

Item 136

#### git\_comments:

1. \* AM ACL disabled. \*
2. Should not reach AM even for second and third times too.
3. The third time around, app is completed, so should go to JHS
4. test only applicable when AM not reachable

#### git\_commits:

1. **summary:** MAPREDUCE-3251. Network ACLs can prevent some clients to talk to MR AM. Improved the earlier patch to not to JobHistoryServer repeatedly. Contributed by Anupam Seth.  
**message:** MAPREDUCE-3251. Network ACLs can prevent some clients to talk to MR AM. Improved the earlier patch to not to JobHistoryServer repeatedly. Contributed by Anupam Seth. svn merge --ignore-ancestry -c 1229787 ../../trunk/ git-svn-id: <https://svn.apache.org/repos/asf/hadoop/common/branches/branch-0.23@1229789> 13f79535-47bb-0310-9956-ffa450edef68

#### github\_issues:

#### github\_issues\_comments:

#### github\_pulls:

#### github\_pulls\_comments:

#### github\_pulls\_reviews:

#### jira\_issues:

1. **summary:** Network ACLs can prevent some clients to talk to MR ApplicationMaster  
**description:** In 0.20.xxx, the JobClient while polling goes to JT to get the job status. With YARN, AM can be launched on any port and the client will have to have ACL open to that port to talk to AM and get the job status. When the client is within the same grid network access to AM is not a problem. But some applications may have one installation per set of clusters and may launch jobs even across such sets (on job trackers in another set of clusters). For that to work only the JT port needs to be open currently. In case of YARN, all ports will have to be opened up for things to work. That would be a security no-no. There are two possible solutions: 1) Make the job client only talk to RM (as an option) to get the job status. 2) Limit the range of ports AM can listen on. Option 2) may not be favorable as there is no direct OS API to find a free port.
2. **summary:** Network ACLs can prevent some clients to talk to MR ApplicationMaster  
**description:** In 0.20.xxx, the JobClient while polling goes to JT to get the job status. With YARN, AM can be launched on any port and the client will have to have ACL open to that port to talk to AM and get the job status. When the client is within the same grid network access to AM is not a problem. But some applications may have one installation per set of clusters and may launch jobs even across such sets (on job trackers in another set of clusters). For that to work only the JT port needs to be open currently. In case of YARN, all ports will have to be opened up for things to work. That would be a security no-no. There are two possible solutions: 1) Make the job client only talk to RM (as an option) to get the job status. 2) Limit the range of ports AM can listen on. Option 2) may not be favorable as there is no direct OS API to find a free port.
3. **summary:** Network ACLs can prevent some clients to talk to MR ApplicationMaster  
**description:** In 0.20.xxx, the JobClient while polling goes to JT to get the job status. With YARN, AM can be launched on any port and the client will have to have ACL open to that port to talk to AM and get the job status. When the client is within the same grid network access to AM is not a problem. But some applications may have one installation per set of clusters and may launch jobs even across such sets (on job trackers in another set of clusters). For that to work only the JT port needs to be open currently. In case of YARN, all ports will have to be opened up for things to work. That would be a security no-no. There are two possible solutions: 1) Make the job client only talk to RM (as an option) to get the job

status. 2) Limit the range of ports AM can listen on. Option 2) may not be favorable as there is no direct OS API to find a free port.

4. **summary:** Network ACLs can prevent some clients to talk to MR ApplicationMaster

**description:** In 0.20.xxx, the JobClient while polling goes to JT to get the job status. With YARN, AM can be launched on any port and the client will have to have ACL open to that port to talk to AM and get the job status. When the client is within the same grid network access to AM is not a problem. But some applications may have one installation per set of clusters and may launch jobs even across such sets (on job trackers in another set of clusters). For that to work only the JT port needs to be open currently. In case of YARN, all ports will have to be opened up for things to work. That would be a security no-no. There are two possible solutions: 1) Make the job client only talk to RM (as an option) to get the job status. 2) Limit the range of ports AM can listen on. Option 2) may not be favorable as there is no direct OS API to find a free port.

5. **summary:** Network ACLs can prevent some clients to talk to MR ApplicationMaster

**description:** In 0.20.xxx, the JobClient while polling goes to JT to get the job status. With YARN, AM can be launched on any port and the client will have to have ACL open to that port to talk to AM and get the job status. When the client is within the same grid network access to AM is not a problem. But some applications may have one installation per set of clusters and may launch jobs even across such sets (on job trackers in another set of clusters). For that to work only the JT port needs to be open currently. In case of YARN, all ports will have to be opened up for things to work. That would be a security no-no. There are two possible solutions: 1) Make the job client only talk to RM (as an option) to get the job status. 2) Limit the range of ports AM can listen on. Option 2) may not be favorable as there is no direct OS API to find a free port.

6. **summary:** Network ACLs can prevent some clients to talk to MR ApplicationMaster

**description:** In 0.20.xxx, the JobClient while polling goes to JT to get the job status. With YARN, AM can be launched on any port and the client will have to have ACL open to that port to talk to AM and get the job status. When the client is within the same grid network access to AM is not a problem. But some applications may have one installation per set of clusters and may launch jobs even across such sets (on job trackers in another set of clusters). For that to work only the JT port needs to be open currently. In case of YARN, all ports will have to be opened up for things to work. That would be a security no-no. There are two possible solutions: 1) Make the job client only talk to RM (as an option) to get the job status. 2) Limit the range of ports AM can listen on. Option 2) may not be favorable as there is no direct OS API to find a free port.

7. **summary:** Network ACLs can prevent some clients to talk to MR ApplicationMaster

**description:** In 0.20.xxx, the JobClient while polling goes to JT to get the job status. With YARN, AM can be launched on any port and the client will have to have ACL open to that port to talk to AM and get the job status. When the client is within the same grid network access to AM is not a problem. But some applications may have one installation per set of clusters and may launch jobs even across such sets (on job trackers in another set of clusters). For that to work only the JT port needs to be open currently. In case of YARN, all ports will have to be opened up for things to work. That would be a security no-no. There are two possible solutions: 1) Make the job client only talk to RM (as an option) to get the job status. 2) Limit the range of ports AM can listen on. Option 2) may not be favorable as there is no direct OS API to find a free port.

8. **summary:** Network ACLs can prevent some clients to talk to MR ApplicationMaster

**description:** In 0.20.xxx, the JobClient while polling goes to JT to get the job status. With YARN, AM can be launched on any port and the client will have to have ACL open to that port to talk to AM and get the job status. When the client is within the same grid network access to AM is not a problem. But some applications may have one installation per set of clusters and may launch jobs even across such sets (on job trackers in another set of clusters). For that to work only the JT port needs to be open currently. In case of YARN, all ports will have to be opened up for things to work. That would be a security no-no. There are two possible solutions: 1) Make the job client only talk to RM (as an option) to get the job status. 2) Limit the range of ports AM can listen on. Option 2) may not be favorable as there is no direct OS API to find a free port.

9. **summary:** Network ACLs can prevent some clients to talk to MR ApplicationMaster

**description:** In 0.20.xxx, the JobClient while polling goes to JT to get the job status. With YARN, AM can be launched on any port and the client will have to have ACL open to that port to talk to AM and get the job status. When the client is within the same grid network access to AM is not a problem. But some applications may have one installation per set of clusters and may launch jobs even across such sets (on job trackers in another set of clusters). For that to work only the JT port needs to be open currently. In case of YARN, all ports will have to be opened up for things to work. That would be a security no-no.

There are two possible solutions: 1) Make the job client only talk to RM (as an option) to get the job status. 2) Limit the range of ports AM can listen on. Option 2) may not be favorable as there is no direct OS API to find a free port.

10. **summary:** Network ACLs can prevent some clients to talk to MR ApplicationMaster  
**description:** In 0.20.xxx, the JobClient while polling goes to JT to get the job status. With YARN, AM can be launched on any port and the client will have to have ACL open to that port to talk to AM and get the job status. When the client is within the same grid network access to AM is not a problem. But some applications may have one installation per set of clusters and may launch jobs even across such sets (on job trackers in another set of clusters). For that to work only the JT port needs to be open currently. In case of YARN, all ports will have to be opened up for things to work. That would be a security no-no. There are two possible solutions: 1) Make the job client only talk to RM (as an option) to get the job status. 2) Limit the range of ports AM can listen on. Option 2) may not be favorable as there is no direct OS API to find a free port.
11. **summary:** Network ACLs can prevent some clients to talk to MR ApplicationMaster  
**description:** In 0.20.xxx, the JobClient while polling goes to JT to get the job status. With YARN, AM can be launched on any port and the client will have to have ACL open to that port to talk to AM and get the job status. When the client is within the same grid network access to AM is not a problem. But some applications may have one installation per set of clusters and may launch jobs even across such sets (on job trackers in another set of clusters). For that to work only the JT port needs to be open currently. In case of YARN, all ports will have to be opened up for things to work. That would be a security no-no. There are two possible solutions: 1) Make the job client only talk to RM (as an option) to get the job status. 2) Limit the range of ports AM can listen on. Option 2) may not be favorable as there is no direct OS API to find a free port.
12. **summary:** Network ACLs can prevent some clients to talk to MR ApplicationMaster  
**description:** In 0.20.xxx, the JobClient while polling goes to JT to get the job status. With YARN, AM can be launched on any port and the client will have to have ACL open to that port to talk to AM and get the job status. When the client is within the same grid network access to AM is not a problem. But some applications may have one installation per set of clusters and may launch jobs even across such sets (on job trackers in another set of clusters). For that to work only the JT port needs to be open currently. In case of YARN, all ports will have to be opened up for things to work. That would be a security no-no. There are two possible solutions: 1) Make the job client only talk to RM (as an option) to get the job status. 2) Limit the range of ports AM can listen on. Option 2) may not be favorable as there is no direct OS API to find a free port.
13. **summary:** Network ACLs can prevent some clients to talk to MR ApplicationMaster  
**description:** In 0.20.xxx, the JobClient while polling goes to JT to get the job status. With YARN, AM can be launched on any port and the client will have to have ACL open to that port to talk to AM and get the job status. When the client is within the same grid network access to AM is not a problem. But some applications may have one installation per set of clusters and may launch jobs even across such sets (on job trackers in another set of clusters). For that to work only the JT port needs to be open currently. In case of YARN, all ports will have to be opened up for things to work. That would be a security no-no. There are two possible solutions: 1) Make the job client only talk to RM (as an option) to get the job status. 2) Limit the range of ports AM can listen on. Option 2) may not be favorable as there is no direct OS API to find a free port.
14. **summary:** Network ACLs can prevent some clients to talk to MR ApplicationMaster  
**description:** In 0.20.xxx, the JobClient while polling goes to JT to get the job status. With YARN, AM can be launched on any port and the client will have to have ACL open to that port to talk to AM and get the job status. When the client is within the same grid network access to AM is not a problem. But some applications may have one installation per set of clusters and may launch jobs even across such sets (on job trackers in another set of clusters). For that to work only the JT port needs to be open currently. In case of YARN, all ports will have to be opened up for things to work. That would be a security no-no. There are two possible solutions: 1) Make the job client only talk to RM (as an option) to get the job status. 2) Limit the range of ports AM can listen on. Option 2) may not be favorable as there is no direct OS API to find a free port.
15. **summary:** Network ACLs can prevent some clients to talk to MR ApplicationMaster  
**description:** In 0.20.xxx, the JobClient while polling goes to JT to get the job status. With YARN, AM can be launched on any port and the client will have to have ACL open to that port to talk to AM and get the job status. When the client is within the same grid network access to AM is not a problem. But some applications may have one installation per set of clusters and may launch jobs even across such sets (on job trackers in another set of clusters). For that to work only the JT port needs to be open currently. In

case of YARN, all ports will have to be opened up for things to work. That would be a security no-no. There are two possible solutions: 1) Make the job client only talk to RM (as an option) to get the job status. 2) Limit the range of ports AM can listen on. Option 2) may not be favorable as there is no direct OS API to find a free port.

