IEP-35 Monitoring & Profiling

Motivation

For now, Ignite has not full, fragmented monitoring API. Those APIs use different protocols, such as - JMX, Java API, SQL System views, text logs, etc.

From the administrator point of view, it's impossible to understand what is going on in a running cluster:

- Which user tasks are executed?
- What resources are used by each task?

It's very hard to export all existing metrics to some arbitrary monitoring system due to a variety of protocols.

The goal of IEP is to provide a way to answer 3 questions:

- **What is running inside the Ignite cluster?**
  - An administrator should be able to enlist each user object that was created(ran) inside a cluster via every monitoring interface we will support(JMX, SQL, CLI, etc).
  - An administrator should be able to identify the source of each user object via some ID or other user-provided info.

- **What is running slow?**
  - If some user code execution violates configured thresholds handler of such events should be executed. By default, the handler should print WARN log message with all available information about a slow piece of user provided code.

- **What will be running slow?**
  - We should provide a way to execute cluster profiling. Consider the following scenario:
    - Enable profiling mode.
    - Executes some arbitrary workload.
    - Collects profiling info.
    - Run some Ignite-provided tool that will create the Report contains statistics of workload. Examples of such tool are:
      - Oracle AWR
      - PostgreSQL pgBadger

Description:

- Motivation
- Description:
  - Phase1: What is running inside the Ignite cluster? + What is running slow?
  - Lists of Ignite objects/entities that should be listed in Phase 2
  - Internal Data Structures and Processes we should provide info for
- Risks and Assumptions
- Discussion Links
- Gap analysis
- Monitoring completely unavailable:
- Monitoring API available:
- Design Principles
- Reference Links
- Tickets
Phase 1: What is running inside the Ignite cluster? + What is running slow?

1. We should add some entities in Ignite:
   
   1. **MetricRegistry** - Ignite subsystem that provides some set of sensors and lists.
      a. Cache,
      b. Compute,
      c. ServiceGrid,
      d. etc.
   
   2. **Metric** - some named number with a well-defined algorithm to calculate the value in any given moments in time.

   ```java
   class Metric {
       String name; //EntryCount, MemoryAvailable, etc
       long value; //or double
       Collection<Tuple2<String, String>> labels; //hostName, cacheName, etc.
   }
   
   class LongMetric extends Metric {
       long ts; //timestamp of the last value update.
   }
   ```

   3. **SystemView** - some named list that contains info about Ignite objects. Examples: List of caches, Transactions list, List of nodes, List of running queries, Las N queries, etc...

   4. **MonitoringEvent** - generated when some user-defined code violates the threshold.

   ```java
   class MonitoringEvent {
       MonitoringEventType type; //Event type.
       T info; //Event info. Type of info differs for different type of events.
   }
   ```

2. **GridMetricManager, GridSystemViewManager**:

   1. **GridMetricManager** - should be able to store and query Ignite metrics.
   2. **GridSystemViewManager** - should be able to store and export SystemViews.

3. **Exporters**:

   Specific interfaces will be supported through exporters.
   Exporters should work only with a read-only version of **GridMetricManager** and don't rely on other knowledge about Ignite internals.
Example of exporters:

1. JMX
2. HTTP
3. SQL System View
4. Log
5. etc.

Lists of Ignite objects/entities that should be listed in Phase 2

1. A list of compute tasks:
   a. Closures
   b. Map-reduce jobs
   c. ComputeJob
   d. Scheduled tasks
2. Service grid:
   a. A list of services with deployment status
3. Caches
4. Cache groups
5. Cluster nodes
6. SQL objects
   a. Schemas
   b. Tables
   c. Views
   d. Tables columns
   e. Views columns
   f. Indexes
7. Queries:
   a. SQL
   b. Scan
   c. Text
   d. ContinuousQuery
8. IgniteCache#invoke
9. put, get, remove, replace, clear operations
10. Transactions with lock list
11. DataStreamers
12. Explicit locks(IgniteCache#lock)
13. DataStructures
   a. Queue
   b. Set
   c. AtomicLong
   d. AtomicReference
   e. CountDownLatch
   f. Sequence
   g. Semaphore
14. Message topics (IgniteMessaging)
15. Thin client connections.
16. Machine Learning - ???

Internal Data Structures and Processes we should provide info for

1. PME queue
2. Service exchange queue
3. Security events

Risks and Assumptions

Backward compatibility is in danger with these changes.

We should consider implementing this IEP as Ignite 3.

