

Формат описания грид-задач на языке JSON

Атрибуты описания задач

Значения некоторых параметров описания задачи содержат относительные пути к файлам. Данные пути разрешаются относительно директории, в которой производится запуск задачи на вычислительном узле (эта директория создается автоматически).

Описание задачи не может иметь атрибутов, отличных от перечисленных ниже.

- `version`, число - версия используемой схемы описания задачи. Версия данной схемы 2.
- `description`, строка, опциональный параметр - описание задачи в произвольной форме, для понимания человеком.
- `executable`, строка - относительный или абсолютный путь выполняемого файла задачи. Если необходимо выполнить программу, выполняемый файл которой находится на удаленном ресурсе, необходимо для запуска передать ее в `input_files`, и она должна иметь атрибут `executable` на удаленном ресурсе.
- `arguments`, список строк, опциональный параметр - аргументы командной строки, которые будут переданы программе при запуске.
- `environment`, объект, опциональный параметр - дополнительные переменные среды окружения, которые будут установлены перед запуском задачи. Атрибуты данного объекта являются названиями переменных окружения (и будут переведены в верхний регистр), значения задают значения переменных.

Пример описания задачи, использующего этот параметр:

```
{ "version": 2,
  "executable": "/usr/bin/env",
  "environment":
  { "FOO": "bar",
    "qux": "XyZzy" }
}
```

При запуске такой задачи будут установлены переменные FOO (значение bar) и QUX (значение XyZzy).

- `count`, число, опциональный параметр - количество процессоров, на которых будет запущена задача. Значение этого атрибута более 1 означает, что задача является MPI-задачей. Любые другие значения или отсутствие атрибута означают, что задача не является MPI-задачей.
- `input_files`, объект, опциональный параметр, `output_files`, объект, опциональный параметр - в данных параметрах указываются файлы, которые передаются перед запуском и после завершения задачи. Названия атрибутов объектов являются относительными или абсолютными путями файлов на вычислительном узле. Соответствующие им значения могут быть либо URL'ами соответствующих файлов на удаленных узлах, либо относительными или абсолютными путями, разрешаемыми относительно

default_storage_base. Если в описании задачи параметр default_storage_base не указан, то будет использоваться значение этого параметра из описания задания. В случае, если этот параметр указан и в описании задачи, и в описании задания, то значение, указанное в описании задачи имеет приоритет над значением, указанным в описании задания. Если параметр default_storage_base отсутствует как в описании задачи, так и в описании задания, то файлы, для которых будет указан путь, а не URL, будут проигнорированы.

Пример описания задания с несколькими задачами и входными/выходными файлами:

```
{ "version": 2,
  "default_storage_base": "gsiftp://example.org/my/files/",
  "tasks": [
    { "id": "a",
      "definition":
      { "version": 2,
        "executable": "/bin/cp",
        "arguments": ["hello.txt", "qux/test.txt"],
        "input_files":
        { "hello.txt": "hello.txt",
          "foo.txt": "/bar.txt",
          "qux": "gsiftp://example.org/my/directory/qux/"
        },
        "output_files":
        { "qux/test.txt": "gsiftp://example.org/my/output/117/test.txt"
        }
      }
    },
    { "id": "b",
      "definition":
      { "version": 2,
        "executable": "/bin/cat",
        "arguments": ["hello.txt", "foo.txt"],
        "default_storage_base": "gsiftp://example.org/other/files/",
        "input_files":
        { "hello.txt": "hello.txt",
          "foo.txt": "/bar.txt"
        }
      }
    }
  ]
}
```

- stdin, строка, опциональный параметр, stdout, строка, опциональный параметр, stderr, строка, опциональный параметр - URL, либо либо относительно или абсолютными путями, разрешаемый относительно default_storage_base для стандартных потоков ввода/вывода задачи.
- default_storage_base, строка, опциональный параметр - URI, относительно которого разрешаются пути к удаленным файлам в соответствующих параметрах
- max_success_code, число, опциональный параметр - максимальное значение кода выхода программы, при котором завершение считается нормальным (не аварийным). Код выхода рассматривается как беззнаковое целое. Если данный параметр не указан, используется

значение по умолчанию, равное 0.

- requirements, объект, опциональный параметр - объект требований к ресурсам (согласно схеме описания требований к ресурсам). Заданные значения из параметра requirements задачи замещают соответствующие значения требований задания.

Например, приведенные ниже фрагменты задают задачи с одинаковыми требованиями к ресурсам:

```
{ "version": 2,
  "tasks": [
    { "id": "a",
      "executable": "/bin/hostname",
      "requirements": {
        "queue": "long"
      }
    }
  ],
  "requirements": {
    "lrms": "Cleo"
  }
}

{ "version": 2,
  "tasks": [
    { "id": "a",
      "executable": "/bin/hostname",
      "requirements": {
        "lrms": "Cleo",
        "queue": "long"
      }
    }
  ]
}
```

- meta, объект, опциональный - данный атрибут предназначен для хранения любых пользовательских параметров задачи, и может иметь любое содержимое.

Описание требований к ресурсам

Результирующее требование задачи к ресурсам получается путем обновления требований к ресурсам всего задания требованиями к ресурсам выбранной задачи.

Все параметры в описании требований к ресурсам являются опциональными. Если это не оговорено специально, то значения по умолчанию для каждого из параметров обеспечивают выбор любого ресурса, удовлетворяющего остальным требованиям.

- hostname, список строк, опциональный параметр - список допустимых имен хостов, на которых может выполняться задача.
- lrms, строка, опциональный параметр - тип подходящей batch-системы (например: Fork, PBS, Cleo).

- fork, boolean, опциональный параметр - разрешить использовать lrms Fork в процессе выбора ресурсов. По умолчанию, если данный параметр не указан, или имеет значение false, Fork будет использоваться только в том случае, если он явно выбран как тип lrms в параметре lrms.
- queue, строка, опциональный параметр - название очереди batch-системы, в которой должна выполняться задача.

JSON Schema описания задач

```
{ "description": "Отдельная задача",
  "type": "object",
  "properties":
  { "created": { "type": "string", "format": "date-time" },
    "modified": { "type": "string", "format": "date-time", "optional": true
  },
  "job": { "type": "string", "format": "uri",
    "description": "URI задания, к которому относится данная задача" },
  "definition": { "type": "string",
    "description": "описание задачи на языке описания задач",
    "format": "application/octet-stream"
  },
  "state": { "type": "array",
    "description": "Состояние задачи, со всей историей его изменений",
    "items": { "type": "object",
      "description": "Запись о состоянии задачи.",
      "properties":
      { "s": { "type": "string",
        "description": "состояние",
        "enum": [ "new", "pending", "running",
"paused", "finished", "aborted" ] },
        "ts": { "type": "string",
          "format": "date-time",
          "description": "время, когда наступило
данное состояние" }
      },
      "additionalProperties": true
    }
  },
  "additionalProperties": false
}
```

From:
<https://theory.sinp.msu.ru/> - THEORY

Permanent link:
<https://theory.sinp.msu.ru/doku.php/dcomp/fcp/rest/taskschema>

Last update: 26/01/2017 12:08