16. **summary:** Network ACLs can prevent some clients to talk to MR ApplicationMaster  
**description:** In 0.20.xxx, the JobClient while polling goes to JT to get the job status. With YARN, AM can be launched on any port and the client will have to have ACL open to that port to talk to AM and get the job status. When the client is within the same grid network access to AM is not a problem. But some applications may have one installation per set of clusters and may launch jobs even across such sets (on job trackers in another set of clusters). For that to work only the JT port needs to be open currently. In case of YARN, all ports will have to be opened up for things to work. That would be a security no-no. There are two possible solutions: 1) Make the job client only talk to RM (as an option) to get the job status. 2) Limit the range of ports AM can listen on. Option 2) may not be favorable as there is no direct OS API to find a free port.
17. **summary:** Network ACLs can prevent some clients to talk to MR ApplicationMaster  
**description:** In 0.20.xxx, the JobClient while polling goes to JT to get the job status. With YARN, AM can be launched on any port and the client will have to have ACL open to that port to talk to AM and get the job status. When the client is within the same grid network access to AM is not a problem. But some applications may have one installation per set of clusters and may launch jobs even across such sets (on job trackers in another set of clusters). For that to work only the JT port needs to be open currently. In case of YARN, all ports will have to be opened up for things to work. That would be a security no-no. There are two possible solutions: 1) Make the job client only talk to RM (as an option) to get the job status. 2) Limit the range of ports AM can listen on. Option 2) may not be favorable as there is no direct OS API to find a free port.
18. **summary:** Network ACLs can prevent some clients to talk to MR ApplicationMaster  
**description:** In 0.20.xxx, the JobClient while polling goes to JT to get the job status. With YARN, AM can be launched on any port and the client will have to have ACL open to that port to talk to AM and get the job status. When the client is within the same grid network access to AM is not a problem. But some applications may have one installation per set of clusters and may launch jobs even across such sets (on job trackers in another set of clusters). For that to work only the JT port needs to be open currently. In case of YARN, all ports will have to be opened up for things to work. That would be a security no-no. There are two possible solutions: 1) Make the job client only talk to RM (as an option) to get the job status. 2) Limit the range of ports AM can listen on. Option 2) may not be favorable as there is no direct OS API to find a free port.
19. **summary:** Network ACLs can prevent some clients to talk to MR ApplicationMaster  
**description:** In 0.20.xxx, the JobClient while polling goes to JT to get the job status. With YARN, AM can be launched on any port and the client will have to have ACL open to that port to talk to AM and get the job status. When the client is within the same grid network access to AM is not a problem. But some applications may have one installation per set of clusters and may launch jobs even across such sets (on job trackers in another set of clusters). For that to work only the JT port needs to be open currently. In case of YARN, all ports will have to be opened up for things to work. That would be a security no-no. There are two possible solutions: 1) Make the job client only talk to RM (as an option) to get the job status. 2) Limit the range of ports AM can listen on. Option 2) may not be favorable as there is no direct OS API to find a free port.
20. **summary:** Network ACLs can prevent some clients to talk to MR ApplicationMaster  
**description:** In 0.20.xxx, the JobClient while polling goes to JT to get the job status. With YARN, AM can be launched on any port and the client will have to have ACL open to that port to talk to AM and get the job status. When the client is within the same grid network access to AM is not a problem. But some applications may have one installation per set of clusters and may launch jobs even across such sets (on job trackers in another set of clusters). For that to work only the JT port needs to be open currently. In case of YARN, all ports will have to be opened up for things to work. That would be a security no-no. There are two possible solutions: 1) Make the job client only talk to RM (as an option) to get the job status. 2) Limit the range of ports AM can listen on. Option 2) may not be favorable as there is no direct OS API to find a free port.
21. **summary:** Network ACLs can prevent some clients to talk to MR ApplicationMaster  
**description:** In 0.20.xxx, the JobClient while polling goes to JT to get the job status. With YARN, AM can be launched on any port and the client will have to have ACL open to that port to talk to AM and get the job status. When the client is within the same grid network access to AM is not a problem. But some applications may have one installation per set of clusters and may launch jobs even across such sets (on

job trackers in another set of clusters). For that to work only the JT port needs to be open currently. In case of YARN, all ports will have to be opened up for things to work. That would be a security no-no. There are two possible solutions: 1) Make the job client only talk to RM (as an option) to get the job status. 2) Limit the range of ports AM can listen on. Option 2) may not be favorable as there is no direct OS API to find a free port.

**label:** test

22. **summary:** Network ACLs can prevent some clients to talk to MR ApplicationMaster  
**description:** In 0.20.xxx, the JobClient while polling goes to JT to get the job status. With YARN, AM can be launched on any port and the client will have to have ACL open to that port to talk to AM and get the job status. When the client is within the same grid network access to AM is not a problem. But some applications may have one installation per set of clusters and may launch jobs even across such sets (on job trackers in another set of clusters). For that to work only the JT port needs to be open currently. In case of YARN, all ports will have to be opened up for things to work. That would be a security no-no. There are two possible solutions: 1) Make the job client only talk to RM (as an option) to get the job status. 2) Limit the range of ports AM can listen on. Option 2) may not be favorable as there is no direct OS API to find a free port.
23. **summary:** Network ACLs can prevent some clients to talk to MR ApplicationMaster  
**description:** In 0.20.xxx, the JobClient while polling goes to JT to get the job status. With YARN, AM can be launched on any port and the client will have to have ACL open to that port to talk to AM and get the job status. When the client is within the same grid network access to AM is not a problem. But some applications may have one installation per set of clusters and may launch jobs even across such sets (on job trackers in another set of clusters). For that to work only the JT port needs to be open currently. In case of YARN, all ports will have to be opened up for things to work. That would be a security no-no. There are two possible solutions: 1) Make the job client only talk to RM (as an option) to get the job status. 2) Limit the range of ports AM can listen on. Option 2) may not be favorable as there is no direct OS API to find a free port.
24. **summary:** Network ACLs can prevent some clients to talk to MR ApplicationMaster  
**description:** In 0.20.xxx, the JobClient while polling goes to JT to get the job status. With YARN, AM can be launched on any port and the client will have to have ACL open to that port to talk to AM and get the job status. When the client is within the same grid network access to AM is not a problem. But some applications may have one installation per set of clusters and may launch jobs even across such sets (on job trackers in another set of clusters). For that to work only the JT port needs to be open currently. In case of YARN, all ports will have to be opened up for things to work. That would be a security no-no. There are two possible solutions: 1) Make the job client only talk to RM (as an option) to get the job status. 2) Limit the range of ports AM can listen on. Option 2) may not be favorable as there is no direct OS API to find a free port.
25. **summary:** Network ACLs can prevent some clients to talk to MR ApplicationMaster  
**description:** In 0.20.xxx, the JobClient while polling goes to JT to get the job status. With YARN, AM can be launched on any port and the client will have to have ACL open to that port to talk to AM and get the job status. When the client is within the same grid network access to AM is not a problem. But some applications may have one installation per set of clusters and may launch jobs even across such sets (on job trackers in another set of clusters). For that to work only the JT port needs to be open currently. In case of YARN, all ports will have to be opened up for things to work. That would be a security no-no. There are two possible solutions: 1) Make the job client only talk to RM (as an option) to get the job status. 2) Limit the range of ports AM can listen on. Option 2) may not be favorable as there is no direct OS API to find a free port.
26. **summary:** Network ACLs can prevent some clients to talk to MR ApplicationMaster  
**description:** In 0.20.xxx, the JobClient while polling goes to JT to get the job status. With YARN, AM can be launched on any port and the client will have to have ACL open to that port to talk to AM and get the job status. When the client is within the same grid network access to AM is not a problem. But some applications may have one installation per set of clusters and may launch jobs even across such sets (on job trackers in another set of clusters). For that to work only the JT port needs to be open currently. In case of YARN, all ports will have to be opened up for things to work. That would be a security no-no. There are two possible solutions: 1) Make the job client only talk to RM (as an option) to get the job status. 2) Limit the range of ports AM can listen on. Option 2) may not be favorable as there is no direct OS API to find a free port.
27. **summary:** Network ACLs can prevent some clients to talk to MR ApplicationMaster  
**description:** In 0.20.xxx, the JobClient while polling goes to JT to get the job status. With YARN, AM can be launched on any port and the client will have to have ACL open to that port to talk to AM and get

the job status. When the client is within the same grid network access to AM is not a problem. But some applications may have one installation per set of clusters and may launch jobs even across such sets (on job trackers in another set of clusters). For that to work only the JT port needs to be open currently. In case of YARN, all ports will have to be opened up for things to work. That would be a security no-no. There are two possible solutions: 1) Make the job client only talk to RM (as an option) to get the job status. 2) Limit the range of ports AM can listen on. Option 2) may not be favorable as there is no direct OS API to find a free port.

28. **summary:** Network ACLs can prevent some clients to talk to MR ApplicationMaster

**description:** In 0.20.xxx, the JobClient while polling goes to JT to get the job status. With YARN, AM can be launched on any port and the client will have to have ACL open to that port to talk to AM and get the job status. When the client is within the same grid network access to AM is not a problem. But some applications may have one installation per set of clusters and may launch jobs even across such sets (on job trackers in another set of clusters). For that to work only the JT port needs to be open currently. In case of YARN, all ports will have to be opened up for things to work. That would be a security no-no. There are two possible solutions: 1) Make the job client only talk to RM (as an option) to get the job status. 2) Limit the range of ports AM can listen on. Option 2) may not be favorable as there is no direct OS API to find a free port.

**label:** documentation

29. **summary:** Network ACLs can prevent some clients to talk to MR ApplicationMaster

**description:** In 0.20.xxx, the JobClient while polling goes to JT to get the job status. With YARN, AM can be launched on any port and the client will have to have ACL open to that port to talk to AM and get the job status. When the client is within the same grid network access to AM is not a problem. But some applications may have one installation per set of clusters and may launch jobs even across such sets (on job trackers in another set of clusters). For that to work only the JT port needs to be open currently. In case of YARN, all ports will have to be opened up for things to work. That would be a security no-no. There are two possible solutions: 1) Make the job client only talk to RM (as an option) to get the job status. 2) Limit the range of ports AM can listen on. Option 2) may not be favorable as there is no direct OS API to find a free port.

