

DOI: 10.15514/ISPRAS-2024-36(5)-6



Программная среда выполнения методических прикладных тестов для численного исследования параметров высокопроизводительных вычислительных систем

А.О. Игнатьев, ORCID: 0000-0003-4902-2123 <a.o.ignatyev@vniitf.ru>

С.Ю. Мокшин, ORCID: 0000-0002-7454-6597 <s.yu.mokshin@vniitf.ru>

*ФГУП «Российский Федеральный Ядерный Центр –
ВНИИ технической физики имени академика Е.И. Забабахина»,
456770, Россия, г. Снежинск, Челябинская область, ул. Васильева, 13.*

Аннотация. Неотъемлемой частью процесса создания высокопроизводительных вычислительных систем, предназначенных для решения задач численного моделирования различных физических процессов, является проверка их соответствия на заявленные при их проектировании характеристики. В статье рассматривается разработанная авторами программная среда выполнения методических прикладных тестов для численного исследования параметров высокопроизводительных вычислительных систем, позволяющая эффективно анализировать результаты выполнения прикладных тестов и выполнять оценку производительности и надежности вычислительных систем.

Ключевые слова: вычислительная система; высокопроизводительные вычисления; математическое моделирование; тестирование высокопроизводительных вычислительных систем.

Для цитирования: Игнатьев А.О., Мокшин С.Ю. Программная среда выполнения методических прикладных тестов для численного исследования параметров высокопроизводительных вычислительных систем. Труды ИСП РАН, том 36, вып. 5, 2024 г., стр. 81–92. DOI: 10.15514/ISPRAS–2024–36(5)–6.

Runtime Environment for Conducting Methodical Applied Tests to Numerically Investigate of HPC Systems Parameters

A.O. Ignatyev, ORCID: 0000-0003-4902-2123 <a.o.ignatyev@vniitf.ru>
S.Yu. Mokshin, ORCID: 0000-0002-7454-6597 <s.yu.mokshin@vniitf.ru>

*Russian Federal Nuclear Center –
Zababakhin All-Russian Research Institute of Technical Physics,
456770, Russia, Snezhinsk, Chelyabinsk region, Vasilieva street, 13.*

Abstract. An integral part of the process of creating high-performance computing systems designed to solve problems of numerical modeling of various physical processes is to check their compliance with the characteristics stated during their design. The article discusses a script-based runtime environment developed by the authors for using methodical applied tests to numerically investigate parameters of high-performance computing systems. This environment enables efficient analysis of the results of applying tests and provides an assessment of the performance and reliability of high-performance computing systems.

Keywords: computing system; high-performance computing simulation; mathematical modeling; high-performance computing system testing, HPC.

For citation: Ignatyev A.O., Mokshin S.Yu. The environment suite for methodological applied tests for numerical research of HPC systems parameters. *Trudy ISP RAN/Proc. ISP RAS*, vol. 36, issue 5, 2024. pp. 81-92 (in Russian). DOI: 10.15514/ISPRAS-2024-36(5)-6.

1. Введение

При решении задач численного моделирования физических процессов требуется огромный объем вычислений, обеспечиваемых высокопроизводительными вычислительными системами (ВВС). Современные ВВС, предназначенные для решения задач численного моделирования, представляют собой множество вычислительных узлов, систему хранения с параллельным доступом и сервисные подсистемы, объединенных высокопроизводительной коммуникационной средой [1]. При проектировании таких ВВС закладываются ожидаемые характеристики по производительности вычислений, основанные на физических показателях используемых элементов и используемой архитектуре ВВС. Однако, как показывает практика, переход от одной ВВС к другой с большей заявленной производительностью не гарантирует кратного сокращения сроков решаемых задач. Поэтому во всем мире разрабатываются специализированные пакеты тестовых программ [2, 3], реализующих различные численные алгоритмы и позволяющие оценить производительность ВВС применительно к данным алгоритмам. В России такие пакеты разрабатываются на предприятиях Госкорпорации «Росатом» [4, 5].

Процедура проведения испытаний, создаваемых в РФ ВВС узаконена Государственным стандартом [6], в котором методические прикладные тесты (МПТ) используются для определения производительности ВВС, эффективности распараллеливания и проверки технологического процесса счета задач.

