Why So Consistent?
Dynamic Enforcement of Fine-grained Weak Consistency

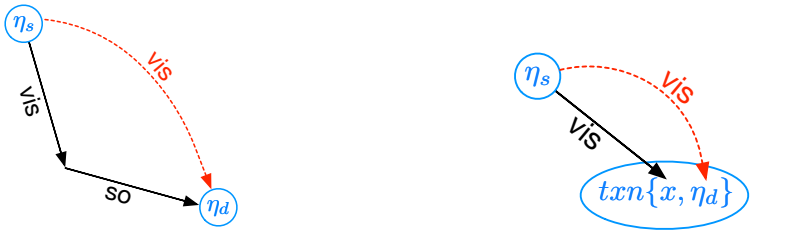
KIA RAHMANI, Purdue University
GOWTHAM KAKI, Purdue University
SURESH JAGANNATHAN, Purdue University

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

ACM Reference format:
Kia Rahmani, Gowtham Kaki, and Suresh Jagannathan. 2017. Why So Consistent?. *PACM Progr. Lang.* 1, 1, Article 1 (January 2017), 3 pages.
DOI: 10.1145/nnnnnnn.nnnnnnn

2017. 2475-1421/2017/1-ART1 $15.00
DOI: 10.1145/nnnnnnn.nnnnnnn



(a) Monotonic Read (MR) (b) Monotonic Atomic View (MAV)

Fig. 1. Representation of Contracts

4.2 Specification Language

. Users in our system are offered with a contract language to specify their application-level consistency requirements. Developers must define a contract for each operation in the application, since the overall correctness of the program requires different levels of consistency for each operation. The constructing blocks of contracts are the relations over the set of effects generated by operations. Relations $vis(a, b)$, $so(a, b)$ and $sameobj(a, b)$ are defined to relate effects a and b, respectively if b was visible to the operation that generated a, if they are the result of two operations submitted by the same session, respecting their submission time, and if they were performed on the same data object.

Contracts are basically logical formulae that specify *when* (the pre-condition), *what* effects should be visible to an operation. For example, the Monotonic Read (MR) session guarantee, requires all the effects that are visible to an operation in a session, to be also visible to the later operations in that session. Figure 1a shows how this can be succinctly defined using the relations over the effects. It simply states that all effects η_s that satisfy the relation $(vis^{-1}(so^{-1}))$ to an effect η_d, must also be visible to η_d. We generalized this structure and came up with blah blah