

In this analysis we will see how the 3 main Mysql data fetching methods in PHP performance under different loads

In most programming situations regarding php-mysql you will need to fetch data from the db to populate your application with data

There are 3 most popular methods of getting more than one data when fetched.

- `mysql_fetch_assoc()` : This gets you an associative array of data.
- `mysql_fetch_array()` : This returns a combination array of associative elements as well as data with numerical index.
- `mysql_fetch_object()` : Returns an object with properties that correspond to the fetched row.

We are comparing the 3 methods here based on benchmark tests done by spearhead softwares on various load conditions. In the test we run each methods result dataset containing 10,100,1000,10000,100000,1000000,10000000 data and see how fast and efficient each function operates. Each data set contains a unique id element, int & a string element which makes it the normal set of values which is being retrieved in most normal cases.

graph

From this we can clearly see that the default functionality of `mysql_fetch_assoc` is running at a higher performance than the other 2 methods (`mysql_fetch_array` & `mysql_fetch_object`). `mysql_fetch_object` performance is justified as it returns objects instead of native arrays which will always bring in a better memory usages than any other kinds of output array. And for using the Object oriented result set approach the other 2 functions cannot satisfy. Lets take the case of `mysql_fetch_array()`; The problem here is in daily programming usages many php programmers have used the `mysql_fetch_array()` as such without going into the details of its usage. By default `mysql_fetch_array` returns 2 types of results sets as given by its default `result_type` value **MYSQL_BOTH**

- result_type : MYSQL_NUM - which gives us only a numbered index array eg:
array([0]=>'apple')
 - result_type : MYSQL_ASSOC - which us only associative array eg: array('fruit'=>'apple')
- where 'fruit' will be the database column name or alias used.

So when we use mysql_fetch_array we get a duplicate set of data consisting of both numbered index as well as associative arrays which ofcourse is our performance blocker. So when ever you are using mysql_fetch_array function always specify the return_type of the result set array needed.

And in most cases we will need an associative array during iterative data management in php so the best function to call will be as shown in the graph ,performance wise
-mysql_fetch_assoc()

Hope you liked our article. If you have more ideas regarding performance in this category, do comment .