30. **summary:** Network ACLs can prevent some clients to talk to MR ApplicationMaster

**description:** In 0.20.xxx, the JobClient while polling goes to JT to get the job status. With YARN, AM can be launched on any port and the client will have to have ACL open to that port to talk to AM and get the job status. When the client is within the same grid network access to AM is not a problem. But some applications may have one installation per set of clusters and may launch jobs even across such sets (on job trackers in another set of clusters). For that to work only the JT port needs to be open currently. In case of YARN, all ports will have to be opened up for things to work. That would be a security no-no. There are two possible solutions: 1) Make the job client only talk to RM (as an option) to get the job status. 2) Limit the range of ports AM can listen on. Option 2) may not be favorable as there is no direct OS API to find a free port.

31. **summary:** Network ACLs can prevent some clients to talk to MR ApplicationMaster

**description:** In 0.20.xxx, the JobClient while polling goes to JT to get the job status. With YARN, AM can be launched on any port and the client will have to have ACL open to that port to talk to AM and get the job status. When the client is within the same grid network access to AM is not a problem. But some applications may have one installation per set of clusters and may launch jobs even across such sets (on job trackers in another set of clusters). For that to work only the JT port needs to be open currently. In case of YARN, all ports will have to be opened up for things to work. That would be a security no-no. There are two possible solutions: 1) Make the job client only talk to RM (as an option) to get the job status. 2) Limit the range of ports AM can listen on. Option 2) may not be favorable as there is no direct OS API to find a free port.

32. **summary:** Network ACLs can prevent some clients to talk to MR ApplicationMaster

**description:** In 0.20.xxx, the JobClient while polling goes to JT to get the job status. With YARN, AM can be launched on any port and the client will have to have ACL open to that port to talk to AM and get the job status. When the client is within the same grid network access to AM is not a problem. But some applications may have one installation per set of clusters and may launch jobs even across such sets (on job trackers in another set of clusters). For that to work only the JT port needs to be open currently. In case of YARN, all ports will have to be opened up for things to work. That would be a security no-no. There are two possible solutions: 1) Make the job client only talk to RM (as an option) to get the job status. 2) Limit the range of ports AM can listen on. Option 2) may not be favorable as there is no direct OS API to find a free port.

33. **summary:** Network ACLs can prevent some clients to talk to MR ApplicationMaster

**description:** In 0.20.xxx, the JobClient while polling goes to JT to get the job status. With YARN, AM can be launched on any port and the client will have to have ACL open to that port to talk to AM and get the job status. When the client is within the same grid network access to AM is not a problem. But some applications may have one installation per set of clusters and may launch jobs even across such sets (on job trackers in another set of clusters). For that to work only the JT port needs to be open currently. In case of YARN, all ports will have to be opened up for things to work. That would be a security no-no. There are two possible solutions: 1) Make the job client only talk to RM (as an option) to get the job status. 2) Limit the range of ports AM can listen on. Option 2) may not be favorable as there is no direct OS API to find a free port.

**label:** code-design

34. **summary:** Network ACLs can prevent some clients to talk to MR ApplicationMaster

**description:** In 0.20.xxx, the JobClient while polling goes to JT to get the job status. With YARN, AM can be launched on any port and the client will have to have ACL open to that port to talk to AM and get the job status. When the client is within the same grid network access to AM is not a problem. But some applications may have one installation per set of clusters and may launch jobs even across such sets (on job trackers in another set of clusters). For that to work only the JT port needs to be open currently. In case of YARN, all ports will have to be opened up for things to work. That would be a security no-no. There are two possible solutions: 1) Make the job client only talk to RM (as an option) to get the job status. 2) Limit the range of ports AM can listen on. Option 2) may not be favorable as there is no direct OS API to find a free port.

35. **summary:** Network ACLs can prevent some clients to talk to MR ApplicationMaster

**description:** In 0.20.xxx, the JobClient while polling goes to JT to get the job status. With YARN, AM can be launched on any port and the client will have to have ACL open to that port to talk to AM and get the job status. When the client is within the same grid network access to AM is not a problem. But some applications may have one installation per set of clusters and may launch jobs even across such sets (on job trackers in another set of clusters). For that to work only the JT port needs to be open currently. In case of YARN, all ports will have to be opened up for things to work. That would be a security no-no. There are two possible solutions: 1) Make the job client only talk to RM (as an option) to get the job status. 2) Limit the range of ports AM can listen on. Option 2) may not be favorable as there is no direct OS API to find a free port.

36. **summary:** Network ACLs can prevent some clients to talk to MR ApplicationMaster

**description:** In 0.20.xxx, the JobClient while polling goes to JT to get the job status. With YARN, AM can be launched on any port and the client will have to have ACL open to that port to talk to AM and get the job status. When the client is within the same grid network access to AM is not a problem. But some applications may have one installation per set of clusters and may launch jobs even across such sets (on job trackers in another set of clusters). For that to work only the JT port needs to be open currently. In case of YARN, all ports will have to be opened up for things to work. That would be a security no-no. There are two possible solutions: 1) Make the job client only talk to RM (as an option) to get the job status. 2) Limit the range of ports AM can listen on. Option 2) may not be favorable as there is no direct OS API to find a free port.

37. **summary:** Network ACLs can prevent some clients to talk to MR ApplicationMaster

**description:** In 0.20.xxx, the JobClient while polling goes to JT to get the job status. With YARN, AM can be launched on any port and the client will have to have ACL open to that port to talk to AM and get the job status. When the client is within the same grid network access to AM is not a problem. But some applications may have one installation per set of clusters and may launch jobs even across such sets (on job trackers in another set of clusters). For that to work only the JT port needs to be open currently. In case of YARN, all ports will have to be opened up for things to work. That would be a security no-no. There are two possible solutions: 1) Make the job client only talk to RM (as an option) to get the job status. 2) Limit the range of ports AM can listen on. Option 2) may not be favorable as there is no direct OS API to find a free port.

38. **summary:** Network ACLs can prevent some clients to talk to MR ApplicationMaster

**description:** In 0.20.xxx, the JobClient while polling goes to JT to get the job status. With YARN, AM can be launched on any port and the client will have to have ACL open to that port to talk to AM and get the job status. When the client is within the same grid network access to AM is not a problem. But some applications may have one installation per set of clusters and may launch jobs even across such sets (on job trackers in another set of clusters). For that to work only the JT port needs to be open currently. In case of YARN, all ports will have to be opened up for things to work. That would be a security no-no. There are two possible solutions: 1) Make the job client only talk to RM (as an option) to get the job

status. 2) Limit the range of ports AM can listen on. Option 2) may not be favorable as there is no direct OS API to find a free port.

**label:** code-design

39. **summary:** Network ACLs can prevent some clients to talk to MR ApplicationMaster  
**description:** In 0.20.xxx, the JobClient while polling goes to JT to get the job status. With YARN, AM can be launched on any port and the client will have to have ACL open to that port to talk to AM and get the job status. When the client is within the same grid network access to AM is not a problem. But some applications may have one installation per set of clusters and may launch jobs even across such sets (on job trackers in another set of clusters). For that to work only the JT port needs to be open currently. In case of YARN, all ports will have to be opened up for things to work. That would be a security no-no. There are two possible solutions: 1) Make the job client only talk to RM (as an option) to get the job status. 2) Limit the range of ports AM can listen on. Option 2) may not be favorable as there is no direct OS API to find a free port.
40. **summary:** Network ACLs can prevent some clients to talk to MR ApplicationMaster  
**description:** In 0.20.xxx, the JobClient while polling goes to JT to get the job status. With YARN, AM can be launched on any port and the client will have to have ACL open to that port to talk to AM and get the job status. When the client is within the same grid network access to AM is not a problem. But some applications may have one installation per set of clusters and may launch jobs even across such sets (on job trackers in another set of clusters). For that to work only the JT port needs to be open currently. In case of YARN, all ports will have to be opened up for things to work. That would be a security no-no. There are two possible solutions: 1) Make the job client only talk to RM (as an option) to get the job status. 2) Limit the range of ports AM can listen on. Option 2) may not be favorable as there is no direct OS API to find a free port.
41. **summary:** Network ACLs can prevent some clients to talk to MR ApplicationMaster  
**description:** In 0.20.xxx, the JobClient while polling goes to JT to get the job status. With YARN, AM can be launched on any port and the client will have to have ACL open to that port to talk to AM and get the job status. When the client is within the same grid network access to AM is not a problem. But some applications may have one installation per set of clusters and may launch jobs even across such sets (on job trackers in another set of clusters). For that to work only the JT port needs to be open currently. In case of YARN, all ports will have to be opened up for things to work. That would be a security no-no. There are two possible solutions: 1) Make the job client only talk to RM (as an option) to get the job status. 2) Limit the range of ports AM can listen on. Option 2) may not be favorable as there is no direct OS API to find a free port.
42. **summary:** Network ACLs can prevent some clients to talk to MR ApplicationMaster  
**description:** In 0.20.xxx, the JobClient while polling goes to JT to get the job status. With YARN, AM can be launched on any port and the client will have to have ACL open to that port to talk to AM and get the job status. When the client is within the same grid network access to AM is not a problem. But some applications may have one installation per set of clusters and may launch jobs even across such sets (on job trackers in another set of clusters). For that to work only the JT port needs to be open currently. In case of YARN, all ports will have to be opened up for things to work. That would be a security no-no. There are two possible solutions: 1) Make the job client only talk to RM (as an option) to get the job status. 2) Limit the range of ports AM can listen on. Option 2) may not be favorable as there is no direct OS API to find a free port.
43. **summary:** Network ACLs can prevent some clients to talk to MR ApplicationMaster  
**description:** In 0.20.xxx, the JobClient while polling goes to JT to get the job status. With YARN, AM can be launched on any port and the client will have to have ACL open to that port to talk to AM and get the job status. When the client is within the same grid network access to AM is not a problem. But some applications may have one installation per set of clusters and may launch jobs even across such sets (on job trackers in another set of clusters). For that to work only the JT port needs to be open currently. In case of YARN, all ports will have to be opened up for things to work. That would be a security no-no. There are two possible solutions: 1) Make the job client only talk to RM (as an option) to get the job status. 2) Limit the range of ports AM can listen on. Option 2) may not be favorable as there is no direct OS API to find a free port.  
**label:** code-design
44. **summary:** Network ACLs can prevent some clients to talk to MR ApplicationMaster  
**description:** In 0.20.xxx, the JobClient while polling goes to JT to get the job status. With YARN, AM can be launched on any port and the client will have to have ACL open to that port to talk to AM and get the job status. When the client is within the same grid network access to AM is not a problem. But some applications may have one installation per set of clusters and may launch jobs even across such sets (on



job trackers in another set of clusters). For that to work only the JT port needs to be open currently. In case of YARN, all ports will have to be opened up for things to work. That would be a security no-no. There are two possible solutions: 1) Make the job client only talk to RM (as an option) to get the job status. 2) Limit the range of ports AM can listen on. Option 2) may not be favorable as there is no direct OS API to find a free port.

