Performance Model Manager

User Manual

Contents

1 Introduction

Performance Model Manager (PMM) is an open source GNU Public Licenced tool for experimentation in the use of Functional Performance Models (FPMs). An FPM describes the speed of a computational routine in terms of the routine's input parameters. It focuses on addressing issues surrounding the construction, maintenance and use of FPMs. To this end, it has three main features:

- It implements the Geometric Bisection Building Procedure for multi-parameter FPMs, optimizing the construction of a problem's performance model.
- It permits the construction of models for a large number of problems by implementing a flexible benchmarking scheduler.
- It provides access to the models in a variety of ways, so that they may be visualised or used to make scheduling decisions.

Construction of the FPM of some general computational procedure is supported by requiring that the user provides a benchmarking executable which behaves according to a simple protocol.

An example implementation of such a 'benchmark binary' is given in Chapter 4.2 along with details of its configuration within PMM. Further examples are also included in the source distribution.

Models can be constructed on demand or in the background by the `pmmd` daemon. Construction of multiple models can be scheduled according to a variety of policies. They will be constructed in turn according to their priority and scheduling criteria.

Access to models will be made available via an API in a future release of PMM, at present only viewing of models is possible, via a plotting program: `pmm_view`.

This manual continues to Chapter 2 where compilation and installation is described. Then the PMM configuration file is described in Chapter 3. Chapter 4 provides notes on how to write a benchmark binary and configure it as a 'routine' to be modelled within PMM.

2 Installation

2.1 Requirements

PMM is developed for the Linux platform but may also compile on other POSIX operating systems. The following softwares are required to install PMM:

- Gnuplot
- GNU Make
- GCC compiler suite (tested with 4.x series only)
- libxml2 2.6.0 or greater
- GNU Scientific Library

The following are optional but enable certain features:

- Octave (2.9.14 or greater) is required for multi-parameter model construction
- PAPI (4.0.0 tested) is required for higher resolution timing and automatic complexity calculation
- GotoBLAS2 is required for for further example problems

2.2 Compiling & Installing

Installation of PMM uses a hierarchy of directories under a certain prefix, by default `/usr`. If this is not desirable the build should be configured with the `--prefix=<dir>` option. A typical installation follows:

```
$ tar -xzf pmm-0.0.1.tar.gz
$ cd pmm-0.0.1
$ ./configure --prefix=$PWD/install
$ make && make install
```

Configuration options to note:

- `--enable-debug` enable debugging messages and flags
- `--disable-octave` disable use of octave and multi-parameter model support
- `--disable-benchgslblas` disable compilation and installation of demonstration GSL problem benchmarks
- `--enable-benchgotoblas2` enable compilation and installation of demonstration GotoBLAS2 problems
- `--with-papi[=path]` enable use of PAPI with optional specification of PAPI installation path

Further options can be viewed by running `./configure --help` .

After installation, the PMM daemon is started by executing the `pmmd` binary and the PMM viewer program is run via the `pmm_view` binary.

3 Configuration

PMM is distributed with a default configuration which will be installed under:

```
<prefix>/etc/pmmd.conf[.sample]
```

This can serve as a template for a user's own configuration and contains sane values for all options. If example problem benchmarks are built, the sample configuration will also describe those problems.

The configuration file has an heirarchical XML structure Configuration is described between `<config>` root element tags. Under this, the load monitor facility is described by a `<load_monitor>` element, and each routine for which a model is to be built, is described by a `<routine>` element.

In the following sections, each element in the configuration file is described. If an element has a default value, it need not be explicitly set in the configuration file, on the other hand, some options must be set. This information, along with the type of the expected element value (string, integer, etc.) and what exactly the element describes is detailed below.

3.1 General Configuration

The following elements (which can be seen in context in Listing 3.1) define some general application configurable options and come directly under the `<config>` tags:

- `<main_sleep_period>` (*integer, default:1*) The benchmark scheduler checks the system state ever *n* seconds and this period may be configured here. The default value is suitable and this variable is made configurable mostly for developmental purposes.