Discussion Links

Gap analysis

Current monitoring APIs availability:

Monitoring completely unavailable:

1. Compute Grid
   a. Some basic number available in ClusterMetrics(getMaximumActiveJobs, getCurrentActiveJobs, etc...)
2. Service Grid
3. Data streamers
4. Distributed Data Structures
5. Ignite messaging (Ignite#message)
6. 3-d party storage
7. ContinuousQuery
8. MVCC transactions
9. ML - What should be available?
10. Explicit locks

Monitoring API available:

1. Cache
   a. PDS + offheap memory
      i. Ignite#dataRegionMetrics
      ii. Ignite#dataStorageMetrics
      iii. Ignite#persistentStoreMetrics
   b. Queries
      i. IgniteCache#queryMetrics
      ii. IgniteCache#queryDetailMetrics
      iii. QueryHistoryMetrics
   c. IgniteCache#mxBean
   d. IgniteCache#localMxBean
2. SQL
   a. LOCAL_SQL_RUNNING QUERIES
   b. INDEXES
3. Transactions
   a. JMX - TransactionMetricsMxBean
   b. JMX - TransactionMXBean
4. ThinClients
   a. JMX - ClientProcessorMxBean
5. IoStatisticsManager, IoStatisticsHolder
6. GridJobMetricsProcessor
7. IgniteMBeansManager
8. IgniteSpiManagementMBean

Design Principles

1. Sensors should contain only raw values. No aggregation of numeric metrics on Ignite side. Min, max, avg and other functions are the matter of external monitoring system.
2. Every user task should have an ID or name provided by a user on start time that allows association between monitoring info and user code. User should be able to find his code reflected in monitoring.
3. Every user task should have an ID or name of "connectionID"("sessionID", "clientID") or similar. User should be able to know that a specific task was triggered by the specific connection(session, client).
4. No computation to get current values. We should change sensors and lists values when specific events occur. When some sensor queries we should only get its value from internal storage. No additional computation involved.
5. User should be able to enable/disable any Sensor group/List at runtime. Ignite should provide some administrator interface(s) to enable/disable each Sensor Group or List separately. No performance penalty for disabled sensors, lists.