45. **summary:** Network ACLs can prevent some clients to talk to MR ApplicationMaster

**description:** In 0.20.xxx, the JobClient while polling goes to JT to get the job status. With YARN, AM can be launched on any port and the client will have to have ACL open to that port to talk to AM and get the job status. When the client is within the same grid network access to AM is not a problem. But some applications may have one installation per set of clusters and may launch jobs even across such sets (on job trackers in another set of clusters). For that to work only the JT port needs to be open currently. In case of YARN, all ports will have to be opened up for things to work. That would be a security no-no. There are two possible solutions: 1) Make the job client only talk to RM (as an option) to get the job status. 2) Limit the range of ports AM can listen on. Option 2) may not be favorable as there is no direct OS API to find a free port.

**label:** code-design

46. **summary:** Network ACLs can prevent some clients to talk to MR ApplicationMaster

**description:** In 0.20.xxx, the JobClient while polling goes to JT to get the job status. With YARN, AM can be launched on any port and the client will have to have ACL open to that port to talk to AM and get the job status. When the client is within the same grid network access to AM is not a problem. But some applications may have one installation per set of clusters and may launch jobs even across such sets (on job trackers in another set of clusters). For that to work only the JT port needs to be open currently. In case of YARN, all ports will have to be opened up for things to work. That would be a security no-no. There are two possible solutions: 1) Make the job client only talk to RM (as an option) to get the job status. 2) Limit the range of ports AM can listen on. Option 2) may not be favorable as there is no direct OS API to find a free port.

47. **summary:** Network ACLs can prevent some clients to talk to MR ApplicationMaster

**description:** In 0.20.xxx, the JobClient while polling goes to JT to get the job status. With YARN, AM can be launched on any port and the client will have to have ACL open to that port to talk to AM and get the job status. When the client is within the same grid network access to AM is not a problem. But some applications may have one installation per set of clusters and may launch jobs even across such sets (on job trackers in another set of clusters). For that to work only the JT port needs to be open currently. In case of YARN, all ports will have to be opened up for things to work. That would be a security no-no. There are two possible solutions: 1) Make the job client only talk to RM (as an option) to get the job status. 2) Limit the range of ports AM can listen on. Option 2) may not be favorable as there is no direct OS API to find a free port.

48. **summary:** Network ACLs can prevent some clients to talk to MR ApplicationMaster

**description:** In 0.20.xxx, the JobClient while polling goes to JT to get the job status. With YARN, AM can be launched on any port and the client will have to have ACL open to that port to talk to AM and get the job status. When the client is within the same grid network access to AM is not a problem. But some applications may have one installation per set of clusters and may launch jobs even across such sets (on job trackers in another set of clusters). For that to work only the JT port needs to be open currently. In case of YARN, all ports will have to be opened up for things to work. That would be a security no-no. There are two possible solutions: 1) Make the job client only talk to RM (as an option) to get the job status. 2) Limit the range of ports AM can listen on. Option 2) may not be favorable as there is no direct OS API to find a free port.

49. **summary:** Network ACLs can prevent some clients to talk to MR ApplicationMaster

**description:** In 0.20.xxx, the JobClient while polling goes to JT to get the job status. With YARN, AM can be launched on any port and the client will have to have ACL open to that port to talk to AM and get the job status. When the client is within the same grid network access to AM is not a problem. But some applications may have one installation per set of clusters and may launch jobs even across such sets (on job trackers in another set of clusters). For that to work only the JT port needs to be open currently. In case of YARN, all ports will have to be opened up for things to work. That would be a security no-no. There are two possible solutions: 1) Make the job client only talk to RM (as an option) to get the job status. 2) Limit the range of ports AM can listen on. Option 2) may not be favorable as there is no direct OS API to find a free port.

50. **summary:** Network ACLs can prevent some clients to talk to MR ApplicationMaster

**description:** In 0.20.xxx, the JobClient while polling goes to JT to get the job status. With YARN, AM can be launched on any port and the client will have to have ACL open to that port to talk to AM and get

the job status. When the client is within the same grid network access to AM is not a problem. But some applications may have one installation per set of clusters and may launch jobs even across such sets (on job trackers in another set of clusters). For that to work only the JT port needs to be open currently. In case of YARN, all ports will have to be opened up for things to work. That would be a security no-no. There are two possible solutions: 1) Make the job client only talk to RM (as an option) to get the job status. 2) Limit the range of ports AM can listen on. Option 2) may not be favorable as there is no direct OS API to find a free port.

#### jira\_issues\_comments:

1. bq. 1) Make the job client only talk to RM (as an option) to get the job status. I'll make that "only talk to RM and JobHistoryServer".
2. I'd think this is a blocker for 0.23.
3. This proposal, as stated, breaks the separation of concerns. RM should not be aware of MR jobs. Proper wrappers need to be provided to keep RM agnostic of jobs. Thus, AM -> RM should provide a K-V status message to RM. The RM client (wrapped by JC) should allow fetching such K-V status specifying a key (which wraps jobid), and the MR-specific portion of JC should interpret the value as a job status. In my understanding of the current code, I do not see such abstractions in RM.
4. This won't break abstractions. Most of this should be JobClient magic, to figure out the 'Application' status from RM and then the corresponding 'Job' status from history-server if it is completed. We need to simply short-circuit a trip to the AM in this special mode because of the security issues. RM doesn't have any information about jobs and will never have any.
5. Vinod, Which method in RMAppAttempt will be used for reporting JobStatus ? I believe, one will have to get the AM container from the RM, and then inquire status on that Container, right ? The ContainerStatus only contains ContainerId and State. There is not job-specific information there. This is where I was proposing a K-V query style interface. (The reason it needs to be K-V style, is if/when the MR AM (i.e. JT) evolves into running multiple MR jobs (or a job chain as in JobControl), it will come it handy to only get the status of a specific job instead of an entire chain.)
6. bq. I believe, one will have to get the AM container from the RM, and then inquire status on that Container, right ? Err, no. It's very different. The proposal is to ask RM for App state via getApplicationReport and then to talk to JHS when the MR App is 'complete'. This doesn't mean that RM knows or cares about MR apps.
7. Arun, Thanks for the clarification on getting completed jobs' status. What happens in case of running jobs ? I do not see any method in ApplicationReport that can include the job status for a running job.
8. In case of running jobs, if this option is enabled, we should never talk to AM and instead just get progress from ApplicationReport to display it. Once the job completes, the job-client should get counters etc. from the MR JobHistoryServer. Makes sense?
9. Arun, sorry to keep harping on this, but with this approach, the current individual task completion status will never be conveyed to the job client, right ? And since ApplicationReport is part of the public API, there is no way to include MR-specific details (even as a blob) there. As a concrete case, I do not see a way for the RM to report individual map progress, and reduce progress, for example, if we stop opening the JT port for job client.
10. Milind, this option is only for jobclients who cannot talk to AM directly, they get to choose. By default we assume JobClient can talk to AM to get full status etc. As described in the description of the jira it's for a few special cases.
11. Got it. Thanks. (I know of a few dashboards that might break if non-gateway access to JT was completely disabled ;-)
12. It was a mistake to describe the solution in the title instead of the problem itself. Editing title for the same reason. I thought more about this and I am getting increasingly concerned about option (1) for two reasons: Special clients like oozie, and clients outside the grid (which are the ones who will see this issue in the presence of network ACLs) will be seriously impaired with respect to getting updated information while the job is running for e.g, they won't see progress information. Also all of them will need to be very clearly aware of this new API which is a regression of sorts. I am leaning towards option (2). Though I agree that the Oses don't have an api to bind to a particular range of ports, it can still be done by randomly generating ports and keep trying till we can successfully bind to one of them. I know it isn't aesthetic, I myself don't like it much, but I do know from my experience with HOD that it will work. At any rate, it is better than the broken experience caused by adding new API.
13. I agree with Vinod. Looking for community's opinion so that we can reach a consensus and resolve this.

14. Option 2 would be a problem for operations folks. A secure cluster deployment is getting more and more complicated (eg. the proxy). I think option 1 might be fine, and we can make improvements to it by letting the AM send progress update to RM (via a string) and let the client get the update from RM and use that only the flag is turned on (flag for no communication to AM's). What do you guys think?
15. Back and forth on this one. I went back and looked at an oozie console in action. Checked both running and completed jobs. Fortunately (and rightly), oozie restricts itself to the workflow level and doesn't peek into the mapreduce bits like progress, counters etc. It instead just points to the job's web-page. So, I think we are good if we just have a mode (a configuration that oozie can explicitly set) to circumvent communication with the not-reachable-due-to-ACLs MR ApplicationMasters. Sure, there can be use-cases beyond oozie that may hit this issue. They can probably make do with the web-proxy that we have in RM.
16. Great. Lets go with Option 1 then.
17. Unless I'm missing something here, a client not only talk with MR side, often it does HDFS operations; and for this it needs network access to all cluster nodes. Unless you force them to use something like Hoop. Also, in MR2 land, where are splits being calculated?, this requires HDFS access.
18. Attaching a preliminary patch to gauge if I am headed in the right space. Feedback will be welcomed.
19. bq. Unless I'm missing something here, a client not only talk with MR side, often it does HDFS operations; and for this it needs network access to all cluster nodes. Unless you force them to use something like Hoop. Also, in MR2 land, where are splits being calculated?, this requires HDFS access. I probably do not just have any depth of understanding here, but will access to HDFS not be an issue for the client in the kind of scenario this is attempting to address since what is blocking is the port ACL not being opened up?
20. @Alejandro, With HDFS, the datanode ports acls are usually openend up to allow folks writing to HDFS from outside the cluster. The datanode ports are fixed and is easy to open acls to that but the issue here is that AM comes up on a random port and its not possible to open acls because of that. Hope that answers the questions.
21. **body:** @Anupam, After a brief look at the patch, a few comments: - we need to be thorough with testing this. We should add test cases that a client never tries connecting to AM. Completes fine without errors when a job is finished and make sure what all counters and others we get on the completion of the job.  
**label:** test
22. Also on counters, - We need to make sure, we get the counters from jobhistory when the job is done.
23. Attaching revised patch with tests added. Kindly review.
24. -1 overall. Here are the results of testing the latest attachment  
[http://issues.apache.org/jira/secure/attachment/12505786/MAPREDUCE-3251-branch\\_0\\_23.patch](http://issues.apache.org/jira/secure/attachment/12505786/MAPREDUCE-3251-branch_0_23.patch)  
against trunk revision . +1 @author. The patch does not contain any @author tags. +1 tests included. The patch appears to include 4 new or modified tests. +1 javadoc. The javadoc tool did not generate any warning messages. +1 javac. The applied patch does not increase the total number of javac compiler warnings. -1 findbugs. The patch appears to introduce 12 new Findbugs (version 1.3.9) warnings. +1 release audit. The applied patch does not increase the total number of release audit warnings. +1 core tests. The patch passed unit tests in . +1 contrib tests. The patch passed contrib unit tests. Test results:  
<https://builds.apache.org/job/PreCommit-MAPREDUCE-Build/1380//testReport/> Findbugs warnings:  
<https://builds.apache.org/job/PreCommit-MAPREDUCE-Build/1380//artifact/trunk/hadoop-mapreduce-project/patchprocess/newPatchFindbugsWarningshadoop-mapreduce-examples.html> Console output:  
<https://builds.apache.org/job/PreCommit-MAPREDUCE-Build/1380//console> This message is automatically generated.
25. Anupam, Just took a look at the patch. Some questions/comments: 1. Can you please click on the box to grant license to Apache for code inclusion for the next patch? 2. The code that create the Log.info() string should probably be a method on its own: I mean: {noformat} LOG.info("AppId: " + application.getApplicationId() + " # reserved containers: " + application.getApplicationResourceUsageReport().getNumReservedContainers() + " # used containers: " + application.getApplicationResourceUsageReport().getNumUsedContainers() + " Needed resources (memory): " + application.getApplicationResourceUsageReport().getNeededResources().getMemory() + " Rese {noformat} 3. Does the test verify that we make sure we get the counters from jobhistory once the job is done?
26. Thanks Mahadev! I have incorporated your suggesstion to move the LOG.info into a separate method and verified that the test invokes the counters from the Job History server. Am attaching the revied patch granting licesnse this time :)