- `<model_write_time_threshold>` (*integer, default:60*) When benchmarking problems of very small size, which execute very quickly, the manager may become overloaded by writing the model to disk after each execution. This option allows us to configure how often the model will be saved to disk, i.e. after a total of *n* seconds has been spent benchmarking a particular model it will be written to disk. The default value is suitable and this variable is made configurable mostly for developmental purposes.

- <model_write_execs_threshold> (*integer, default:10*) This option serves the same purpose as the previous one, except that it specifies the number of benchmark executions that must occur before it is written to disk. It may take hundreds of small benchmarks exceed the time threshold (above), so this second threshold allows us to write based on execution frequency as well. The default value is suitable and this variable is made configurable mostly for developmental purposes.

Listing 3.1: Basic Configuration

```
<?xml version="1.0"?>
<config>
   <main_sleep_period>1</main_sleep_period>
   <model_write_time_threshold>60
      </model_write_time_threshold>
   <model_write_execs_threshold>20
      </model_write_execs_threshold>

   <load_monitor>
      <load_path>/usr/var/pmm/loadhistory</load_path>
      <write_period>60</write_period>
      <history_size>60</history_size>
   </load_monitor>

   ....
</config>
```

3.2 Load Monitor Configuration

The load monitoring facility is is described by a <load_monitor> element, this has the following children:

- <load_path> (*string, required*) path to a file where load observations are recorded

- <write_period> (*integer, default:360*) frequency with which to save the load file to disk (in seconds)

- `<history_size>` (*integer, default:60*) number of load observations to store

3.3 Routine Configuration

Each routine is described by a `<routine>` element. Routines have detailed descriptions of the parameters to be passed to them and the construction method that should be used to build their models. An example routine configuration can been seen in Listing 3.2, it describes a 2-parameter routine. First, the general options are set using the following child elements:

- `<name>` (*string, required*) The routine name.
- `<exe_path>` (*string, required*) The path to the benchmarking executable
- `<model_path>` (*string, required*) The path to the file where the performance model of the routine will be saved.
- `<priority>` (*integer, default:0*) The construction priority this routine has (logically, the higher the value, the higher the priority)

The parameters of a routine are described by a `<parameters>` elmenet. These are the parameters that will be passed to the benchmark binary which ultimately executes the routine which is being modeled. The parameters passed to the benchmark are those that influnce the volume of computations or the speed at which the computations are carried outt. This is further described in Chapter 4. The first child of the `<parameters>` element must be the number of parameter descriptions which will follow:

- `<n_p>` (*integer, required*) number of parameters which the benchmark accepts

Following that, each parameter is described by a `<param>` element. The `<param>` element has a number of child elements which are:

- `<order>` (*integer, required*) The order in which this parameter should be passed to the benchmark binary.
- `<name>` (*string, required*) The name of this parameter.
- `<min>` (*integer, required*) The minimum value this parameter may have. If modelling the performance of the processor while operating in cache only is *not* important, this should be set so the overall problem size is large enough to occupy main memory.

- `<max>` (*integer, required*) The maximum value this parameter may have. This should be large enough to induce significant paging.
- `<stride>` (*integer, default:1*) The stride with which this parameter should be incremented. Stride influences the climbing phase of optimised construction (where successive benchmarks are incremented in size by this value) as well as naive construction (where all points on the stride between min and max are benchmarked). A reasonable value for stride would be, for example, 1/100th of the range between max and min. If stride is too low, excessive time may be spent building a model.
- `<offset>` (*integer, default:0*) Offset for this parameter. If required, you can specify that a parameter value must always be a certain offset from zero.
- `<fuzzy_max>` (*boolean, default:false*) This specifies that the maximum parameter size defined is not a true max and speed at this maximum should be measured.

