Proposal for High-Performance Wall-Clock Timers in UPC
Version 0.2

Dan Bonachea

bonachea@cs.berkeley.edu

February 5, 2007

This document describes a proposed extension to the UPC language specification which has NOT been ratified by the UPC Consortium. The information presented here does not necessarily reflect the position or the policy of the UPC Consortium and no official endorsement should be inferred.
## Contents

A Proposed Additions and Extensions 3
   A.1 UPC Utilities / High-Performance Wall-Clock Timers `<upc.h>` 3
      A.1.1 Background ........................................ 3
      A.1.2 `upc_tick_t` Type .................................. 4
      A.1.3 `upc_tick_t` Type .................................. 4
      A.1.4 Implementation Notes .............................. 5

Index 6
A Proposed Additions and Extensions

A.1 UPC Utilities / High-Performance Wall-Clock Timers <upc.h>

This subsection provides extensions of [ISO/IEC00 Sec 7.23]. All the characteristics of library functions described in [ISO/IEC00 Sec 7.1.4] apply to these as well. Implementations that support this interface shall predefine the feature macro \_UPC\_TICK\_ to the value 1.

A.1.1 Background

A.1.1.1 Rationale

The \texttt{upc\_tick\_t} type and associated functions (hereafter referred to as the \texttt{upc\_tick} interface) provide convenient and portable support for querying high-precision system timers for obtaining high-precision wall-clock timings of sections of code. Many hardware implementations offer access to high-performance timers with a handful of instructions, providing timer precision and overhead that can be several orders of magnitude better than can be obtained through the use of existing interfaces in [ISO/IEC00] or POSIX (e.g. the \texttt{gettimeofday()} system call).

EXAMPLE 1: an example of the \texttt{upc\_tick} interface in use:

```c
#include <upc.h>
#include <stdio.h>

upc_tick_t start = upc_ticks_now();
    compute_foo(); /* do something that needs to be timed */
upc_tick_t end = upc_ticks_now();

printf("Time was: \%f seconds\n", upc_ticks_to_ns(end-start)/1.0E-9);
```
A.1.2 \texttt{upc\_tick\_t} Type

1 The following type is defined in \texttt{upc.h}

2 \texttt{upc\_tick\_t} is an unsigned integral type representing a quantity of abstract
  timer ticks, whose ratio to wall-clock time is system-dependent and thread-
  dependent.

3 \textbf{NOTE:} \texttt{upc\_tick\_t} values are thread-specific quantities with a thread-specific
  interpretation (e.g. they might represent a hardware cycle count on a partic-
  ular processor, starting at some arbitrary time in the past). More specifically,
  \texttt{upc\_tick\_t} values do \textit{not} provide a globally-synchronized timer (i.e. the si-
  multaneous absolute tick values may differ across threads), and furthermore
  the tick-to-wall-clock conversion ratio might also differ across UPC threads
  (e.g. on a system with heterogeneous processor clock rates, the tick val-
  ues may advance at different rates for different UPC threads). Therefore
  as a rule of thumb, \texttt{upc\_tick\_t} values and intervals obtained by \textit{different}
  threads should never be directly compared or arithmetically combined, with-
  out first converting the relevant tick intervals to wall time intervals (using
  \texttt{upc\_ticks\_to\_ns}).

4 \texttt{UPC\_TICK\_MAX} and \texttt{UPC\_TICK\_MIN} are constants of type \texttt{upc\_tick\_t} which are
  defined in \texttt{upc.h}. They respectively provide the minimal and maximal values
  representable in a variable of type \texttt{upc\_tick\_t}.

A.1.3 \texttt{upc\_tick\_t} Type

A.1.3.1 The \texttt{upc\_ticks\_now} function

\textbf{Synopsis}

1 \begin{verbatim}
#include <upc.h>

upc\_tick\_t upc\_ticks\_now();
\end{verbatim}

\textbf{Description}

2 \texttt{upc\_ticks\_now} returns the current value of the tick timer for the calling
  thread, as measured from an arbitrary, thread-specific point of time in the
  past (which is fixed during any given program execution).
A.1.3.2 The upc_ticks_to_ns function

Synopsis

```c
#include <upc.h>
#include <stdint.h>

uint64_t upc_ticks_to_ns(upc_tick_t ticks);
```

Description

The `upc_ticks_to_ns` converts a quantity of ticks obtained by the calling thread into wall-clock nanoseconds.

The function always succeeds.

A.1.4 Implementation Notes

A prototype implementation of this extension is available in the Berkeley UPC compiler (http://upc.lbl.gov) and the GCCUPC compiler with the Berkeley UPC runtime. In the prototype, all names use the `bupc_` naming prefix, and the appropriate feature macro is `__BERKELEY_UPC_RUNTIME__`.

A compiler-independent, portable and high-performance implementation of the functionality provided in this interface is available in the GASNet tools-only distribution (http://gasnet.cs.berkeley.edu). That implementation is written in ISO C and assembly and distributed under a BSD license, to assist implementation efforts in other UPC compilers (the tools-only distribution is decoupled from the GASNet communication services).
Index

__UPC_TICK__, 3

cycle counter, 3
tick counter, 3
timer, 3

upc.h, 3
UPC_TICK_MAX, 4
UPC_TICK_MIN, 4
upc_tck_t, 4
upc_tcks_now, 4
upc_tcks_to_ns, 5

wall-clock, 3