27. +1 overall. Here are the results of testing the latest attachment [http://issues.apache.org/jira/secure/attachment/12506153/MAPREDUCE-3251-branch\\_0\\_23.patch](http://issues.apache.org/jira/secure/attachment/12506153/MAPREDUCE-3251-branch_0_23.patch) against trunk revision . +1 @author. The patch does not contain any @author tags. +1 tests included. The patch appears to include 4 new or modified tests. +1 javadoc. The javadoc tool did not generate any warning messages. +1 javac. The applied patch does not increase the total number of javac compiler warnings. +1 findbugs. The patch does not introduce any new Findbugs (version 1.3.9) warnings. +1 release audit. The applied patch does not increase the total number of release audit warnings. +1 core tests. The patch passed unit tests in . +1 contrib tests. The patch passed contrib unit tests. Test results: <https://builds.apache.org/job/PreCommit-MAPREDUCE-Build/1392//testReport/> Console output: <https://builds.apache.org/job/PreCommit-MAPREDUCE-Build/1392//console> This message is automatically generated.
28. **body:** @Anupam, I am not sure I understand the patch correctly: {noformat}  
 logApplicationReportInfo(application); + LOG.info("Network ACL closed to AM for job " + jobId + + ". Redirecting to job history server."); + return checkAndGetHSPROXY(null, JobState.RUNNING);  
 {noformat} YOu are returning a jobhistory proxy when the switch is turned on? Why is that? Also, one more thing, we should document the property in yarn-default.xml with default as false.  
**label:** documentation
29. Cancelling the patch to update the config and description.
30. bq. YOu are returning a jobhistory proxy when the switch is turned on? Why is that? Thanks Mahadev! I am returning the jobhistory proxy because the attempt to connect to the AM happens inside the call to getProxy, which is expecting some proxy back. In case, the network ACL is disabled, this behavior of returning the JH proxy is identical to the case when the connection to the AM fails for some unknown reason. The JH proxy is returned by get Proxy() with the application state set as RUNNING so that the user can still get a meaningful response in this condition. {noformat} } catch (IOException e) { //possibly the AM has crashed //there may be some time before AM is restarted //keep retrying by getting the address from RM LOG.info("Could not connect to " + serviceAddr + ". Waiting for getting the latest AM address..."); ... .. application = rm.getApplicationReport(appId); if (application == null) { LOG.debug("Could not get Job info from RM for job " + jobId + ". Redirecting to job history server."); return checkAndGetHSPROXY(null, JobState.RUNNING); } } {noformat} If that doesn't make sense, what would be a good alternative to return?
31. Uploading patch with inclusion of config property in yarn-defaults.xml
32. +1 overall. Here are the results of testing the latest attachment [http://issues.apache.org/jira/secure/attachment/12506759/MAPREDUCE-3251-branch\\_0\\_23.patch](http://issues.apache.org/jira/secure/attachment/12506759/MAPREDUCE-3251-branch_0_23.patch) against trunk revision . +1 @author. The patch does not contain any @author tags. +1 tests included. The patch appears to include 4 new or modified tests. +1 javadoc. The javadoc tool did not generate any warning messages. +1 javac. The applied patch does not increase the total number of javac compiler warnings. +1 findbugs. The patch does not introduce any new Findbugs (version 1.3.9) warnings. +1 release audit. The applied patch does not increase the total number of release audit warnings. +1 core tests. The patch passed unit tests in . +1 contrib tests. The patch passed contrib unit tests. Test results: <https://builds.apache.org/job/PreCommit-MAPREDUCE-Build/1416//testReport/> Console output: <https://builds.apache.org/job/PreCommit-MAPREDUCE-Build/1416//console> This message is automatically generated.
33. **body:** +1 the patch looks good Anupam. One very minor nit: You are using tabs in one line in ClientServiceDelegate.java. Ill just reupload a new one with the change. {noformat}  
 if(!conf.getBoolean(YarnConfiguration.RM\_AM\_NETWORK\_ACL\_CLOSED, false)) {  
 UserGroupInformation newUgi = UserGroupInformation.createRemoteUser(  
 UserGroupInformation.getCurrentUser().getUserName()); {noformat} Other than that the patch is good to go. Thanks for your patience on this!  
**label:** code-design
34. Just fixed the tabs issue.
35. I just committed this. Thanks Anupam!
36. Integrated in Hadoop-Common-0.23-Commit #292 (See [<https://builds.apache.org/job/Hadoop-Common-0.23-Commit/292/>]) MAPREDUCE-3251. Network ACLs can prevent some clients to talk to MR ApplicationMaster. (Anupam Seth via mahadev) - Merging r1214662 from trunk. mahadev :  
<http://svn.apache.org/viewcvcs.cgi/?root=Apache-SVN&view=rev&rev=1214664> Files : \*  
 /hadoop/common/branches/branch-0.23/hadoop-mapreduce-project/CHANGES.txt \*  
 /hadoop/common/branches/branch-0.23/hadoop-mapreduce-project/hadoop-mapreduce-client/hadoop-mapreduce-client-jobclient/src/main/java/org/apache/hadoop/mapred/ClientServiceDelegate.java \*  
 /hadoop/common/branches/branch-0.23/hadoop-mapreduce-project/hadoop-mapreduce-client/hadoop-

- mapreduce-client-jobclient/src/test/java/org/apache/hadoop/mapred/TestClientServiceDelegate.java \*  
/hadoop/common/branches/branch-0.23/hadoop-mapreduce-project/hadoop-yarn/hadoop-yarn-common/src/main/java/org/apache/hadoop/yarn/conf/YarnConfiguration.java \*  
/hadoop/common/branches/branch-0.23/hadoop-mapreduce-project/hadoop-yarn/hadoop-yarn-server/hadoop-yarn-server-common/src/main/resources/yarn-default.xml
37. Integrated in Hadoop-Hdfs-0.23-Commit #282 (See [<https://builds.apache.org/job/Hadoop-Hdfs-0.23-Commit/282/>]) MAPREDUCE-3251. Network ACLs can prevent some clients to talk to MR ApplicationMaster. (Anupam Seth via mahadev) - Merging r1214662 from trunk. mahadev : <http://svn.apache.org/viewcvcs.cgi/?root=Apache-SVN&view=rev&rev=1214664> Files : \*  
/hadoop/common/branches/branch-0.23/hadoop-mapreduce-project/CHANGES.txt \*  
/hadoop/common/branches/branch-0.23/hadoop-mapreduce-project/hadoop-mapreduce-client/hadoop-mapreduce-client-jobclient/src/main/java/org/apache/hadoop/mapred/ClientServiceDelegate.java \*  
/hadoop/common/branches/branch-0.23/hadoop-mapreduce-project/hadoop-mapreduce-client/hadoop-mapreduce-client-jobclient/src/test/java/org/apache/hadoop/mapred/TestClientServiceDelegate.java \*  
/hadoop/common/branches/branch-0.23/hadoop-mapreduce-project/hadoop-yarn/hadoop-yarn-common/src/main/java/org/apache/hadoop/yarn/conf/YarnConfiguration.java \*  
/hadoop/common/branches/branch-0.23/hadoop-mapreduce-project/hadoop-yarn/hadoop-yarn-server/hadoop-yarn-server-common/src/main/resources/yarn-default.xml
38. Integrated in Hadoop-Mapreduce-0.23-Commit #304 (See [<https://builds.apache.org/job/Hadoop-Mapreduce-0.23-Commit/304/>]) MAPREDUCE-3251. Network ACLs can prevent some clients to talk to MR ApplicationMaster. (Anupam Seth via mahadev) - Merging r1214662 from trunk. mahadev : <http://svn.apache.org/viewcvcs.cgi/?root=Apache-SVN&view=rev&rev=1214664> Files : \*  
/hadoop/common/branches/branch-0.23/hadoop-mapreduce-project/CHANGES.txt \*  
/hadoop/common/branches/branch-0.23/hadoop-mapreduce-project/hadoop-mapreduce-client/hadoop-mapreduce-client-jobclient/src/main/java/org/apache/hadoop/mapred/ClientServiceDelegate.java \*  
/hadoop/common/branches/branch-0.23/hadoop-mapreduce-project/hadoop-mapreduce-client/hadoop-mapreduce-client-jobclient/src/test/java/org/apache/hadoop/mapred/TestClientServiceDelegate.java \*  
/hadoop/common/branches/branch-0.23/hadoop-mapreduce-project/hadoop-yarn/hadoop-yarn-common/src/main/java/org/apache/hadoop/yarn/conf/YarnConfiguration.java \*  
/hadoop/common/branches/branch-0.23/hadoop-mapreduce-project/hadoop-yarn/hadoop-yarn-server/hadoop-yarn-server-common/src/main/resources/yarn-default.xml
39. Integrated in Hadoop-Mapreduce-trunk-Commit #1460 (See [<https://builds.apache.org/job/Hadoop-Mapreduce-trunk-Commit/1460/>]) MAPREDUCE-3251. Network ACLs can prevent some clients to talk to MR ApplicationMaster (Anupam Seth via mahadev) mahadev : <http://svn.apache.org/viewcvcs.cgi/?root=Apache-SVN&view=rev&rev=1214662> Files : \* /hadoop/common/trunk/hadoop-mapreduce-project/CHANGES.txt \* /hadoop/common/trunk/hadoop-mapreduce-project/hadoop-mapreduce-client/hadoop-mapreduce-client-jobclient/src/main/java/org/apache/hadoop/mapred/ClientServiceDelegate.java \*  
/hadoop/common/trunk/hadoop-mapreduce-project/hadoop-mapreduce-client/hadoop-mapreduce-client-jobclient/src/test/java/org/apache/hadoop/mapred/TestClientServiceDelegate.java \*  
/hadoop/common/trunk/hadoop-mapreduce-project/hadoop-yarn/hadoop-yarn-common/src/main/java/org/apache/hadoop/yarn/conf/YarnConfiguration.java \*  
/hadoop/common/trunk/hadoop-mapreduce-project/hadoop-yarn/hadoop-yarn-server/hadoop-yarn-server-common/src/main/resources/yarn-default.xml
40. Integrated in Hadoop-Common-trunk-Commit #1436 (See [<https://builds.apache.org/job/Hadoop-Common-trunk-Commit/1436/>]) MAPREDUCE-3251. Network ACLs can prevent some clients to talk to MR ApplicationMaster (Anupam Seth via mahadev) mahadev : <http://svn.apache.org/viewcvcs.cgi/?root=Apache-SVN&view=rev&rev=1214662> Files : \* /hadoop/common/trunk/hadoop-mapreduce-project/CHANGES.txt \* /hadoop/common/trunk/hadoop-mapreduce-project/hadoop-mapreduce-client/hadoop-mapreduce-client-jobclient/src/main/java/org/apache/hadoop/mapred/ClientServiceDelegate.java \*  
/hadoop/common/trunk/hadoop-mapreduce-project/hadoop-mapreduce-client/hadoop-mapreduce-client-jobclient/src/test/java/org/apache/hadoop/mapred/TestClientServiceDelegate.java \*  
/hadoop/common/trunk/hadoop-mapreduce-project/hadoop-yarn/hadoop-yarn-common/src/main/java/org/apache/hadoop/yarn/conf/YarnConfiguration.java \*  
/hadoop/common/trunk/hadoop-mapreduce-project/hadoop-yarn/hadoop-yarn-server/hadoop-yarn-server-common/src/main/resources/yarn-default.xml
41. Integrated in Hadoop-Hdfs-trunk-Commit #1509 (See [<https://builds.apache.org/job/Hadoop-Hdfs-trunk-Commit/1509/>]) MAPREDUCE-3251. Network ACLs can prevent some clients to talk to MR