Использование МПТ на испытаниях ВВС требует проведения многочисленных серий расчетов с вариацией ряда параметров запуска тестов (объем обрабатываемых тестовой программой данных, размещение процессов задачи на вычислительных узлах, используемые компиляторы и библиотеки и т.д.), поэтому их применение предполагает использование средств автоматизации проведения расчетов и сравнения полученных результатов.

Автоматизированные среды выполнения тестов (тестовые мейнфреймы) давно уже используются при отладке программного обеспечения [7, 8]. В данной статье рассматривается среда выполнения методических прикладных тестов (СВ МПТ), разработанная и успешно применяемая в РФЯЦ-ВНИИТФ. Принципиальным отличием рассматриваемой СВ МПТ является то, что она предназначена для оценки качества наладки

оборудования и получения характеристик производительности и надежности исполнения вычислительных систем, облегчая труд разработчиков и изготовителей ВВС.

2. Описание среды выполнения методических прикладных тестов

2.1 Назначение и область применения

Практически все современные ВВС, предназначенные для решения задач численного моделирования различных физических процессов, имеют однотипную архитектуру: вычислительная подсистема ВВС представляет собой множество многопроцессорных вычислительных узлов (ВУ), объединенных высокоскоростным коммуникационным оборудованием, к которому также подключены система хранения с параллельным доступом, подсистема управления и сервисные подсистемы [1]. Все узлы ВВС работают под управлением операционной системы (ОС) семейства Linux [9]. Для распараллеливания на общей памяти используется стандарт OpenMP [10] или системная библиотека libpthread, для распараллеливания между ВУ используется MPI [11]. Такая архитектура ВВС определяет состав и выбор программных средств, используемых в СВ МПТ.

Состав вычислительной подсистемы ВВС: количество ВУ, характеристики установленных в них процессоров, используемые компиляторы и библиотеки передачи сообщений условно назовем программно-аппаратной архитектурой ВВС. СВ МПТ предполагает наличие описания разных программных архитектур, обеспечивая для них унифицированную подготовку тестовых программ к выполнению, запуску и формированию отчетов в виде таблиц. Тем самым облегчается тестирование ВВС при наладке и проведении испытаний.

2.2 Назначение и область применения

Структура СВ МПТ представлена на рис. 1. Корневой каталог СВ МПТ содержит:

- описания тестируемых программно-аппаратных архитектур (ПА);
- каталоги с МПТ;
- общие программные компоненты, используемые для сборки, запуска МПТ и для вывода полученных результатов, включая функции для генерации последовательностей узлов или потоков, разложения числа процессов по направлениям сетки задачи, запуска задания на счет, привязки процессов к ядрам;
- каталог для выполнения комплексного теста (КТ).

2.3 Описание общих компонент СВ МПТ

2.3.1 Описание программной архитектуры ВВС

Каждая программно-аппаратная архитектура ВВС содержит ряд атрибутов, характеризующих как саму тестируемую ВВС, так и используемое для прогона тестов программное обеспечение. Описание ПА ВВС снабжается идентификатором и включается в общее описание всех тестируемых ПА ВВС.

Описание ПА ВВС включает в себя:

- количество вычислительных узлов (ВУ);
- количество и тип процессоров на ВУ,
- количество ядер в процессоре;
- тактовую частоту процессора;
- пути доступа к компиляторам и библиотекам MPI.

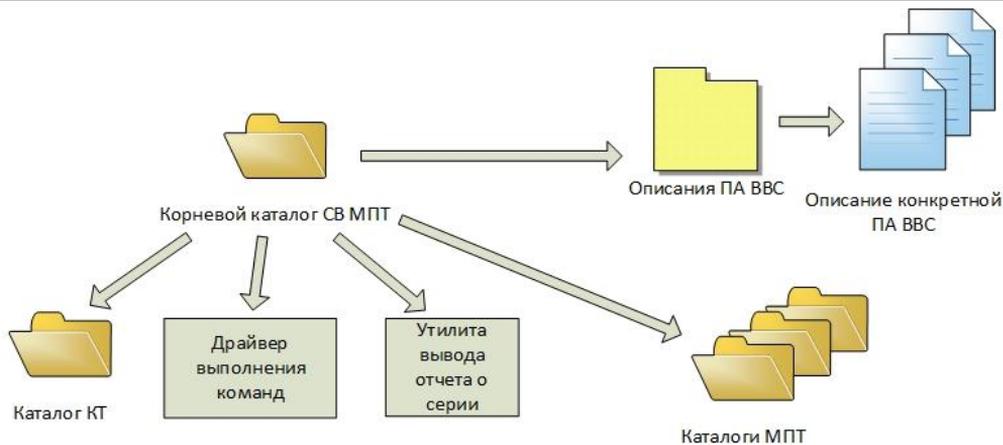


Рис. 1. Структура СВ МПТ.
Fig. 1. Program environment structure.

