Debug cluster

[Flow control](#)

Flow control

API Priority and Fairness controls the behavior of the Kubernetes API server in an overload situation. You can find more information about it in the API Priority and Fairness documentation.

Diagnostics

Every HTTP response from an API server with the priority and fairness feature enabled has two extra headers: `X-Kubernetes-PF-FlowSchema-UID` and `X-Kubernetes-PF-PriorityLevel-UID`, noting the flow schema that matched the request and the priority level to which it was assigned, respectively. The API objects' names are not included in these headers (to avoid revealing details in case the requesting user does not have permission to view them). When debugging, you can use a command such as:

```
kubectl get flowschemas -o custom-columns="uid:{metadata.uid},name:{metadata.name}"
kubectl get prioritylevelconfigurations -o custom-columns="uid:{metadata.uid},name:{metadata.name}"
```

to get a mapping of UIDs to names for both FlowSchemas and PriorityLevelConfigurations.

Debug endpoints

With the `APIPriorityAndFairness` feature enabled, the `kube-apiserver` serves the following additional paths at its HTTP(S) ports.

You need to ensure you have permissions to access these endpoints. You don't have to do anything if you are using admin. Permissions can be granted if needed following the RBAC doc to access `/debug/api_priority_and_fairness/` by specifying `nonResourceURLs`.

- `/debug/api_priority_and_fairness/dump_priority_levels` - a listing of all the priority levels and the current state of each. You can fetch like this:

  ```
  kubectl get --raw /debug/api_priority_and_fairness/dump_priority_levels
  ```

  The output will be in CSV and similar to this:

  ```
  PriorityLevelName, ActiveQueues, IsIdle, IsQuiescing, WaitingRequests, ExecutingRequests, DispatchedRequests, RejectedRequest
  catch-all,         0,            true,   false,       0,               0,                 1,                  0,
  exempt,            0,            true,   false,       0,               0,                 0,                  0,
  global-default,    0,            true,   false,       0,               0,                 46,                 0,
  leader-election,   0,            true,   false,       0,               0,                 4,                  0,
  node-high,         0,            true,   false,       0,               0,                 34,                 0,
  system,            0,            true,   false,       0,               0,                 48,                 0,
  workload-high,     0,            true,   false,       0,               0,                 500,                0,
  workload-low,      0,            true,   false,       0,               0,                 0,                  0,
  ```

  Explanation for selected column names:

  - `IsQuiescing` indicates if this priority level will be removed when its queues have been drained.

- `/debug/api_priority_and_fairness/dump_queues` - a listing of all the queues and their current state. You can fetch like this:

  ```
  kubectl get --raw /debug/api_priority_and_fairness/dump_queues
  ```

  The output will be in CSV and similar to this:

  ```
  PriorityLevelName, Index,  PendingRequests, ExecutingRequests, SeatsInUse, NextDispatchR,   InitialSeatsSum, MaxSeatsSum, Tot
  workload-low,      14,     27,              0,                 0,          77.64342019ss,   270,             270,          0.8
  workload-low,      74,     26,              0,                 0,          76.95387841ss,   260,             260,          0.7
  ...
  leader-election,   0,      0,               0,                 0,          5088.87053833ss, 0,               0,            0.0
  leader-election,   1,      0,               0,                 0,          0.00000000ss,    0,               0,            0.0
  ...
  workload-high,     0,      0,               0,                 0,          0.00000000ss,    0,               0,            0.0
  workload-high,     1,      0,               0,                 0,          1119.44936475ss, 0,               0,            0.0
  ```

  Explanation for selected column names:

  - `NextDispatchR`: The R progress meter reading, in units of seat-seconds, at which the next request will be dispatched.
  - `InitialSeatsSum`: The sum of InitialSeats associated with all requests in a given queue.
  - `MaxSeatsSum`: The sum of MaxSeats associated with all requests in a given queue.
  - `TotalWorkSum`: The sum of total work, in units of seat-seconds, of all waiting requests in a given queue.

  Note: `seat-second` (abbreviate as `ss`) is a measure of work, in units of seat-seconds, in the APF world.

- `/debug/api_priority_and_fairness/dump_requests` - a listing of all the requests including requests waiting in a queue and requests being executing. You can fetch like this:

  ```
  kubectl get --raw /debug/api_priority_and_fairness/dump_requests
  ```

The output will be in CSV and similar to this:

```
PriorityLevelName, FlowSchemaName,   QueueIndex, RequestIndexInQueue, FlowDistingsher,                         ArriveTime,
exempt,            exempt,           -1,         -1,                 ,                                       2023-07-15T04:5
workload-low,      service-accounts, 14,         0,                  system:serviceaccount:default:loadtest, 2023-07-18T00:1
workload-low,      service-accounts, 14,         1,                  system:serviceaccount:default:loadtest, 2023-07-18T00:1
```

You can get a more detailed listing with a command like this:

```
kubectl get --raw '/debug/api_priority_and_fairness/dump_requests?includeRequestDetails=1'
```

The output will be in CSV and similar to this:

```
PriorityLevelName, FlowSchemaName,   QueueIndex, RequestIndexInQueue, FlowDistingsher,                         ArriveTime,
exempt,            exempt,           -1,         -1,                 ,                                       2023-07-15T04:5
workload-low,      service-accounts, 14,         0,                  system:serviceaccount:default:loadtest, 2023-07-18T00:1
workload-low,      service-accounts, 14,         1,                  system:serviceaccount:default:loadtest, 2023-07-18T00:1
```

Explanation for selected column names:

- `QueueIndex`: The index of the queue. It will be -1 for priority levels without queues.
- `RequestIndexInQueue`: The index in the queue for a given request. It will be -1 for executing requests.
- `InitialSeats`: The number of seats will be occupied during the initial (normal) stage of execution of the request.
- `FinalSeats`: The number of seats will be occupied during the final stage of request execution, accounting for the associated WATCH notifications.
- `AdditionalLatency`: The extra time taken during the final stage of request execution. FinalSeats will be occupied during this time period. It does not mean any latency that a user will observe.
- `StartTime`: The time a request starts to execute. It will be 0001-01-01T00:00:00Z for queued requests.

Debug logging

At `-v=3` or more verbosity, the API server outputs an httplog line for every request in the API server log, and it includes the following attributes.

- `apf_fs`: the name of the flow schema to which the request was classified.
- `apf_pl`: the name of the priority level for that flow schema.
- `apf_iseats`: the number of seats determined for the initial (normal) stage of execution of the request.
- `apf_fseats`: the number of seats determined for the final stage of execution (accounting for the associated `watch` notifications) of the request.
- `apf_additionalLatency`: the duration of the final stage of execution of the request.

At higher levels of verbosity there will be log lines exposing details of how APF handled the request, primarily for debugging purposes.

Response headers

APF adds the following two headers to each HTTP response message. They won't appear in the audit log. They can be viewed from the client side. For client using `klog`, use verbosity `-v=8` or higher to view these headers.

- `X-Kubernetes-PF-FlowSchema-UID` holds the UID of the FlowSchema object to which the corresponding request was classified.
- `X-Kubernetes-PF-PriorityLevel-UID` holds the UID of the PriorityLevelConfiguration object associated with that FlowSchema.

What's next

For background information on design details for API priority and fairness, see the enhancement proposal.