About Task Splitting and Distribution to Clients

Vincent LEFÈVRE

AriC, INRIA Grenoble – Rhône-Alpes / LIP, ENS-Lyon

Journées TaMaDi, Paris, 2012-10-18/19

Current Implementation

- Radix 2 only.
- For each function and exponent, the whole binade is split into 2^s intervals of the same size, numbered from 0 to $2^s - 1$.
- 3 parameters are chosen by the user: s and the numbers of the first interval i_{min} and of the last interval i_{max} to consider (default: 0 and $2^s - 1$, i.e. the whole binade).
- If intervals of different sizes are needed (because of huge variations of the exponent of the result, e.g. around 1 for log, around $k\pi$ for sin and around $k\pi/2$ for tan), different splittings $(s, i_{\mathrm{min}}, i_{\mathrm{max}})$ must be done manually by the user, as if the binades were different.

  No possible dynamic changes, e.g. if an interval is to difficult to test or a machine is too slow.
- Implementation limits: the interval numbers are 32-bit or 64-bit integers, thus must fit on 32 bits; memory in $O(i_{\mathrm{max}})$ instead of $O(i_{\mathrm{max}} - i_{\mathrm{min}})$.

Current Implementation [2]

- The server supports only one task (function, exponent, splitting) at a time. Limited support for automatic task switching (by using a new protocol) was added in 2008 in order to test the x^n functions.
- The server is monothreaded and accepts only one connection at a time. $\rightarrow$ Timeouts observed in practice.
- Very basic security, no authentication. Anyone can send commands to the server; thus DoS is easy (however this is the only problem, since the results are not handled by the server, and their integrity can be checked by other programs).

Toward a New System: Task Splitting

Requirements:

- Support for different tasks at the same time (e.g. different functions, different exponents. . .).
- Flexible interval splitting, possibly dynamic.
- Avoid too much fragmentation.

$\rightarrow$ Subtask identified by a task id and an interval whose endpoints are multiple-precision integers?

Each subtask is assumed to be independent (or can be made independent) from the other ones.

Other possible requirements:

- Dynamic resplitting.
- Multi-dimensional boxes instead of intervals?

Toward a New System: Distribution to Clients

Some existing implementations of parallelization systems:

- **OpenMP**. Very easy to use, but limitations: same machine, very basic distribution (e.g. uniform loop parallelization), no queue system.
- **MPI**. Limitations: no queue system (must be added), seems too static.
- **BOINC**. Probably does too much. → Incompatible with the use of other batch systems like SGE. (?)
- **SGE**. This is a batch system, available on the MI-LIP machines. Jobs could be generated statically and submitted. An additional client-server system would be more flexible (and usable without SGE), like currently.