2.3.2 Каталоги с МПТ

Среда выполнения методических прикладных тестов содержит множество тестовых программ. Каждая тестовая программа располагается в отдельном каталоге и снабжается своим идентификатором.

2.3.3 Запуск задания на выполнение работы

Управляющий модуль (скрипт Execute) предназначен для выполнения работы по запуску других скриптов. В его параметрах указывается:

- идентификатор ПА ВВС;
- идентификатор МПТ;
- вид выполняемой работы (сборка исполнимого кода, запуск серии тестовых расчетов, вывод отчета);
- дополнительные параметры, специфичные для конкретной работы.

Этот скрипт считывает описание указанной в параметрах ПА ВВС, формирует на его основе переменные окружения, используемые при выполнении работы, и, в общем случае, переходит в каталог указанной в параметрах программы, запуская один из скриптов, выполняющих требуемую работу.

2.3.4 Формирование отчета о серии тестов

Утилита вывода отчета о серии тестов (ReportSeria.py) вызывается непосредственно из управляющего модуля. Для выбора отчетных показателей конкретной тестовой программы используется утилита ReportTest.py из каталога МПТ. Утилита просматривает все файлы в каталоге результатов указанной серии и выводит отчет в виде таблицы. В заголовке таблицы указывается название МПТ, используемые для ее сборки компилятор и библиотека MPI, а также опции компиляции. Эта таблица содержит информацию об использованном количестве узлов и ядер, времени выполнения теста, достигнутой производительности, эффективности распараллеливания и признак верификации расчета.

2.4 Описание МПТ

Структура каталога МПТ представлена на рис. 2.

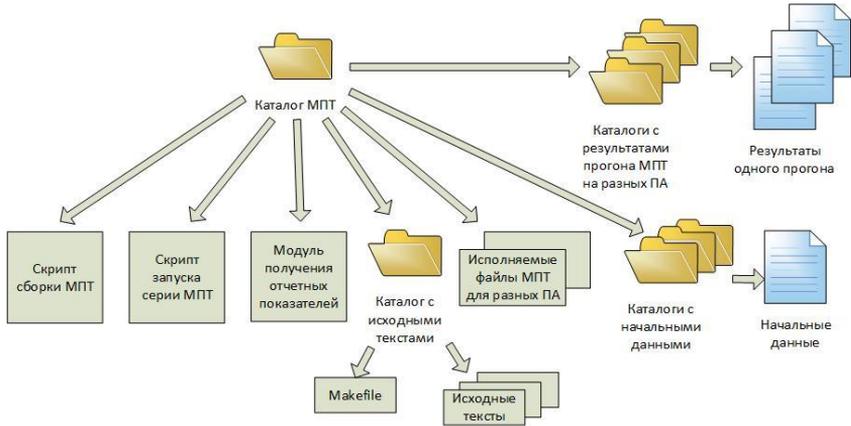


Рис. 2. Структура каталога МПТ.

Fig. 2. Tests directory structure.

Каталог с МПТ содержит:

- подкаталог с исходными текстами тестовой программы и файлом, используемым для сборки программы (makefile);
- скрипт сборки программы под используемую ПА;
- скрипт запуска серии расчетов для используемой ПА;
- модуль получения отчетных показателей для вывода результатов расчета серии;
- исполнимые коды программы для различных ПА;
- каталоги с начальными данными для расчетов серии (исходные данные генерируются автоматически для каждой серии расчетов);
- каталоги, содержащие результаты прогона тестовой программы на выбранных ПА ВВС.

2.4.1 Скрипт сборки МПТ

Скрипт `Compile` вызывается из управляющего модуля и используется для компиляции тестовой программы из исходных текстов и сборки исполнимого файла программы. Перед вызовом этого скрипта производится считывание описания указанной ПА ВВС и формируются переменные окружения, указывающие пути доступа к компиляторам и библиотекам, необходимым для сборки программы, а также опции компиляции. Полученный в процессе сборки исполняемый код программы помечается идентификатором ПА и, вместе с перечнем используемых при компиляции и линковке опций, помещается в каталог тестовой программы.