  In normal circumstances the FPM is constructed across a complete range of problem sizes, from small to so large that speed is effectively zero. The maximum parameter value will be so large that it induces heavy paging. Speed at this maximum is not measured, but assumed to be zero. If this is *not* the case, and the maximum parameter size will not induce heavy paging, `<fuzzy_max>` must be set to *true* for the GBBP algorithm to complete successfully.

Directives for the construction method must be described by a `<construction>` element. It has the following child elements:

- `<method>` (*string, default:gbbp*) The construction method, this element may have the following values:
  - *gbbp* - the Geometric Bisection Building Procedure will be used to select benchmark points, minimising the number of points required to accurately estimate the model
  - *naive* - all possible points between the parameter ranges will be benchmarked.
  - *rand* - points between the parameter ranges will be selected at random
- `<min_sample_num>` (*integer, default:1*) Specify the minimum number of benchmarks to be taken at a single point in the model. Once this is met, the point will be considered as measured and the construction method will proceed to the next point of its choosing.

- <min_sample_time> (*integer, default:0*) Specify the minimum number of seconds that should be spent in the benchmarking of a single point before it is considered as measured. I.e. if set to 60 seconds, a benchmark taking 20 seconds will be measured 3 times.

Finally, priority and scheduling policy may be specified. When multiple routines are configured in PMM priorities allow the user to specify which models will be built first. Scheduling policies allow the user limit the execution of benchmarks to certain time periods or certain system conditions.

- <priority> (*integer, default:0*) Priority of construction for the routine. Higher priority routines will have their models constructed before lower ones

- <condition> (*string, default:now*) Condition under which benchmarking of a routine is permitted. Note: Once started, a benchmark will not be interrupted, even if the conditions that permitted its execution have changed to ones which would otherwise prevent execution.
  - *now* - construction is permitted at all times
  - *idle* - construction is only permitted when the observed 5 minute load average is less than 0.10 (note: the act of benchmarking will influence the load average of the system. After the benchmark is complete, PMM will probably have to wait 5 minutes before the next execution can occur)
  - *nousers* - construction is only permitted when no users are logged into the system. Logged in users would be those reported by utilities such as `w`, `who`, `users` and so on.

Listing 3.2: Routine Configuration Example

```
<routine>
   <name>dgemm2</name>
   <exe_path>/usr/local/lib/pmm/dgemm2</exe_path>
   <model_path>/usr/local/var/pmm/dgemm2_model</model_path>
   <parameters>
      <n_p>2</n_p>
      <param>
         <order>0</order>
         <name>m</name>
         <min>32</min>
         <max>4096</max>
```

```
            <stride>32</stride>
            <offset>0</offset>
        </param>
        <param>
            <order>1</order>
            <name>n</name>
            <min>32</min>
            <max>4096</max>
            <stride>32</stride>
            <offset>0</offset>
        </param>
    </parameters>
    <construction>
        <method>gbbp</method>
        <min_sample_num>5</min_sample_num>
        <min_sample_time>120</min_sample_time>
    </construction>
    <condition>now</condition>
    <priority>30</priority>
</routine>
```

4 Building the FPM of a Computation

This chapter outlines what a user must do to have PMM build the FPM of some computation. The computation may be a library subroutine, a code fragment or an entire process. Throughout this document this computation will be referred to as a *routine*. The users routine must be wrapped in a benchmarking binary or script which should behave in a specific manner:

- It must accept arguments from the command line which define the volume of computations it must carry out.
- It must execute and time the computation that is to be modeled. Execution may be via a script or compiled binary, written in any language, and the details of how it perform or times the computation do not concern the PMM tool. If the benchmark is written in C/C++, PMM provides some utilities to aid this in a shared library, `libpmm`.
- It must output timing and volume of computations (complexity) in a standard manner. `libpmm` also supports this.

Implementation of this benchmark is a task left to the user. The following sections describe how to choose input parameters, write the benchmark and configure PMM to build a FPM for the routine.

