evidence of a third ever having existed.

Also you say you can't imagine anything beyond overseer... well I could imagine that a cluster would want to say these 10 machines (but not the other 40) are allowed to have zookeeper, and I want the cluster to to have 3 node level of redundancy in zookeeper, and a new zookeeper should be created and the data replicated to it if a zk node is lost for more than 30 minutes.
[Quoted text hidden]

Ilan Ginzburg <ilansolr@gmail.com> Wed, Nov 24, 2021 at 9:41 PM
Reply-To: dev@solr.apache.org
To: dev@solr.apache.org

I agree with Ishan that "currently providing" shouldn't be a current concern for the SIP.
Moreover I think it's a very role specific notion, that is not a good fit in the roles framework, so maybe should never be part of this SIP.

Even on the Overseer example. Suppose we changed the current design to have a "main" node being Overseer and a secondary to be somewhat active standby in case the main one goes down (imagine some form of replication between the two?). Both would have the "Overseer" role, but what they provide would be different. Overseer specific code should be able to know what is provided by which node, because a node trying to use the Overseer service would have to talk to a specific one of the two, not just one that currently provides.

Same for data nodes. A data node can have replicas on it. But it may not. If it doesn't, shall we say it has the capability but does not provide? Of course not, nobody would suggest that. Because we know that tracking the actual data is not a role level responsibility but a data (collection/shard/replica etc) tracking responsibility, done using all the nice structures the code maintains in ZK and in caches all over the place.

Ilan
[Quoted text hidden]

Gus Heck <gus.heck@gmail.com> Wed, Nov 24, 2021 at 10:38 PM
Reply-To: dev@solr.apache.org
To: dev@solr.apache.org

You do make a good point that the details can be very role specific. Maybe one way to provide for the future is to place some requirements on roles, and designate that role-specific coordination information be placed within the subtree for that role under /node-roles/<rolename>/ To facilitate this we could

1. move the currently designed set of ephemeral nodes down one level into am "/available" node that reflects which nodes are currently up with the role (exactly what they do in the present design)
2. Roles must store role related info in the data for the role node, or in tree(s) that are peers to /available as they see fit.
3. Roles must have documentation detailing what they store where and how it's used.
4. Roles (beyond those implemented in this SIP) should be discussed in their own SIP (enforcing the non-trivial label concept we all seem to agree on)

The bit that's not clear is where that documentation would live. It's not really material for the users, but more for the developers.
[Quoted text hidden]

Ishan Chattopadhyaya <ichattopadhyaya@gmail.com> Thu, Nov 25, 2021 at 1:50 AM
To: dev@solr.apache.org

> Moreover I think it's a very role specific notion, that is not a good
> fit in the roles framework, so maybe should never be part of this SIP.

Regarding "capable" vs "currently providing", I agree with Ilan here ^

On Wed, Nov 24, 2021 at 10:38 PM Gus Heck <gus.heck@gmail.com> wrote:

> You do make a good point that the details can be very role specific. Maybe one way to provide for the future is to place some requirements on roles, and designate that role-specific coordination information be placed within the subtree for that role under /node-roles/<rolename>/ To facilitate this we could
>
> 1. move the currently designed set of ephemeral nodes down one level into am "/available" node that

> reflects which nodes are currently up with the role (exactly what they do in the present design)
> 2. Roles must store role related info in the data for the role node, or in tree(s) that are peers to /available as they see fit.
> 3. Roles must have documentation detailing what they store where and how it's used.
> 4. Roles (beyond those implemented in this SIP) should be discussed in their own SIP (enforcing the non-trivial label concept we all seem to agree on)

I am not in favour of complicating the currently proposed design by doing any of this now ^ If we need to, we can discuss this then.
[Quoted text hidden]

Gus Heck <gus.heck@gmail.com> Thu, Nov 25, 2021 at 4:35 AM
Reply-To: dev@solr.apache.org
To: dev@solr.apache.org

Things I think still need to be clarified to define "Role" explicitly. Primarily to guide future implementations. Things I can think of include:

- What the bar for something to be a role vs being more appropriate as a property/tag type concept. Essentially what is and is not expected (at this time) to be a role. What's the process for this? Do we want a SIP per role to ensure good discussion? Are there some simple things a role might do that are "normal" and don't require a SIP? Or maybe no sip generally?
- Where a role is defined (enum? class? which module? core? solrj?) and what motivates that choice (specifics such as methods/properties can be left to impl of course)
- Naming conventions for roles... camel case? snake case? all caps? what characters are allowed, Possibly discourage any ideas with role names that embed config options such as overseer_priority_1 and any parsing of role names.

If we answer those and bundle them with the 3 sections under the example api calls into a "Roles" section or "Definition of Roles" or something like that it seems good to me.

Sure we could discuss any of this independently for each role in each sip but if we put guidance here we probably can skip that discussion for N out of M sips/tickets in the future (N being maximized if we choose well and write with clarity). Thus we'd be saving time in the long run and reduce the possibility that people propose something awkward and it slips through on lazy consensus.

And of course it's Apache software so it can all be changed later by a vote or concensus. :)

-Gus
[Quoted text hidden]

Jan Høydahl <jan.asf@cominvent.com> Thu, Nov 25, 2021 at 5:25 AM
Reply-To: dev@solr.apache.org
To: dev@solr.apache.org

Very good points to clarify Gus.