2.4.2 Скрипт запуска серии расчетов

Скрипт `RunSerial` вызывается из управляющего модуля и выполняет запуск серии расчетов одного из двух типов:

- распараллеливание внутри ВУ методом деления с использованием `OpenMP`;
- распараллеливание на множестве ВУ методом умножения с использованием коммуникационной библиотеки `MPI`.

Распараллеливание внутри ВУ методом деления выполняется на разном числе ядер, указанном в описании ПА ВВС. Как правило, это значения 1, 2, 4 и до максимально возможного значения.

Распараллеливание на множестве ВУ выполняется на разном числе ВУ. Для большинства тестов это число изменяется по степеням двойки от 1 до предельного числа ВУ, указанного в описании ПА ВВС, хотя есть тесты, в которых задается другой алгоритм изменения числа используемых ВУ.

Тип серии расчетов указывается в параметре скрипта. Для распараллеливания на множестве ВУ также может быть указано число MPI-процессов на одном ВУ (по умолчанию – один). Число используемых ядер для MPI-процесса задачи рассчитывается исходя из полного числа ядер на ВУ и числа MPI-процессов на ВУ, обеспечивая тем самым полную загрузку всего ВУ. Для каждого из расчетов в серии формируются необходимые начальные данные, после этого задание на расчет запускается с помощью системы управления задачами (используется менеджер ресурсов SLURM) [12]. Для указанной серии расчетов создается отдельный каталог в каталоге тестовой программы, в который помещаются все файлы с полученными результатами.

2.4.3 Модуль получения отчетных показателей

Модуль получения отчетных показателей состоит из утилиты ReportTest.py и вызывается из другой утилиты ReportSesia.py, при этом для каждого выводного файла серии выбираются соответствующие отчетные показатели.

2.5 Каталог комплексного теста

Комплексный тест (КТ) предназначен для проверки технологического процесса счета задач и согласно [6] должен обеспечивать непрерывное выполнение большого количества МПТ в течение длительного времени. Во время комплексного теста проверяется как будет работать ВВС в условиях, близких к реальным. Для прохождения КТ используются МПТ из состава СВ МПТ и программа – генератор заданий. Структура каталога КТ представлена на рис. 3.

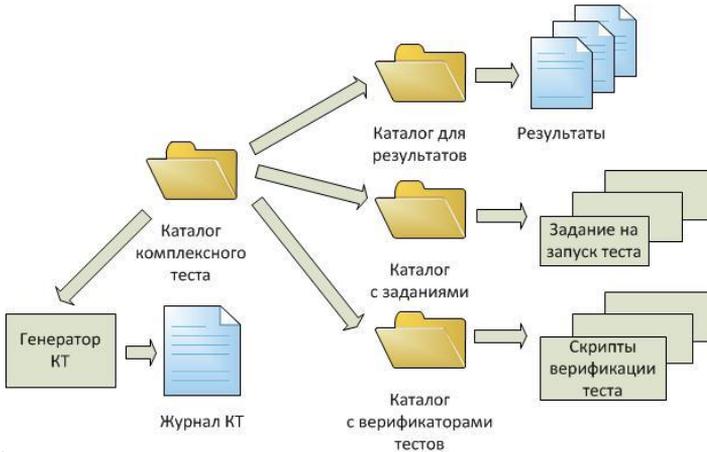


Рис. 3. Структура каталога КТ.
Fig. 3. Structure of the complex test directory.

Задания, используемые в КТ, содержатся в отдельном подкаталоге и представляют собой скрипты, содержащие обращения к управляющему модулю СВ МПТ с параметрами, определяющими запуск одной конкретной тестовой программы на заданном числе ВУ. Каждому заданию соответствует свой скрипт верификации результатов.

Генератору заданий в параметрах указывается время выполнения КТ (обычно, не менее 8 часов). В процессе работы генератор в течение заданного времени генерирует и запускает задачи из каталога заданий, отслеживает их завершение и анализирует полученные

результаты. По окончании работы формируется отчет, содержащий времена начала и завершения КТ, информацию о запускаемых и завершаемых задачах, общее количество запущенных задач, количество задач завершаемых успешно и с ошибками, а также разброс времен выполнения задач каждого типа.

