

Visualization Priority

Problem 1 (Visualization Priority) *A syntax for developers to specify visualization constraints and priorities, in order to come up with placement policies.*

Suppose we have:

Visualizations $V = \{V_1, V_2, \dots, V_n\}$

Interactions $I = \{I_1, I_2, \dots, I_n\}$

where V and I have many-to-many relations. For example, we might have a view that can be updated by multiple different interactions, and we can have an interaction that changes multiple views simultaneously. This means prioritization should be expressed at the granularity of $\langle V_m, I_n \rangle$ pairs.

Specifically, there could be hard and soft constraints. Developers specify hard constraints like “Interaction I_1 should update V_2 in 100ms for 95% of the time”, and soft constraints like “Interaction I_1 updating V_1 should be faster than I_2 updating V_2 ”.

For hard constraints, the goal would be to find the lowest priority that meets each of the constraints, if there exists such a placement.

If developers don’t have specific latency bounds in mind, they can specify soft constraints. A naive approach would be to let the developer to specify the weights of selected $\langle V_m, I_n \rangle$ pairs, and the system calculates a placement policy that minimizes the weighted average latency based on remaining resources. The execution time should also be considered.

An example spec would be:

Hard:

(I_1, V_1) within 100ms with 100% confidence

$(I_2, *)$ within 200ms with 95% confidence

Soft:

(I_1, V_1) weight 2

(I_1, V_2) weight 3

$(*, V_1)$ weight 5

For example, one may use “ $(*, V_1)$ weight 1, $(*, V_2)$ weight 1” to suggest that “most interactions that updates V_1 and V_2 should be prioritized”.