4.1 Choosing Parameters of a Routine

The first step a user must take is to identify the parameters of the routine which effect the volume of computations it must carry out. Typically, the volume of computations would be floating point operation count, however PMM is agnostic to the type of computations the routine carries out, and the volume may be expressed as the user wishes. The performance model that we build will be expressed in terms of these parameters. Throughout this chapter we will refer to an example of a square matrix multiplication. In this scenario, there is only one parameter that effects the volume of computations, N, the length of a matrix side in the multiplication.

For a more general case, were two matrices of sizes $N \times K$ and $K \times M$ are multiplied and the result stored in an third matrix of size $N \times M$, then the volume of computations would depend on three parameters, N, M and K.

A general purpose matrix multiplication routine usually has other associated parameters defining transpositions of the input data and other coefficients. These however do not contribute significantly to the computational complexity of the routine and they should not be considered as parameters of the model in the PMM framework. It is important to note than building an FPM which is in terms of more than one parameter is very intensive as the number of points required to accurately approximate scales exponentially with the number of parameters the model is in terms of. Any parameters of a routine that can be excluded from the functional performance model should be.

4.2 Writing a Benchmark for PMM

For PMM to build the performance model of a routine, it must be able to execute benchmarks of that routine for various problem sizes. As previously stated, the problem's size is determined by the parameters which effect the computational complexity of the routine, and the performance model is a function of these parameters.

PMM must be provided with an executable which carries out a benchmark with given input parameters. The user must write this executable so that it behaves in a specified way. PMM is distributed with the source of a number of example benchmarks, here we will list one and reference it as the required behaviours are described below. Listing 4.1 shows an example benchmark for a square matrix multiplication routine. The multiplication is provided by the Gnu Scientific Library. Note in a square matrix multiplication, the volume of computations is determined by the size of one side of matrices to be multiplied.

The benchmark code behaves in the following manner:

- The executable must accept a number of parameters on the command line. These parameters will also be described in the configuration entry for the routine. As of version 0.0.1 parameters can only have integer types. (Lines 17-23)

- Based on the parameters passed on the command line, the benchmark must initialise memory and data structures that are to be passed to the routine. If the computation that is to be modelled is just a simple code fragment, no allocation of memory that occurs within the code fragment should be done in this initialisation phase. (Lines 29-35)

- The benchmark must start a timer, either using timers provided by the PMM shared library, libpmm, or using his own methods (Lines 38-41)
- The benchmark must execute the routine directly after timing is initiated (Line 44)
- the benchmark must stop timing directly after the routine has finished (Line 47)
- the benchmark must print on a single line, to `stdout`, the seconds and microseconds, separated by a single space, elapsed during the routine execution. This can be done using a function provided by libpmm or the users own method. (Line 50)
- the benchmark must print on a new line, the volume of computations made by the routine (typically the number of floating point operations carried out). Long long integers are supported. (Lines 26,38,50)
- on successful completion of the above operations, the benchmark should terminate and return successful exit status, PMM expects this to be equivalent to `EXIT_SUCCESS` as defined by the C standard. (Line 59)

Listing 4.1 shows an example benchmark for a square matrix multiplication routine provided by GSL. The inline comments refer to each of the points made above