3. Использование СВ МПТ

3.1 Формирование ПА ВВС

Множество ПА ВВС описывается в файле TstEnv в корневом каталоге СВ МПТ. Этот файл является shell-скриптом. Выбор конкретной ПА производится оператором case по идентификатору ПА. Поэтому описание ПА сводится к формированию строк вида, приведенным на листинге 1.

```
1 case $ARCH in
2   GNU11_MV237)
3     # Описание оборудования
4     export VNODE=4608
5     export SOCKETS=2
6     export CORpSOC=64
7     export THRpCOR=1
8     export CORES=$(( $SOCKETS * $CORpSOC * $THRpCOR )
9     export NTMAX=128
10    export NHMAX=64
11    # Используемое для сборки ПО и опции
12    module load mpi/gnu
13    export MPIF90=mpifort
14    export MPICXX=mpicxx
15    module load gcc/11.2.0
16    FFOPTS="-ffree-line-length-none"
17    export FFLAGS="-fopenmp -O3 -I${MPI}/include $FFOPTS"
18    export CFLAGS="-fopenmp -O3 -I${MPI}/include"
19    # Переменные окружения при запуске тестов
20    export PART=VKADMIN
21    export TST_AFFINITY=GOMP_CPU_AFFINITY
22    ;;
23    ICF_MV237)
24    ...
30    module load mpi/intel
31
32    ...
40    module load intel/2020
41
42    ...
50    ;;
51    *) BREAK No such arch $ARCH
52 esac
```

*Листинг 1. Пример описания программных архитектур.
Listing 1. Program architecture description example.*

В приведенном примере описаны две ПА, используемые при тестировании, описывающие ВВС из 64 ВУ, содержащих по 2 процессора. Первая ПА предназначена для программ, собираемых с использованием компиляторов семейства GNU, вторая – Intel.

Описание ПА состоит из трех частей:

- описание используемых ВУ;
- описание используемого программного обеспечения;

- особенности исполнения (партиция или резервация системы управления задачами, тип привязки процессов к ядрам процессоров и т.д.).

В описании второй ПА приведены только строки, отличающиеся от первой. В начале управляющего модуля устанавливается соответствие первого параметра описанным ПА, как показано на листинге 2.

```
1 case $1 in
2   -alg) ARCH=GNU11_MV235
3     shift
4     ;;
5   -ali) ARCH=ICF_MV235
6     shift
7     ;;
8   -a2) ...
9     ;;
10  *)
11 esac
```

*Листинг 2. Пример выбора тестируемой ПА в управляющем модуле.
Listing 2. Example of selecting the software architecture in the control module.*

3.2 Компиляция тестовой программы

Компиляция тестовой программы на выбранную ПА (параметр `-alg` определяет ПА с идентификатором GNU11_MV235) осуществляется следующим образом:

```
# ./Execute -alg <test_name>
```

Компиляция всех тестовых программ может быть выполнена командой:

```
# ./Execute -alg all
```

3.3 Запуск серии расчетов

ГОСТ [6] предписывает проведение двух серий расчетов на определение эффективности распараллеливания для каждой МПТ:

- распараллеливание на общей памяти ВУ средствами OpenMP или `libpthread`;
- распараллеливание на распределенной памяти множества узлов средствами MPI.

В первом случае запускается серия расчетов с использованием одинакового объема пространства данных задачи, вся вычислительная нагрузка распределяется между потоками выполнения программы (метод деления). Серия запускается на одном ВУ и разном числе потоков от 1 до максимально возможного (параметр NTMAX в описании ПА).

Запуск выполняется командой:

```
# ./Execute -alg <test_name> r1
```

Во втором случае запускается серия расчетов на разном числе узлов от 1 до максимально возможного (параметр NHMAX в описании ПА) с увеличением числа ВУ по степеням двойки (или иным, предусмотренном в тестовой программе способом).

Запуск выполняется командой:

```
# ./Execute -alg <test_name> rN
```

где N – число MPI-процессов на ВУ (1,2,4 или 8).

При запуске серии в каталоге тестовой программы создается подкаталог, помеченный идентификатором ПА, в который будут записаны файлы результатов расчетов.