- ApplicationMaster (Anupam Seth via mahadev) mahadev : <http://svn.apache.org/viewcvcs.cgi/?root=Apache-SVN&view=rev&rev=1214662> Files : \* /hadoop/common/trunk/hadoop-mapreduce-project/CHANGES.txt \* /hadoop/common/trunk/hadoop-mapreduce-project/hadoop-mapreduce-client/hadoop-mapreduce-client-jobclient/src/main/java/org/apache/hadoop/mapred/ClientServiceDelegate.java \* /hadoop/common/trunk/hadoop-mapreduce-project/hadoop-mapreduce-client/hadoop-mapreduce-client-jobclient/src/test/java/org/apache/hadoop/mapred/TestClientServiceDelegate.java \* /hadoop/common/trunk/hadoop-mapreduce-project/hadoop-yarn/hadoop-yarn-common/src/main/java/org/apache/hadoop/yarn/conf/YarnConfiguration.java \* /hadoop/common/trunk/hadoop-mapreduce-project/hadoop-yarn/hadoop-yarn-server/hadoop-yarn-server-common/src/main/resources/yarn-default.xml
42. Integrated in Hadoop-Mapreduce-0.23-Build #128 (See [<https://builds.apache.org/job/Hadoop-Mapreduce-0.23-Build/128/>]) MAPREDUCE-3251. Network ACLs can prevent some clients to talk to MR ApplicationMaster. (Anupam Seth via mahadev) - Merging r1214662 from trunk. mahadev : <http://svn.apache.org/viewcvcs.cgi/?root=Apache-SVN&view=rev&rev=1214664> Files : \* /hadoop/common/branches/branch-0.23/hadoop-mapreduce-project/CHANGES.txt \* /hadoop/common/branches/branch-0.23/hadoop-mapreduce-project/hadoop-mapreduce-client/hadoop-mapreduce-client-jobclient/src/main/java/org/apache/hadoop/mapred/ClientServiceDelegate.java \* /hadoop/common/branches/branch-0.23/hadoop-mapreduce-project/hadoop-mapreduce-client/hadoop-mapreduce-client-jobclient/src/test/java/org/apache/hadoop/mapred/TestClientServiceDelegate.java \* /hadoop/common/branches/branch-0.23/hadoop-mapreduce-project/hadoop-yarn/hadoop-yarn-common/src/main/java/org/apache/hadoop/yarn/conf/YarnConfiguration.java \* /hadoop/common/branches/branch-0.23/hadoop-mapreduce-project/hadoop-yarn/hadoop-yarn-server/hadoop-yarn-server-common/src/main/resources/yarn-default.xml
43. Integrated in Hadoop-Hdfs-0.23-Build #108 (See [<https://builds.apache.org/job/Hadoop-Hdfs-0.23-Build/108/>]) MAPREDUCE-3251. Network ACLs can prevent some clients to talk to MR ApplicationMaster. (Anupam Seth via mahadev) - Merging r1214662 from trunk. mahadev : <http://svn.apache.org/viewcvcs.cgi/?root=Apache-SVN&view=rev&rev=1214664> Files : \* /hadoop/common/branches/branch-0.23/hadoop-mapreduce-project/CHANGES.txt \* /hadoop/common/branches/branch-0.23/hadoop-mapreduce-project/hadoop-mapreduce-client/hadoop-mapreduce-client-jobclient/src/main/java/org/apache/hadoop/mapred/ClientServiceDelegate.java \* /hadoop/common/branches/branch-0.23/hadoop-mapreduce-project/hadoop-mapreduce-client/hadoop-mapreduce-client-jobclient/src/test/java/org/apache/hadoop/mapred/TestClientServiceDelegate.java \* /hadoop/common/branches/branch-0.23/hadoop-mapreduce-project/hadoop-yarn/hadoop-yarn-common/src/main/java/org/apache/hadoop/yarn/conf/YarnConfiguration.java \* /hadoop/common/branches/branch-0.23/hadoop-mapreduce-project/hadoop-yarn/hadoop-yarn-server/hadoop-yarn-server-common/src/main/resources/yarn-default.xml
44. Integrated in Hadoop-Mapreduce-trunk #928 (See [<https://builds.apache.org/job/Hadoop-Mapreduce-trunk/928/>]) MAPREDUCE-3251. Network ACLs can prevent some clients to talk to MR ApplicationMaster (Anupam Seth via mahadev) mahadev : <http://svn.apache.org/viewcvcs.cgi/?root=Apache-SVN&view=rev&rev=1214662> Files : \* /hadoop/common/trunk/hadoop-mapreduce-project/CHANGES.txt \* /hadoop/common/trunk/hadoop-mapreduce-project/hadoop-mapreduce-client/hadoop-mapreduce-client-jobclient/src/main/java/org/apache/hadoop/mapred/ClientServiceDelegate.java \* /hadoop/common/trunk/hadoop-mapreduce-project/hadoop-mapreduce-client/hadoop-mapreduce-client-jobclient/src/test/java/org/apache/hadoop/mapred/TestClientServiceDelegate.java \* /hadoop/common/trunk/hadoop-mapreduce-project/hadoop-yarn/hadoop-yarn-common/src/main/java/org/apache/hadoop/yarn/conf/YarnConfiguration.java \* /hadoop/common/trunk/hadoop-mapreduce-project/hadoop-yarn/hadoop-yarn-server/hadoop-yarn-server-common/src/main/resources/yarn-default.xml
45. I think this patch is broken. While the job is running, the client will keep going to the HistoryServer which will keep throwing exception saying that it does not know about this job. I am reopening this ticket. Anupam, did you do a real cluster-test or an integration test? May be I am missing something from the code and the patch is working accidentally?
46. @vinod, Good catch. You are right the patch needs more work. I am against reverting patches unless they are really broken. We can continue on this jira itself. @Anupam, here is what we need to do: - when you find that a job is running, you need to return fake proxy that just returns fake stubs on api calls. - on a job completion, you will have to make sure that if its completed, you connect to history server and get all the

counters and others from it, on failed job (AM crash), you won't be able to get anything from history server, so you'll have to return a job failed error on that.

47. **body:** bq. Anupam, did you do a real cluster-test or an integration test? @Vinod, yes I did. Here is console output upon disabling the ACL and running a word count job. I think I see the intent of what you are saying, and it will probably definitely be cleaner, but for some reason, it isn't as broken I think. @Mahadev, I will upload a new patch with the suggestions you have outlined in pursuance of the above comments. 11/12/15 20:41:32 WARN conf.Configuration: fs.default.name is deprecated. Instead, use fs.defaultFS 11/12/15 20:41:32 WARN conf.Configuration: mapred.used.genericoptionsparser is deprecated. Instead, use mapreduce.client.genericoptionsparser.used 11/12/15 20:41:32 INFO input.FileInputFormat: Total input paths to process : 1 11/12/15 20:41:32 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable 11/12/15 20:41:32 WARN snappy.LoadSnappy: Snappy native library not loaded 11/12/15 20:41:32 INFO mapreduce.JobSubmitter: number of splits:1 11/12/15 20:41:33 INFO mapred.ResourceMgrDelegate: Submitted application application\_1323981651676\_0001 to ResourceManager at <hostname>/98.139.92.65:8040 11/12/15 20:41:33 INFO mapreduce.Job: Running job: job\_1323981651676\_0001 11/12/15 20:41:42 INFO mapred.ClientServiceDelegate: AppId: application\_1323981651676\_0001 # reserved containers: 0 # used containers: 1 Needed resources (memory): 2048 Reserved resources (memory): 0 Used resources (memory): 2048 Diagnostics: Start time: 1323981693246 Finish time: 0 Host: <hostname> Name: word count Orig. tracking url: <hostname>:50256 Queue: default RPC port: 55191 Tracking url: <hostname>:8088/proxy/application\_1323981651676\_0001/ User: <user> Client token: null Final appl. status: UNDEFINED Yarn appl. state: RUNNING .... 11/12/15 20:41:56 INFO mapred.ClientServiceDelegate: Network ACL closed to AM for job job\_1323981651676\_0001. Redirecting to job history server. 11/12/15 20:41:56 WARN mapred.ClientServiceDelegate: Job History Server is not configured or job information not yet available on History Server. 11/12/15 20:41:56 INFO mapred.ClientServiceDelegate: AppId: application\_1323981651676\_0001 # reserved containers: 0 # used containers: 1 Needed resources (memory): 2048 Reserved resources (memory): 0 Used resources (memory): 2048 Diagnostics: Start time: 1323981693246 Finish time: 0 Host: <hostname> Name: word count Orig. tracking url: <hostname>:50256 Queue: default RPC port: 55191 Tracking url: <hostname>:8088/proxy/application\_1323981651676\_0001/ User: <user> Client token: null Final appl. status: UNDEFINED Yarn appl. state: RUNNING 11/12/15 20:41:56 INFO mapred.ClientServiceDelegate: Network ACL closed to AM for job job\_1323981651676\_0001. Redirecting to job history server. 11/12/15 20:41:56 WARN mapred.ClientServiceDelegate: Job History Server is not configured or job information not yet available on History Server. 11/12/15 20:41:57 INFO mapred.ClientServiceDelegate: Application state is completed. FinalApplicationStatus=SUCCEEDED. Redirecting to job history server 11/12/15 20:41:57 WARN mapred.ClientServiceDelegate: Job History Server is not configured or job information not yet available on History Server. 11/12/15 20:41:57 INFO mapreduce.Job: Job job\_1323981651676\_0001 completed successfully 11/12/15 20:41:57 INFO mapreduce.Job: Counters: 0

