How to use the Tomograph

The Tomograph illustrates the parallel processing behavior of individual queries. The information can be used to identify possible expensive and blocking portions in your queries. It helps the developers to understand the actual parallel behavior and may spot areas for improvement.

In a Tomograph output, the top most part ("memory in GB" and "IO per ms") illustrates the memory RSS (shown as blue dots) and I/O activities (reads as grey dots, writes as red dots) as reported by the OS during each heartbeat event. Note that all values displayed here reflect system wide activities, thus they include not only MonetDB activities. Also note that the I/O counts shown here do not reflect the actual amount of data read/written to the hardware drives, as this information is generally not available to all users of a Linux system.

The second part ("cores") shows a heat map of the available CPU cores in the system, one line per CPU. This information is gathered from a heart-beat monitor in the server. The arrival rate can be controlled using the "--beat" option. The colors show the level of core utilization (white <2% yellow>2%, gold>25%, orange>50%, orangered>75%, red >90%). Again, the information given here includes the activities of the whole system.

The third part ("worker threads") shows the activities of all MonetDB worker threads. Therefore, this part contains the most interesting information to find the bottlenecks in query performance. Note that the activities of the software threads here are not *directly* linked to the CPU activities shown in the second part, as tomograph does not give any information about which core is executing which thread (although some deductions can be made). Along the time axes, we show for each software thread the MAL instructions in progress. A hollow box with smaller height indicates that the thread was waiting for the next instruction to become available for execution. The little red-filled boxes, often placed at the end of a thread, indicate there is no instruction left at that moment.

Finally, at the bottom of a tomograph, we provide detailed statistics on the most frequently executed MAL instructions. Among others, the "parallelism" is calculated by taking the sum over the maximum worker running time X numbers of cores and divide it by the reported number of ticks spent on each MAL instruction.