3.4 Вывод результатов серии расчетов

Вывод результатов серии на одном ВУ производится командой:

```
# ./Execute -alg <test_name> R1
```

В результате исполнения получается табличный файл вида, представленного на рис. 4.

```
=== sPPM GNU11MV235 ===
/home/s4145/TestsRosatom/SPPM-OMP-MPI
MPIF90=/opt/mpi/mvapich2-2.3.5/gcc/10.2.0/bin/mpifort
FFLAGS=-fopenmp -O3 -I/opt/mpi/mvapich2-2.3.5/gcc/10.2.0/include -ffree-
line-length-none -fdefault-real-8 -fdefault-double-8 -std=legacy -
fopenmp
mpifort for MVAPICH2 version 2.3.5 gcc version 11.2.0 (GCC)
/home/s4145/TestsRosatom/SPPM-OMP-MPI
One Node Serial
NP  DProc  T          Esp Mem/Proc(MB) Ver V(Gflops)  Ecp  Size(cell)
1    1x1x001 5.899e+01  100.00 8.493e+02  PASS 4.895e+00  13.60
192x192x192
1    1x1x004 1.486e+01   99.26 8.493e+02  PASS 1.944e+01  13.50
192x192x192
1    1x1x008 7.676e+00   96.07 8.493e+02  PASS 3.762e+01  13.06
192x192x192
1    1x1x016 3.967e+00   92.94 8.493e+02  PASS 7.280e+01  12.64
192x192x192
1    1x1x032 2.078e+00   88.72 8.493e+02  PASS 1.390e+02  12.06
192x192x192
1    1x1x064 1.263e+00   72.98 8.493e+02  PASS 2.286e+02  9.92
192x192x192
1    1x1x128 1.532e+00   30.08 8.493e+02  PASS 1.885e+02  4.09
192x192x192
```

Рис. 4. Таблица результатов одноузловой серии расчетов.

Fig. 4. Single-node calculation results.

В заголовке этого табличного файла приведены имя тестовой программы, идентификатор ПА, используемые компиляторы и опции компиляции. В столбцах таблицы приводятся:

- число процессов MPI;
- конфигурация распараллеливания запуска в виде <число ВУ>×<число MPI-процессов на ВУ>×<число потоков на MPI-процессе>;
- время выполнения;
- объем используемой на одном процессе MPI оперативной памяти;
- признак верификации результатов;
- производительность в гигафлопах (если программа умеет ее рассчитывать);
- эффективность использования процессора (в случае, если тестовая программа умеет рассчитывать производительность в операциях в секунду);
- размер задачи в ячейках.

Эффективность использования процессора рассчитывается как отношение полученной на тесте производительности к пиковой производительности ВУ (в процентах).

Вывод результатов многоузловой серии производится командой:

```
# ./Execute-alg <test_name> RS
```

В результате получается табличный файл, содержимое которого представлено на рис. 5.

Обозначения в табличном виде те же самые.

3.5 Проверка технологического процесса счета задач

Проверка выполнения технологического процесса счета задач выполняется путем запуска комплексного теста. Традиционно в состав КТ входят задачи, разработанные в РФЯЦ-ВНИИТФ и РФЯЦ-ВНИИЭФ, охватывающие основные этапы численного моделирования, производимого в этих институтах, а также три задачи из NAS Parallel Benchmark (Block Tri-

diagonal solver, Lower-Upper Gauss-Seidel solver, calar Penta-diagonal solver). Параметры тестовых задач подобраны так, чтобы время счета каждой составляло не более нескольких минут. Таким образом, за время выполнения КТ (обычно, не менее 8 часов) просчитывается несколько тысяч задач, создавая нагрузку как на оборудование ВВС, так и на компоненты операционной системы.

```

=== sPPM GNU11MV235 ===
/home/4145/TestsRosatom/SPPM-OMP-MPI
MPIF90=/opt/mpi/mvapich2-2.3.5/gcc/10.2.0/bin/mpifort
FFLAGS=-fopenmp -O3 -I/opt/mpi/mvapich2-2.3.5/gcc/10.2.0/include -ffree-
line-length-none -fdefault-real-8 -fdefault-double-8 -std=legacy -
fopenmp
mpifort for MVAPICH2 version 2.3.5 gcc version 11.2.0 (GCC)
/home/s4145/TestsRosatom/SPPM-OMP-MPI
Scalable Nodes Serial
NP   DProc   T           Esp   Mem/Proc(MB) Ver V(Gflops)   Ecp
Size(cell)
 4   1x4x032  2.422e+00   100.00 8.493e+02   PASS 4.769e+02   10.35
192x192x192
 8   2x4x032  2.514e+00   96.34 8.493e+02   PASS 9.189e+02   9.97
192x192x192
16   4x4x032  2.526e+00   95.88 8.493e+02   PASS 1.829e+03   9.92
192x192x192
32   8x4x032  2.595e+00   93.33 8.493e+02   PASS 3.561e+03   9.66
192x192x192
64  16x4x032 2.650e+00   91.40 8.493e+02   PASS 6.974e+03   9.46
192x192x192
128 32x4x032 2.663e+00   90.95 8.493e+02   PASS 1.388e+04   9.41
192x192x192
256 64x4x032 2.667e+00   90.81 8.493e+02   PASS 2.772e+04   9.40
192x192x192
    
```

