Making Dyninst Thread-Safe for Tools
Mark Krentel, Bill Williams, Xiaozhu Meng,
Josh Stone, Steve Song and Alexander Morris

Motivation  Tools built on top of Dyninst would like to use multiple threads for
efficiency. But currently, making queries to ParseAPI or SymtabAPI is not thread
safe. Further, Dyninst does its own analysis of binaries with a single thread.

Discussion  We discussed various problems with adding threads to parts of Dyninst,
especially ParseAPI and SymtabAPI. For example, at what granularity could ParseAPI
analyze a file, what requests and queries would modify internal data structures, which
operations could proceed concurrently, where locks would be needed, etc.

Proposal  Dyninst could accept a configure-time parameter to identify what thread-
ing model to use. For example, MPI_Init_thread() takes a parameter to indicate
the level of thread support (single threaded, threaded but serialized, fully threaded,
etc.) Not all thread models would need to be implemented right away.

1. (Non-threaded) The application is not threaded and Dyninst would not use any
internal locks or synchronization.

2. (Partially-threaded) The application is threaded but does not concurrently an-
alyze a binary (write operation) and make queries on it (read operation). How-
ever, the application may perform concurrent queries on an already analyzed
binary. For example, the application could use two phases where it first analyzes
a binary serially and then queries the binary from multiple threads.

3. (Fully-threaded) The application makes unrestricted, concurrent queries and
incremental analysis.

4. (Internally threaded) Dyninst itself uses threads for its analysis.

Note: a subtle problem is that due to lazy evaluation and caching, some queries that
would appear as read-only and reentrant actually trigger changes in the internal data
structures. We don’t want to expose these details in the API, so it makes sense to let
Dyninst decide what needs to be locked internally.

Followup  Rice and Wisconsin can followup with test programs and begin adding
locking to ParseAPI and SymtabAPI to support some threaded cases.