**label:** code-design

48. Thanks Mahadev, Vinod. Uploading fix for the problem.
49. -1 overall. Here are the results of testing the latest attachment [http://issues.apache.org/jira/secure/attachment/12507760/MAPREDUCE-3251-branch\\_0\\_23\\_incremental\\_fix.patch](http://issues.apache.org/jira/secure/attachment/12507760/MAPREDUCE-3251-branch_0_23_incremental_fix.patch) against trunk revision . +1 @author. The patch does not contain any @author tags. -1 tests included. The patch doesn't appear to include any new or modified tests. Please justify why no new tests are needed for this patch. Also please list what manual steps were performed to verify this patch. +1 javadoc. The javadoc tool did not generate any warning messages. +1 javac. The applied patch does not increase the total number of javac compiler warnings. +1 eclipse:eclipse. The patch built with eclipse:eclipse. +1 findbugs. The patch does not introduce any new Findbugs (version 1.3.9) warnings. +1 release audit. The applied patch does not increase the total number of release audit warnings. +1 core tests. The patch passed unit tests in . +1 contrib tests. The patch passed contrib unit tests. Test results: <https://builds.apache.org/job/PreCommit-MAPREDUCE-Build/1472//testReport/> Console output: <https://builds.apache.org/job/PreCommit-MAPREDUCE-Build/1472//console> This message is automatically generated.
50. This looks better. - At least for now, the configuration is a MapReduce-only flag and definitely not related to ResourceManager. Let's rename it as {{mapreduce.job.am-access-disabled}} and move it to {{MRJobConfig}}. - Not sure why logApplicationReportInfo() is needed. Let's drop this unless you did it explicitly for some reason. - Correct the log statement "Network ACL closed to AM for job " + jobId + ". Redirecting to job history server." We aren't redirecting to the history server. - Can you add a new test in

- {{TestClientServiceDelegate}}? None of the tests which run in the access-disabled mode do not explicitly test the current code. We need something like this: -- Client goes to RM, gets running state -- Tries to create a proxy, but doesn't reach the AM even though AM is alive, while the job is running -- Keeps doing the above till the job completes -- on job-completion, the client goes to the history-server.
51. Thanks Vinod. I have addressed all your comments in the new incremental patch.
52. **body:** The patch looks better now. +1 overall. There is excessive logging. I am fixing it and reuploading the patch myselfes.  
**label:** code-design
53. +1 overall. Here are the results of testing the latest attachment  
[http://issues.apache.org/jira/secure/attachment/12510096/MAPREDUCE-3251-branch\\_0\\_23\\_incremental\\_fix\\_2.patch](http://issues.apache.org/jira/secure/attachment/12510096/MAPREDUCE-3251-branch_0_23_incremental_fix_2.patch) against trunk revision . +1 @author. The patch does not contain any @author tags. +1 tests included. The patch appears to include 6 new or modified tests. +1 javadoc. The javadoc tool did not generate any warning messages. +1 javac. The applied patch does not increase the total number of javac compiler warnings. +1 eclipse:eclipse. The patch built with eclipse:eclipse. +1 findbugs. The patch does not introduce any new Findbugs (version 1.3.9) warnings. +1 release audit. The applied patch does not increase the total number of release audit warnings. +1 core tests. The patch passed unit tests in . +1 contrib tests. The patch passed contrib unit tests. Test results: <https://builds.apache.org/job/PreCommit-MAPREDUCE-Build/1583/testReport/> Console output: <https://builds.apache.org/job/PreCommit-MAPREDUCE-Build/1583/console> This message is automatically generated.
54. **body:** Updated patch which removes excessive logging when ACLs to AM disable access.  
**label:** code-design
55. +1 overall. Here are the results of testing the latest attachment  
<http://issues.apache.org/jira/secure/attachment/12510112/MAPREDUCE-3251-20120110.txt> against trunk revision . +1 @author. The patch does not contain any @author tags. +1 tests included. The patch appears to include 6 new or modified tests. +1 javadoc. The javadoc tool did not generate any warning messages. +1 javac. The applied patch does not increase the total number of javac compiler warnings. +1 eclipse:eclipse. The patch built with eclipse:eclipse. +1 findbugs. The patch does not introduce any new Findbugs (version 1.3.9) warnings. +1 release audit. The applied patch does not increase the total number of release audit warnings. +1 core tests. The patch passed unit tests in . +1 contrib tests. The patch passed contrib unit tests. Test results: <https://builds.apache.org/job/PreCommit-MAPREDUCE-Build/1585/testReport/> Console output: <https://builds.apache.org/job/PreCommit-MAPREDUCE-Build/1585/console> This message is automatically generated.
56. +1 for the patch. Ill go ahead and commit it.
57. Thanks for the review Mahadev, but never mind the commit, I was already half way through ;) I committed this to trunk and branch-0.23. Thanks Anupam! That was a long drawn affair. \*smile\*
58. Integrated in Hadoop-Hdfs-0.23-Commit #352 (See [<https://builds.apache.org/job/Hadoop-Hdfs-0.23-Commit/352/>]) MAPREDUCE-3251. Network ACLs can prevent some clients to talk to MR AM. Improved the earlier patch to not to JobHistoryServer repeatedly. Contributed by Anupam Seth. svn merge --ignore-ancestry -c 1229787 ../trunk/ vinodkv : <http://svn.apache.org/viewcvs.cgi/?root=Apache-SVN&view=rev&rev=1229789> Files : \* /hadoop/common/branches/branch-0.23/hadoop-mapreduce-project/CHANGES.txt \* /hadoop/common/branches/branch-0.23/hadoop-mapreduce-project/hadoop-mapreduce-client/hadoop-mapreduce-client-core/src/main/java/org/apache/hadoop/mapreduce/MRJobConfig.java \* /hadoop/common/branches/branch-0.23/hadoop-mapreduce-project/hadoop-mapreduce-client/hadoop-mapreduce-client-jobclient/src/main/java/org/apache/hadoop/mapred/ClientServiceDelegate.java \* /hadoop/common/branches/branch-0.23/hadoop-mapreduce-project/hadoop-mapreduce-client/hadoop-mapreduce-client-jobclient/src/test/java/org/apache/hadoop/mapred/TestClientServiceDelegate.java \* /hadoop/common/branches/branch-0.23/hadoop-mapreduce-project/hadoop-yarn/hadoop-yarn-common/src/main/java/org/apache/hadoop/yarn/conf/YarnConfiguration.java \* /hadoop/common/branches/branch-0.23/hadoop-mapreduce-project/hadoop-yarn/hadoop-yarn-server/hadoop-yarn-server-common/src/main/resources/yarn-default.xml \* /hadoop/common/branches/branch-0.23/hadoop-mapreduce-project/src/java/mapred-default.xml
59. Integrated in Hadoop-Common-trunk-Commit #1522 (See [<https://builds.apache.org/job/Hadoop-Common-trunk-Commit/1522/>]) MAPREDUCE-3251. Network ACLs can prevent some clients to talk to MR AM. Improved the earlier patch to not to JobHistoryServer repeatedly. Contributed by Anupam Seth. vinodkv : <http://svn.apache.org/viewcvs.cgi/?root=Apache-SVN&view=rev&rev=1229787> Files : \* /hadoop/common/trunk/hadoop-mapreduce-project/CHANGES.txt \* /hadoop/common/trunk/hadoop-mapreduce-project/hadoop-mapreduce-client/hadoop-mapreduce-client-



- core/src/main/java/org/apache/hadoop/mapreduce/MRJobConfig.java \* /hadoop/common/trunk/hadoop-mapreduce-project/hadoop-mapreduce-client/hadoop-mapreduce-client-jobclient/src/main/java/org/apache/hadoop/mapred/ClientServiceDelegate.java \*  
/hadoop/common/trunk/hadoop-mapreduce-project/hadoop-mapreduce-client/hadoop-mapreduce-client-jobclient/src/test/java/org/apache/hadoop/mapred/TestClientServiceDelegate.java \*  
/hadoop/common/trunk/hadoop-mapreduce-project/hadoop-yarn/hadoop-yarn-common/src/main/java/org/apache/hadoop/yarn/conf/YarnConfiguration.java \*  
/hadoop/common/trunk/hadoop-mapreduce-project/hadoop-yarn/hadoop-yarn-server/hadoop-yarn-server-common/src/main/resources/yarn-default.xml \* /hadoop/common/trunk/hadoop-mapreduce-project/src/java/mapred-default.xml
60. Integrated in Hadoop-Common-0.23-Commit #362 (See [<https://builds.apache.org/job/Hadoop-Common-0.23-Commit/362/>]) MAPREDUCE-3251. Network ACLs can prevent some clients to talk to MR AM. Improved the earlier patch to not to JobHistoryServer repeatedly. Contributed by Anupam Seth. svn merge --ignore-ancestry -c 1229787 ../trunk/ vinodkv : <http://svn.apache.org/viewcvcs.cgi/?root=Apache-SVN&view=rev&rev=1229789> Files : \* /hadoop/common/branches/branch-0.23/hadoop-mapreduce-project/CHANGES.txt \* /hadoop/common/branches/branch-0.23/hadoop-mapreduce-project/hadoop-mapreduce-client/hadoop-mapreduce-client-core/src/main/java/org/apache/hadoop/mapreduce/MRJobConfig.java \*  
/hadoop/common/branches/branch-0.23/hadoop-mapreduce-project/hadoop-mapreduce-client/hadoop-mapreduce-client-jobclient/src/main/java/org/apache/hadoop/mapred/ClientServiceDelegate.java \*  
/hadoop/common/branches/branch-0.23/hadoop-mapreduce-project/hadoop-mapreduce-client/hadoop-mapreduce-client-jobclient/src/test/java/org/apache/hadoop/mapred/TestClientServiceDelegate.java \*  
/hadoop/common/branches/branch-0.23/hadoop-mapreduce-project/hadoop-yarn/hadoop-yarn-common/src/main/java/org/apache/hadoop/yarn/conf/YarnConfiguration.java \*  
/hadoop/common/branches/branch-0.23/hadoop-mapreduce-project/hadoop-yarn/hadoop-yarn-server/hadoop-yarn-server-common/src/main/resources/yarn-default.xml \*  
/hadoop/common/branches/branch-0.23/hadoop-mapreduce-project/src/java/mapred-default.xml
61. Integrated in Hadoop-Hdfs-trunk-Commit #1595 (See [<https://builds.apache.org/job/Hadoop-Hdfs-trunk-Commit/1595/>]) MAPREDUCE-3251. Network ACLs can prevent some clients to talk to MR AM. Improved the earlier patch to not to JobHistoryServer repeatedly. Contributed by Anupam Seth. vinodkv : <http://svn.apache.org/viewcvcs.cgi/?root=Apache-SVN&view=rev&rev=1229787> Files : \*  
/hadoop/common/trunk/hadoop-mapreduce-project/CHANGES.txt \* /hadoop/common/trunk/hadoop-mapreduce-project/hadoop-mapreduce-client/hadoop-mapreduce-client-core/src/main/java/org/apache/hadoop/mapreduce/MRJobConfig.java \* /hadoop/common/trunk/hadoop-mapreduce-project/hadoop-mapreduce-client/hadoop-mapreduce-client-jobclient/src/main/java/org/apache/hadoop/mapred/ClientServiceDelegate.java \*  
/hadoop/common/trunk/hadoop-mapreduce-project/hadoop-mapreduce-client/hadoop-mapreduce-client-jobclient/src/test/java/org/apache/hadoop/mapred/TestClientServiceDelegate.java \*  
/hadoop/common/trunk/hadoop-mapreduce-project/hadoop-yarn/hadoop-yarn-common/src/main/java/org/apache/hadoop/yarn/conf/YarnConfiguration.java \*  
/hadoop/common/trunk/hadoop-mapreduce-project/hadoop-yarn/hadoop-yarn-server/hadoop-yarn-server-common/src/main/resources/yarn-default.xml \* /hadoop/common/trunk/hadoop-mapreduce-project/src/java/mapred-default.xml
62. Integrated in Hadoop-Mapreduce-0.23-Commit #373 (See [<https://builds.apache.org/job/Hadoop-Mapreduce-0.23-Commit/373/>]) MAPREDUCE-3251. Network ACLs can prevent some clients to talk to MR AM. Improved the earlier patch to not to JobHistoryServer repeatedly. Contributed by Anupam Seth. svn merge --ignore-ancestry -c 1229787 ../trunk/ vinodkv : <http://svn.apache.org/viewcvcs.cgi/?root=Apache-SVN&view=rev&rev=1229789> Files : \* /hadoop/common/branches/branch-0.23/hadoop-mapreduce-project/CHANGES.txt \* /hadoop/common/branches/branch-0.23/hadoop-mapreduce-project/hadoop-mapreduce-client/hadoop-mapreduce-client-core/src/main/java/org/apache/hadoop/mapreduce/MRJobConfig.java \*  
/hadoop/common/branches/branch-0.23/hadoop-mapreduce-project/hadoop-mapreduce-client/hadoop-mapreduce-client-jobclient/src/main/java/org/apache/hadoop/mapred/ClientServiceDelegate.java \*  
/hadoop/common/branches/branch-0.23/hadoop-mapreduce-project/hadoop-mapreduce-client/hadoop-mapreduce-client-jobclient/src/test/java/org/apache/hadoop/mapred/TestClientServiceDelegate.java \*  
/hadoop/common/branches/branch-0.23/hadoop-mapreduce-project/hadoop-yarn/hadoop-yarn-common/src/main/java/org/apache/hadoop/yarn/conf/YarnConfiguration.java \*  
/hadoop/common/branches/branch-0.23/hadoop-mapreduce-project/hadoop-yarn/hadoop-yarn-