Рис. 5. Таблица результатов многоузловой серии расчетов.

Fig. 5. Multi-node calculation results.

Он содержит следующую информацию:

- дата и время начала и конца выполнения теста;
- информацию о запуске и завершении задач;
- результат проверки правильности расчета каждой задачи;
- общее количество запущенных задач, полное количество завершенных задач и количество задач, завершенных с ошибками или аварийно.

При завершении КТ для каждого типа задач выдается:

- имя задачи (test);
- количество запущенных задач этого типа (start);
- количество завершенных задач (fin);
- количество успешно завершенных задач (ver);
- минимально, максимальное и среднее времена выполнения успешно завершенных задач этого типа (tmin, tmax, tave);
- величины разброса между минимальным и максимальным (dtM), а также между минимальным и средним временами выполнения (dtA) в процентах.

Последние значения позволяют оценить стабильность работы оборудования ВВС в условиях многозадачного режима.

Исполнимые коды МПТ берутся те же, что и при проведении серий расчетов на определение эффективности распараллеливания.

Запуск КТ производится скриптом RunAutoLauncher, в котором указывается необходимое время выполнения теста.

Журнал результатов работы КТ имеет вид, представленный на рис. 6.

```
2023-11-22 16:04:47 s4145@f1:/home/s4145/TestsRosatom/Complex pid=51531
запущен на 8.0 часов из Tasks
2023-11-22 16:04:47 В очереди 0 задач, в счете 0. Всего запущено 0,
завершено 0, с ошибками 0
Запуск md-16 Submitted batch job 322730
...
Завершено хорошо mht-16-336700.out , переносится в GOOD
2023-11-23 00:10:44 В очереди 0 задач, в счете 0. Всего запущено 13957,
завершено 13957, с ошибками 0 Не запускаем
2023-11-23 00-11-04 Завершение работы, время выполнения 4.86e+02 минут
-----
task      start  fin    ver  tmin    tmax    tave    dtM%    dtA%
lu-512    821    821    821  4.651e+01 5.572e+01 4.717e+01 19.80    1.43
bt-225    821    821    821  1.230e+02 1.356e+02 1.246e+02 10.25    1.32
md-16     821    821    821  4.775e+00 5.446e+00 4.877e+00 14.05    2.14
bt-1089   821    821    821  3.086e+01 4.291e+01 3.431e+01 39.05    11.18
bt-625    821    821    821  5.098e+01 6.178e+01 5.206e+01 21.18    2.12
egida-16  821    821    821  1.994e+00 2.336e+00 2.084e+00 17.14    4.48
lu-256    821    821    821  8.654e+01 9.882e+01 8.775e+01 14.19    1.40
slon-16   821    821    821  2.135e+01 2.455e+01 2.163e+01 14.96    1.32
sprm-16   821    821    821  1.567e+00 2.759e+00 1.662e+00 76.07    6.04
lu-1024   821    821    821  2.461e+01 3.365e+01 2.530e+01 36.73    2.81
sp-625    821    821    821  6.622e+01 8.664e+01 6.784e+01 30.84    2.44
sp-225    821    821    821  1.711e+02 2.116e+02 1.740e+02 23.70    1.70
hctest-16 821    821    821  1.581e+02 1.838e+02 1.638e+02 16.26    3.63
sp-1089   821    821    821  3.902e+01 5.431e+01 4.096e+01 39.19    4.98
pauk-16   821    821    821  6.556e+01 8.369e+01 6.693e+01 27.64    2.09
mht-16    821    821    821  2.802e+02 2.845e+02 2.831e+02 1.51     1.03
machete-8 821    821    821  1.689e+02 1.898e+02 1.729e+02 12.41    2.36
```

Рис. 6. Пример журнала результатов работы комплексного теста
Fig. 6. Complex test results

4. Заключение

Представленная в статье программная среда выполнения методических прикладных тестов, разработанная в РФЯЦ-ВНИИТФ более трех лет назад, позволила автоматизировать процесс тестирования ВВС при их наладке и проведении испытаний, и тем самым значительно сократить сроки проведения этих работ.