Reference Links

https://docs.oracle.com/cd/E11882_01/server.112/e41573/autostat.htm#PFGRF027
https://github.com/darold/pgbadger
<table>
<thead>
<tr>
<th>Key</th>
<th>Summary</th>
<th>T</th>
<th>Created</th>
<th>Updated</th>
<th>Due</th>
<th>Assignee</th>
<th>Reporter</th>
<th>P</th>
<th>Status</th>
<th>Resolution</th>
</tr>
</thead>
<tbody>
<tr>
<td>IGNITE-15284</td>
<td>NPE when exporting metrics on deactivation</td>
<td>🔄</td>
<td>Aug 11, 2021</td>
<td>Sep 16, 2021</td>
<td>Vladimir Slestin</td>
<td>Nikolay Izhikov</td>
<td>🔄</td>
<td></td>
<td>RESOLVED</td>
<td>Not A Problem</td>
</tr>
<tr>
<td>IGNITE-14718</td>
<td>Extension: Performance Statistics. Update chartjs version to 3.2.1</td>
<td>✔️</td>
<td>May 14, 2021</td>
<td>Jul 02, 2021</td>
<td>Sergei Ryzhov</td>
<td>Sergei Ryzhov</td>
<td>🔄</td>
<td></td>
<td>PATCH AVAILABLE</td>
<td>Unresolved</td>
</tr>
<tr>
<td>IGNITE-14658</td>
<td>[IEP-35] SSL metrics</td>
<td>🔄</td>
<td>Apr 27, 2021</td>
<td>Jul 01, 2021</td>
<td>Mikhail Petrov</td>
<td>Nikolay Izhikov</td>
<td>🔄</td>
<td></td>
<td>RESOLVED</td>
<td>Fixed</td>
</tr>
<tr>
<td>IGNITE-14428</td>
<td>Formalize the names of the metrics included in the metric registry.</td>
<td>🔄</td>
<td>Mar 26, 2021</td>
<td>Jun 29, 2021</td>
<td>Mikhail Petrov</td>
<td>Mikhail Petrov</td>
<td>🔄</td>
<td></td>
<td>RESOLVED</td>
<td>Fixed</td>
</tr>
<tr>
<td>IGNITE-14368</td>
<td>System view for DataStructures</td>
<td>🔄</td>
<td>Mar 21, 2021</td>
<td>Sep 18, 2021</td>
<td>Nikolay Izhikov</td>
<td>Nikolay Izhikov</td>
<td>🔄</td>
<td></td>
<td>RESOLVED</td>
<td>Fixed</td>
</tr>
<tr>
<td>IGNITE-14255</td>
<td>Added the ability to rotate collecting performance statistics</td>
<td>🔄</td>
<td>Feb 28, 2021</td>
<td>Jul 02, 2021</td>
<td>Sergei Ryzhov</td>
<td>Sergei Ryzhov</td>
<td>🔄</td>
<td></td>
<td>RESOLVED</td>
<td>Fixed</td>
</tr>
<tr>
<td>IGNITE-13854</td>
<td>Add documentation for the cluster performance profiling tool</td>
<td>🔄</td>
<td>Dec 15, 2020</td>
<td>Jan 18, 2021</td>
<td>Amelchev Nikita</td>
<td>Amelchev Nikita</td>
<td>🔄</td>
<td></td>
<td>RESOLVED</td>
<td>Fixed</td>
</tr>
<tr>
<td>IGNITE-13726</td>
<td>Add an ability to view count of hot and cold pages in page memory</td>
<td>🔄</td>
<td>Nov 18, 2020</td>
<td>Nov 22, 2021</td>
<td>Aleksey Plekhanov</td>
<td>Aleksey Plekhanov</td>
<td>🔄</td>
<td></td>
<td>PATCH AVAILABLE</td>
<td>Unresolved</td>
</tr>
<tr>
<td>IGNITE-13458</td>
<td>RebalancingPartitionsTotal metrics</td>
<td>🔄</td>
<td>Sep 17, 2020</td>
<td>Nov 03, 2020</td>
<td>Stepaichev Maksim</td>
<td>Stepaichev Maksim</td>
<td>🔄</td>
<td></td>
<td>RESOLVED</td>
<td>Fixed</td>
</tr>
<tr>
<td>IGNITE-13409</td>
<td>System view for metastorage items</td>
<td>🔄</td>
<td>Sep 07, 2020</td>
<td>Nov 10, 2020</td>
<td>Nikolay Izhikov</td>
<td>Nikolay Izhikov</td>
<td>🔄</td>
<td></td>
<td>RESOLVED</td>
<td>Fixed</td>
</tr>
<tr>
<td>IGNITE-13408</td>
<td>System view for binary metadata</td>
<td>🔄</td>
<td>Sep 07, 2020</td>
<td>Mar 20, 2021</td>
<td>Nikolay Izhikov</td>
<td>Nikolay Izhikov</td>
<td>🔄</td>
<td></td>
<td>RESOLVED</td>
<td>Fixed</td>
</tr>
<tr>
<td>IGNITE-13354</td>
<td>Add ClusterMetrics to the new framework</td>
<td>🔄</td>
<td>Aug 12, 2020</td>
<td>Nov 10, 2020</td>
<td>Amelchev Nikita</td>
<td>Nikolay Izhikov</td>
<td>🔄</td>
<td></td>
<td>RESOLVED</td>
<td>Fixed</td>
</tr>
<tr>
<td>IGNITE-13347</td>
<td>Add InitialSize and MaxSize to DataRegion metrics</td>
<td>🔄</td>
<td>Aug 11, 2020</td>
<td>Aug 11, 2020</td>
<td>Nikolay Izhikov</td>
<td>Nikolay Izhikov</td>
<td>🔄</td>
<td></td>
<td>RESOLVED</td>
<td>Fixed</td>
</tr>
<tr>
<td>IGNITE-13224</td>
<td>Add metrics to OpenCensus exporter</td>
<td>🔄</td>
<td>Jul 07, 2020</td>
<td>Aug 18, 2020</td>
<td>Unassigned Dmitry Frolov</td>
<td>Dmitry Frolov</td>
<td>🔄</td>
<td></td>
<td>RESOLVED</td>
<td>Implemented</td>
</tr>
<tr>
<td>IGNITE-13173</td>
<td>Add offHeapEntries and similar values to cache metrics</td>
<td>🔄</td>
<td>Jun 22, 2020</td>
<td>Jul 16, 2020</td>
<td>Amelchev Nikita</td>
<td>Dmitry Frolov</td>
<td>🔄</td>
<td></td>
<td>RESOLVED</td>
<td>Fixed</td>
</tr>
<tr>
<td>IGNITE-13131</td>
<td>OpenCensusMetricExport erSpi and prometheus on server node</td>
<td>🔄</td>
<td>Jun 08, 2020</td>
<td>Jun 08, 2020</td>
<td>Unassigned Alexander Emelianov</td>
<td>Alexander Emelianov</td>
<td>🔄</td>
<td></td>
<td>OPEN</td>
<td>Unresolved</td>
</tr>
</tbody>
</table>
Showing 20 out of 121 issues