Listing 4.1: Square Matrix Multiplication Benchmark

```
1  #include <stdlib.h>
2  #include <stdio.h>
3  #include "pmm_util.h"
4  #include <gsl/gsl_blas.h>
5
6  #define NARGS 1
7
8  int main(int argc, char **argv) {
9
10         /* declare variables */
11         gsl_matrix *A, *B, *C;
12         double arg;
13         size_t n;
```

```c
	long long int c;

	/* parse arguments */
	if(argc != NARGS+1) {
		return PMM_EXIT_ARGFAIL;
	}
	if(sscanf(argv[1], "%lf", &arg) == 0) {
		return PMM_EXIT_ARGPARSEFAIL;
	}
	n = (size_t)arg;

	/* calculate complexity */
	c = 2*n*n*(long long int)n;

	/* initialise data */
	A = gsl_matrix_alloc(n, n);
	B = gsl_matrix_alloc(n, n);
	C = gsl_matrix_alloc(n, n);

	gsl_matrix_set_all(A, 2.5);
	gsl_matrix_set_all(B, 4.9);
	gsl_matrix_set_zero(C);

	/* initialise timer */
	pmm_timer_init(c);

	/* start timer */
	pmm_timer_start();

	/* execute routine */
	gsl_blas_dgemm(CblasNoTrans, CblasNoTrans, 1.0, A,
	   B, 0.0, C);

	/* stop timer */
	pmm_timer_stop();

	/* get timing result */
	pmm_timer_result();

	/* destroy timer */
	pmm_timer_destroy();

	gsl_matrix_free(A);
```

```
        gsl_matrix_free(B);
        gsl_matrix_free(C);

        return PMM_EXIT_SUCCESS;
}
```

4.3 Configuring the Benchmark in PMM

For now, use the example configurations as a template. Small parameter strides should be avoided as they make naive/optimised construction very slow.

GNU Free Documentation License

Version 1.3, 3 November 2008

Preamble

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The

“**Document**”, below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as “**you**”. You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A “**Modified Version**” of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A “**Secondary Section**” is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document’s overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The “**Invariant Sections**” are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The “**Cover Texts**” are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A “**Transparent**” copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not “Transparent” is called “**Opaque**”.

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "**Title Page**" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

The "**publisher**" means any person or entity that distributes copies of the Document to the public.

A section "**Entitled XYZ**" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "**Acknowledgements**", "**Dedications**", "**Endorsements**", or "**History**".) To "**Preserve the Title**" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.

B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.

C. State on the Title page the name of the publisher of the Modified Version, as the publisher.

D. Preserve all the copyright notices of the Document.

E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.

F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.

G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.

H. Include an unaltered copy of this License.

I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.

J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.

L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

M. Delete any section Entitled “Endorsements”. Such a section may not be included in the Modified Version.

N. Do not retitle any existing section to be Entitled “Endorsements” or to conflict in title with any Invariant Section.

O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version’s license notice. These titles must be distinct from any other section titles.

You may add a section Entitled “Endorsements”, provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements".

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled “Acknowledgements”, “Dedications”, or “History”, the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, or distribute it is void, and will automatically terminate your rights under this License.

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, receipt of a copy of some or all of the same material does not give you any rights to use it.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See http://www.gnu.org/copyleft/.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation. If the Document specifies that a proxy can decide which future versions of this License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Document.

11. RELICENSING

"Massive Multiauthor Collaboration Site" (or "MMC Site") means any World Wide Web server that publishes copyrightable works and also provides prominent facilities for anybody to edit those works. A public wiki that anybody can edit is an example of such a server. A "Massive Multiauthor Collaboration" (or "MMC") contained in the site means any set of copyrightable works thus published on the MMC site.

"CC-BY-SA" means the Creative Commons Attribution-Share Alike 3.0 license published by Creative Commons Corporation, a not-for-profit corporation with a principal place of business in San Francisco, California, as well as future copyleft versions of that license published by that same organization.

"Incorporate" means to publish or republish a Document, in whole or in part, as part of another Document.

An MMC is "eligible for relicensing" if it is licensed under this License, and if all works that were first published under this License somewhere other than this MMC, and subsequently incorporated in whole or in part into the MMC, (1) had no cover texts or invariant sections, and (2) were thus incorporated prior to November 1, 2008.

The operator of an MMC Site may republish an MMC contained in the site under CC-BY-SA on the same site at any time before August 1, 2009, provided the MMC is eligible for relicensing.

ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

> Copyright © YEAR YOUR NAME. Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the "with ... Texts." line with this:

> with the Invariant Sections being LIST THEIR TITLES, with the Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.