СВ МПТ полностью учитывает требования ГОСТ по проведению приемочных испытаний и успешно применяется на создаваемых РФЯЦ-ВНИИТФ ВВС.

Авторы выражают надежду, что изложенная в статье информация о среде выполнения методических прикладных тестов окажется полезной другим специалистам, занимающимся разработкой ВВС.

Список литературы/References

- [1]. Игнатьев А.О., Мокшин С.Ю. Типовая архитектура высокопроизводительной вычислительной системы для решения задач численного моделирования, Препринт РФЯЦ-ВНИИТФ № 265, Снежинск, 2020 г., 21 с.
- [2]. NAS Parallel Benchmarks. Available at: <https://www.nas.nasa.gov/software/npb.html>, accessed 07.07.2024.
- [3]. Benchmarks and Performance Analysis. Available at <https://www.lanl.gov/projects/crossroads/benchmarks-performance-analysis.php>, accessed 19.07.2024.
- [4]. Алексеев А.В., Беляев С.П., Бочков А.И. и др. Методические прикладные тесты РФЯЦ-ВНИИТФ для численного исследования параметров высокопроизводительных вычислительных систем. ВАНТ, сер. Математическое моделирование физических процессов. 2020. Вып. 2., с. 86-100.

- [5]. Игнатъев А.О., Мокшин С.Ю., Романова Е.М. и др. Набор методических тестовых программ для численного исследования параметров высокопроизводительных вычислительных систем. Труды ИСП РАН, том 36, вып. 2, 2024 г., с. 109-126.
- [6]. ГОСТ Р 55700.26-2020. Высокопроизводительные вычислительные системы. Требования приемочных испытаний.
- [7]. Linux Test Project. Available at <http://github.com/Linux-test-project/releases/latest>, accessed 18.07.2024.
- [8]. Robot Framework. Available at <http://robotframework.org>, accessed 18.07.2024.
- [9]. «СПО Супер-ЭВМ». Available at <http://vniitf.ru/article/spo-super-evm>, accessed 18.05.2024.
- [10]. OpenMP. Available at: <https://www.openmp.org/>, accessed 07.05.2024.
- [11]. Gropp W., Lusk E., Skjellum A. Using MPI: Portable parallel programming with the message-passing interface. – Second edition. – The MIT Press, 1999.
- [12]. Slurm workload manager, Available at: <https://slurm.schedmd.com/documentation.html>, accessed 08.07.2024.

Информация об авторах / Information about authors

Алексей Олегович ИГНАТЬЕВ – начальник лаборатории Федерального государственного унитарного предприятия «Российский Федеральный Ядерный Центр – Всероссийский научно-исследовательский институт технической физики имени академика Е.И. Забабахина» с 1998 года. Сфера научных интересов: проектирование вычислительных систем, разработка параллельных программ численного моделирования, разработка операционных систем, методы и средства защиты информации.

Alexey Olegovich IGNATYEV – Head of the Laboratory of Russian Federal Nuclear Center E. I. Zababakhin «All-Russian Scientific Research Institute of Technical Physics» since 1998. Research interests: design of supercomputer systems, parallel numerical simulation programs development, operating systems development, methods and means of information security.

Сергей Юрьевич МОКШИН – начальник отдела Федерального государственного унитарного предприятия «Российский Федеральный Ядерный Центр – Всероссийский научно-исследовательский институт технической физики имени академика Е.И. Забабахина» с 2016 года. Сфера научных интересов: проектирование вычислительных систем, разработка функциональных подсистем для высокопроизводительных вычислительных систем, разработка операционных систем, методы и средства защиты информации.

Sergey Yurievich MOKSHIN – Head of the Department of Russian Federal Nuclear Center E. I. Zababakhin «All-Russian Scientific Research Institute of Technical Physics» since 2016. Research interests: design of supercomputer systems, development of functional subsystems for high performance supercomputing systems, operating systems development, methods and means for protecting information.