- server/hadoop-yarn-server-common/src/main/resources/yarn-default.xml \*
- /hadoop/common/branches/branch-0.23/hadoop-mapreduce-project/src/java/mapred-default.xml
63. Integrated in Hadoop-Mapreduce-trunk-Commit #1541 (See [<https://builds.apache.org/job/Hadoop-Mapreduce-trunk-Commit/1541/>]) MAPREDUCE-3251. Network ACLs can prevent some clients to talk to MR AM. Improved the earlier patch to not to JobHistoryServer repeatedly. Contributed by Anupam Seth. vinodkv : <http://svn.apache.org/viewcvcs.cgi/?root=Apache-SVN&view=rev&rev=1229787> Files : \*
- /hadoop/common/trunk/hadoop-mapreduce-project/CHANGES.txt \* /hadoop/common/trunk/hadoop-mapreduce-project/hadoop-mapreduce-client/hadoop-mapreduce-client-core/src/main/java/org/apache/hadoop/mapreduce/MRJobConfig.java \* /hadoop/common/trunk/hadoop-mapreduce-project/hadoop-mapreduce-client/hadoop-mapreduce-client-jobclient/src/main/java/org/apache/hadoop/mapred/ClientServiceDelegate.java \*
- /hadoop/common/trunk/hadoop-mapreduce-project/hadoop-mapreduce-client/hadoop-mapreduce-client-jobclient/src/test/java/org/apache/hadoop/mapred/TestClientServiceDelegate.java \*
- /hadoop/common/trunk/hadoop-mapreduce-project/hadoop-yarn/hadoop-yarn-common/src/main/java/org/apache/hadoop/yarn/conf/YarnConfiguration.java \*
- /hadoop/common/trunk/hadoop-mapreduce-project/hadoop-yarn/hadoop-yarn-server/hadoop-yarn-server-common/src/main/resources/yarn-default.xml \* /hadoop/common/trunk/hadoop-mapreduce-project/src/java/mapred-default.xml
64. Integrated in Hadoop-Hdfs-trunk #922 (See [<https://builds.apache.org/job/Hadoop-Hdfs-trunk/922/>]) MAPREDUCE-3251. Network ACLs can prevent some clients to talk to MR AM. Improved the earlier patch to not to JobHistoryServer repeatedly. Contributed by Anupam Seth. vinodkv : <http://svn.apache.org/viewcvcs.cgi/?root=Apache-SVN&view=rev&rev=1229787> Files : \*
- /hadoop/common/trunk/hadoop-mapreduce-project/CHANGES.txt \* /hadoop/common/trunk/hadoop-mapreduce-project/hadoop-mapreduce-client/hadoop-mapreduce-client-core/src/main/java/org/apache/hadoop/mapreduce/MRJobConfig.java \* /hadoop/common/trunk/hadoop-mapreduce-project/hadoop-mapreduce-client/hadoop-mapreduce-client-jobclient/src/main/java/org/apache/hadoop/mapred/ClientServiceDelegate.java \*
- /hadoop/common/trunk/hadoop-mapreduce-project/hadoop-mapreduce-client/hadoop-mapreduce-client-jobclient/src/test/java/org/apache/hadoop/mapred/TestClientServiceDelegate.java \*
- /hadoop/common/trunk/hadoop-mapreduce-project/hadoop-yarn/hadoop-yarn-common/src/main/java/org/apache/hadoop/yarn/conf/YarnConfiguration.java \*
- /hadoop/common/trunk/hadoop-mapreduce-project/hadoop-yarn/hadoop-yarn-server/hadoop-yarn-server-common/src/main/resources/yarn-default.xml \* /hadoop/common/trunk/hadoop-mapreduce-project/src/java/mapred-default.xml
65. Integrated in Hadoop-Hdfs-0.23-Build #135 (See [<https://builds.apache.org/job/Hadoop-Hdfs-0.23-Build/135/>]) MAPREDUCE-3251. Network ACLs can prevent some clients to talk to MR AM. Improved the earlier patch to not to JobHistoryServer repeatedly. Contributed by Anupam Seth. svn merge --ignore-ancestry -c 1229787 ../trunk/ vinodkv : <http://svn.apache.org/viewcvcs.cgi/?root=Apache-SVN&view=rev&rev=1229789> Files : \*
- /hadoop/common/branches/branch-0.23/hadoop-mapreduce-project/CHANGES.txt \* /hadoop/common/branches/branch-0.23/hadoop-mapreduce-project/hadoop-mapreduce-client/hadoop-mapreduce-client-core/src/main/java/org/apache/hadoop/mapreduce/MRJobConfig.java \*
- /hadoop/common/branches/branch-0.23/hadoop-mapreduce-project/hadoop-mapreduce-client/hadoop-mapreduce-client-jobclient/src/main/java/org/apache/hadoop/mapred/ClientServiceDelegate.java \*
- /hadoop/common/branches/branch-0.23/hadoop-mapreduce-project/hadoop-mapreduce-client/hadoop-mapreduce-client-jobclient/src/test/java/org/apache/hadoop/mapred/TestClientServiceDelegate.java \*
- /hadoop/common/branches/branch-0.23/hadoop-mapreduce-project/hadoop-yarn/hadoop-yarn-common/src/main/java/org/apache/hadoop/yarn/conf/YarnConfiguration.java \*
- /hadoop/common/branches/branch-0.23/hadoop-mapreduce-project/hadoop-yarn/hadoop-yarn-server/hadoop-yarn-server-common/src/main/resources/yarn-default.xml \*
- /hadoop/common/branches/branch-0.23/hadoop-mapreduce-project/src/java/mapred-default.xml
66. Integrated in Hadoop-Mapreduce-0.23-Build #157 (See [<https://builds.apache.org/job/Hadoop-Mapreduce-0.23-Build/157/>]) MAPREDUCE-3251. Network ACLs can prevent some clients to talk to MR AM. Improved the earlier patch to not to JobHistoryServer repeatedly. Contributed by Anupam Seth. svn merge --ignore-ancestry -c 1229787 ../trunk/ vinodkv : <http://svn.apache.org/viewcvcs.cgi/?root=Apache-SVN&view=rev&rev=1229789> Files : \*
- /hadoop/common/branches/branch-0.23/hadoop-mapreduce-project/CHANGES.txt \* /hadoop/common/branches/branch-0.23/hadoop-mapreduce-project/hadoop-mapreduce-client/hadoop-mapreduce-client-core/src/main/java/org/apache/hadoop/mapreduce/MRJobConfig.java \*

/hadoop/common/branches/branch-0.23/hadoop-mapreduce-project/hadoop-mapreduce-client/hadoop-mapreduce-client-jobclient/src/main/java/org/apache/hadoop/mapred/ClientServiceDelegate.java \*  
 /hadoop/common/branches/branch-0.23/hadoop-mapreduce-project/hadoop-mapreduce-client/hadoop-mapreduce-client-jobclient/src/test/java/org/apache/hadoop/mapred/TestClientServiceDelegate.java \*  
 /hadoop/common/branches/branch-0.23/hadoop-mapreduce-project/hadoop-yarn/hadoop-yarn-common/src/main/java/org/apache/hadoop/yarn/conf/YarnConfiguration.java \*  
 /hadoop/common/branches/branch-0.23/hadoop-mapreduce-project/hadoop-yarn/hadoop-yarn-server/hadoop-yarn-server-common/src/main/resources/yarn-default.xml \*  
 /hadoop/common/branches/branch-0.23/hadoop-mapreduce-project/src/java/mapred-default.xml  
 67. Integrated in Hadoop-Mapreduce-trunk #955 (See [<https://builds.apache.org/job/Hadoop-Mapreduce-trunk/955/>]) MAPREDUCE-3251. Network ACLs can prevent some clients to talk to MR AM. Improved the earlier patch to not to JobHistoryServer repeatedly. Contributed by Anupam Seth. vinodkv :  
<http://svn.apache.org/viewcvcs.cgi/?root=Apache-SVN&view=rev&rev=1229787> Files : \*  
 /hadoop/common/trunk/hadoop-mapreduce-project/CHANGES.txt \* /hadoop/common/trunk/hadoop-mapreduce-project/hadoop-mapreduce-client/hadoop-mapreduce-client-core/src/main/java/org/apache/hadoop/mapreduce/MRJobConfig.java \* /hadoop/common/trunk/hadoop-mapreduce-project/hadoop-mapreduce-client/hadoop-mapreduce-client-jobclient/src/main/java/org/apache/hadoop/mapred/ClientServiceDelegate.java \*  
 /hadoop/common/trunk/hadoop-mapreduce-project/hadoop-mapreduce-client/hadoop-mapreduce-client-jobclient/src/test/java/org/apache/hadoop/mapred/TestClientServiceDelegate.java \*  
 /hadoop/common/trunk/hadoop-mapreduce-project/hadoop-yarn/hadoop-yarn-common/src/main/java/org/apache/hadoop/yarn/conf/YarnConfiguration.java \*  
 /hadoop/common/trunk/hadoop-mapreduce-project/hadoop-yarn/hadoop-yarn-server/hadoop-yarn-server-common/src/main/resources/yarn-default.xml \* /hadoop/common/trunk/hadoop-mapreduce-project/src/java/mapred-default.xml