Also, as mentioned in Google Doc, I think the role framework should clarify the expectations for a node having a role.
Does it mean that the node is eligible for - "MAY" - a certain feature/task, or does it mean that the node "MUST" execute that task.
I think "MAY" is the correct interpretation.
- Nodes with "data" role MAY host replicas (i.e. nodes without CANNOT)
- Nodes with (future) "zk" role MAY run zk (i.e. nodes without the role CANNOT)
- Nodes with (imaginary) "worker" role MAY execute streaming map/reduce work
- Nodes with (imaginary) "ingest" role MAY run Tika parsing, OCR, data prepping etc

If we adopt the MAY definition for roles instead of MUST, it makes sense that all roles are enabled by default, as a single-node Solr would be expected to do all of the above. Once you start specializing nodes for certain roles, you need to start all other nodes without those roles. Thus a best practice would be to ALWAYS specify roles on all nodes if you want to specialize roles.

Jan
[Quoted text hidden]

Ishan Chattopadhyaya <ichattopadhyaya@gmail.com> Thu, Nov 25, 2021 at 11:55 AM
To: dev@solr.apache.org

> On Thu, Nov 25, 2021 at 5:25 AM Jan Høydahl <jan.asf@cominvent.com> wrote:
> > Very good points to clarify Gus.
> >
> > Also, as mentioned in Google Doc, I think the role framework should clarify the expectations for a node having a role.
> > Does it mean that the node is eligible for - "MAY" - a certain feature/task, or does it mean that the node "MUST" execute that task.
> > I think "MAY" is the correct interpretation.
> > - Nodes with "data" role MAY host replicas (i.e. nodes without CANNOT)
> > - Nodes with (future) "zk" role MAY run zk (i.e. nodes without the role CANNOT)
> > - Nodes with (imaginary) "worker" role MAY execute streaming map/reduce work
> > - Nodes with (imaginary) "ingest" role MAY run Tika parsing, OCR, data prepping etc

Thanks Jan, I've copied (with minor edits) these points over to the SIP document (in the "What is a role" section).
[Quoted text hidden]

Noble Paul <noble.paul@gmail.com> Thu, Nov 25, 2021 at 12:00 PM
Reply-To: dev@solr.apache.org
To: dev@solr.apache.org

> On Thu, Nov 25, 2021 at 10:05 AM Gus Heck <gus.heck@gmail.com> wrote:
> > Things I think still need to be clarified to define "Role" explicitly. Primarily to guide future implementations. Things I can think of include:
> >
> > - What the bar for something to be a role vs being more appropriate as a property/tag type concept. Essentially what is and is not expected (at this time) to be a role. What's the process for this? Do we want a SIP per role to ensure good discussion? Are there some simple things a role might do that are "normal" and don't require a SIP? Or maybe no sip generally?

We should not dictate how future functionality is added into Solr. Anyway nobody will be able to add a new functionality without discussion in the community

> > - Where a role is defined (enum? class? which module? core? solrj?) and what motivates that choice (specifics such as methods/properties can be left to impl of course)

This is an implementation detail . Should we discuss this in the PR?

> > - Naming conventions for roles... camel case? snake case? all caps? what characters are allowed, Possibly discourage any ideas with role names that embed config options such as overseer_priority_1 and any parsing of role names.

It should be a part of the new functionality and the associated ticket. We are only creating 2 roles as a part of this ticket. "overseer" , "data" .
[Quoted text hidden]

--

Noble Paul

Ishan Chattopadhyaya <ichattopadhyaya@gmail.com> Thu, Nov 25, 2021 at 12:04 PM
To: dev@solr.apache.org

> On Thu, Nov 25, 2021 at 4:35 AM Gus Heck <gus.heck@gmail.com> wrote:
> > Things I think still need to be clarified to define "Role" explicitly. Primarily to guide future implementations. Things I can think of include:

> - What the bar for something to be a role vs being more appropriate as a property/tag type concept. Essentially what is and is not expected (at this time) to be a role.

We don't wish to set a bar for anything at this point, since I can't imagine all future innovation that can happen. All we can do is present representative examples. Common sense will dictate what is useful to have as a role, and clearly we can't set a bar for that.

> - What's the process for this? Do we want a SIP per role to ensure good discussion? Are there some simple things a role might do that are "normal" and don't require a SIP? Or maybe no sip generally?

The community can decide on a case by case basis using established community processes. I don't see the need to define a new process for this.

> - Where a role is defined (enum? class? which module? core? solrj?) and what motivates that choice (specifics such as methods/properties can be left to impl of course)

This can be discussed during the implementation phase. This SIP is for the general concept, public interface and other non-code concerns. I don't want to start another 20 page thread here debating enum vs class and other things that might tickle someone's fancy. Lets cross the bridge once we get to it.

> - Naming conventions for roles... camel case? snake case? all caps? what characters are allowed, Possibly discourage any ideas with role names that embed config options such as overseer_priority_1 and any parsing of role names.

We can discuss that on a case by case basis when new roles are introduced. I've already proposed that the sysprops for roles should be in lowercase, which rules out camelCase.
[Quoted text hidden]

Ishan Chattopadhyaya <ichattopadhyaya@gmail.com> Thu, Nov 25, 2021 at 12:09 PM
To: dev@solr.apache.org

> Maybe one way to provide for the future is to place some requirements on roles, and designate that role-specific coordination information
> be placed within the subtree for that role under /node-roles/<rolename>/ To facilitate this we could
> 1. move the currently designed set of ephemeral nodes down one level into am "/available" node that reflects which nodes are currently up with the role (exactly what they do in the present design)

We've introduced nesting in the ZK structure now. Ephemeral nodes for Solr node names will now be placed at /node-roles/<rolename>/nodes/<ephemeralnode>.

[Quoted text hidden]

Ishan Chattopadhyaya <ichattopadhyaya@gmail.com> Thu, Nov 25, 2021 at 12:10 PM
To: dev@solr.apache.org

We have closed the Google Doc for further comments, and all changes have been incorporated into the confluence document: https://cwiki.apache.org/confluence/display/SOLR/SIP-15+Node+roles
Thanks for all the comments and feedback.
[Quoted text hidden]