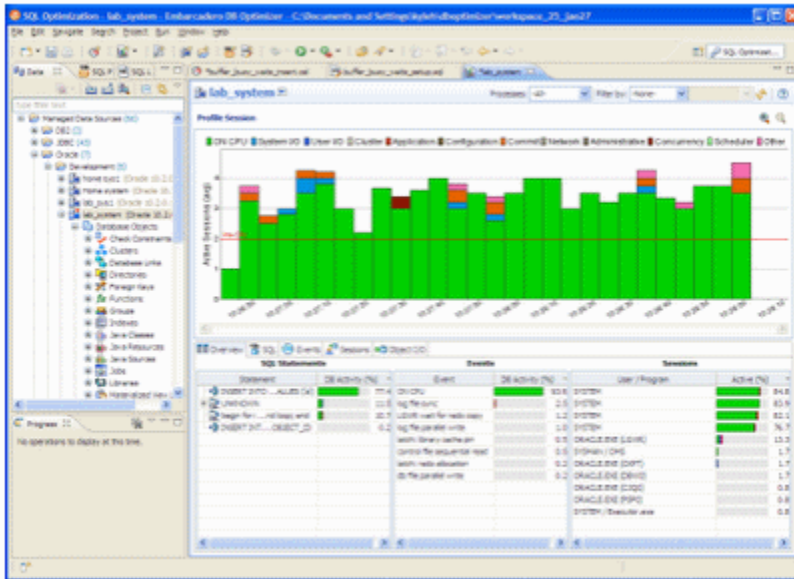# The machine caused the problem

Now that the application is tuned and the database is tuned, let's run a bigger load:

We can see that the CPU load is constantly over the max CPU line. How can we have a bigger CPU load than there are actually CPUs on the machine? Because the demand for CPU is higher than the CPU available on the machine.

In the image above, there are two CPUs on the machine but an average of three users think they are on the CPU, which means that on average one user is not really on the CPU but ready to run on the CPU and waiting for the CPU.

At this point we have two options. In this case we are only running one kind of load, the insert. For inserts we can actually go even further tuning this insert and use Oracle's bulk load commands:

```
declare

    TYPE IDX IS TABLE OF Integer INDEX BY BINARY_INTEGER;
    MY_IDX IDX;

    begin

        for i in 1..8000 loop

            MY_IDX(i):=1;

        end loop;
        FORALL indx IN MY_IDX.FIRST .. MY_IDX.LAST
        INSERT INTO foo ( dummy )
        VALUES ( MY_IDX(indx) );
        COMMIT;

    end;
    /
```

But if this was an application that had a lot of different SQL and the SQL load was well distributed across the system then we'd have a case for adding more hardware to the system. Making the decision to add more hardware can be a difficult decision because in general the information to make the decision is unknown, unclear or just plain confusing, but DB Optimizer makes it easy and clear, which can save weeks and months of wasteful meetings and debates. For example:
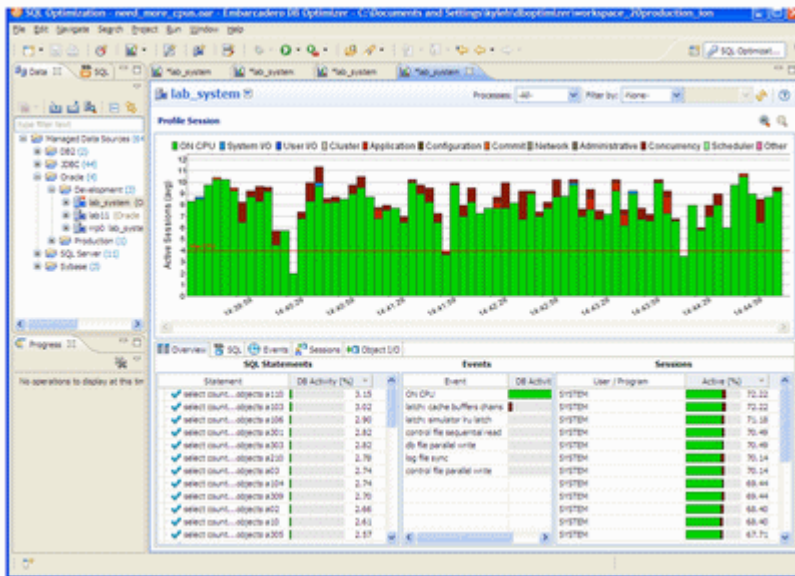
If we look in the bottom left, there is no SQL that takes up a significant amount of load, there is no outlier SQL that we could tune and gain back a lot of wasted CPU. We'd have to tune many SQL and make improvements on most of them to gain back enough CPU to get our load down below the max CPU line. In this case, adding CPUs to the machine might be the easiest and most cost effective solution.

Conclusion:

- With the load chart we can quickly and easily identify the bottlenecks in the database, take corrective actions, and see the results. In Part 1, we had an application problem, in Part 2 we had three database configuration issues, and in Part 3 we had a hardware sizing issue. In all three instances DB Optimizer provides a clear and easy presentation of the data and issues making solutions clear.

IDERA | **Products** | **Purchase** | **Support** | **Community** | **Resources** | **About Us** | **Legal**