Факультет
компьютерных наук

Петренко Ксения
Курочкин Илья

Москва
2024

Simulation of Volunteer Computing in a Desktop Grid System

BOINC

Вычислительная мощность

- Frontier: 1.206 ExaFLOPS
- Fugaku: 442 PetaFLOPS
- Summit: 200 PetaFLOPS
- Sierra: 125 PetaFLOPS
- Perlmutter: 70.9 PetaFLOPS
- BOINC: 14.5 PetaFLOPS

Проекты, использующие BOINC

Project name	Users	last day	Hosts	last day	Teams	last day	Countries
BOINC combined	4,034,014	11	165,281	177	112,021	-1	278
GPUGRID	47,897	1	108,797	6	1,846	0	177
Einstein@Home	493,751	11	2,073,556	104	12,067	0	226
PrimeGrid	354,839	4	851,087	155	3,307	0	207
Amicable Numbers	17,048	7	8,825	9	533	0	175
Moo! Wrapper	64,269	3	537,222	8	853	0	161
SRBase	3,020	3	96,286	40	235	0	103
Gerasim@Home	7,380	5	12,100	4	385	0	103
NumberFields@home	14,814	7	2,694,962	30	822	0	142
Rosetta@Home	1,386,953	19	4,545,703	74	12,687	0	229
Asteroids@home	153,064	10	361,056	72	2,109	0	211
MilkyWay@home	257,800	13	693,430	72	4,789	0	221
Universe@Home	55,298	0	455,299	0	995	0	173
NFS@Home	19,835	3	318,694	26	993	0	169
SiDock@home	9,172	1	40,328	24	291	0	127

Мотивация и актуальность

- Систему нужно тестировать и развивать.
- Для масштабной системы как BOINC это можно делать с помощью симуляторов и эмуляторов.
- Для симулятора важны легкость его понимания и изменения, корректность работы, полнота симуляции.

Моделирование

	Серверная часть	Клиентская часть	Код доступен	Вид
ComBoS[1]	проекты	кластеры	да	симулятор
SimBOINC	-	главный фокус	нет	симулятор
SimBA	главный фокус	трейсы	нет	симулятор
EmBoinc[2]	код BOINC	трейсы	да	эмулятор

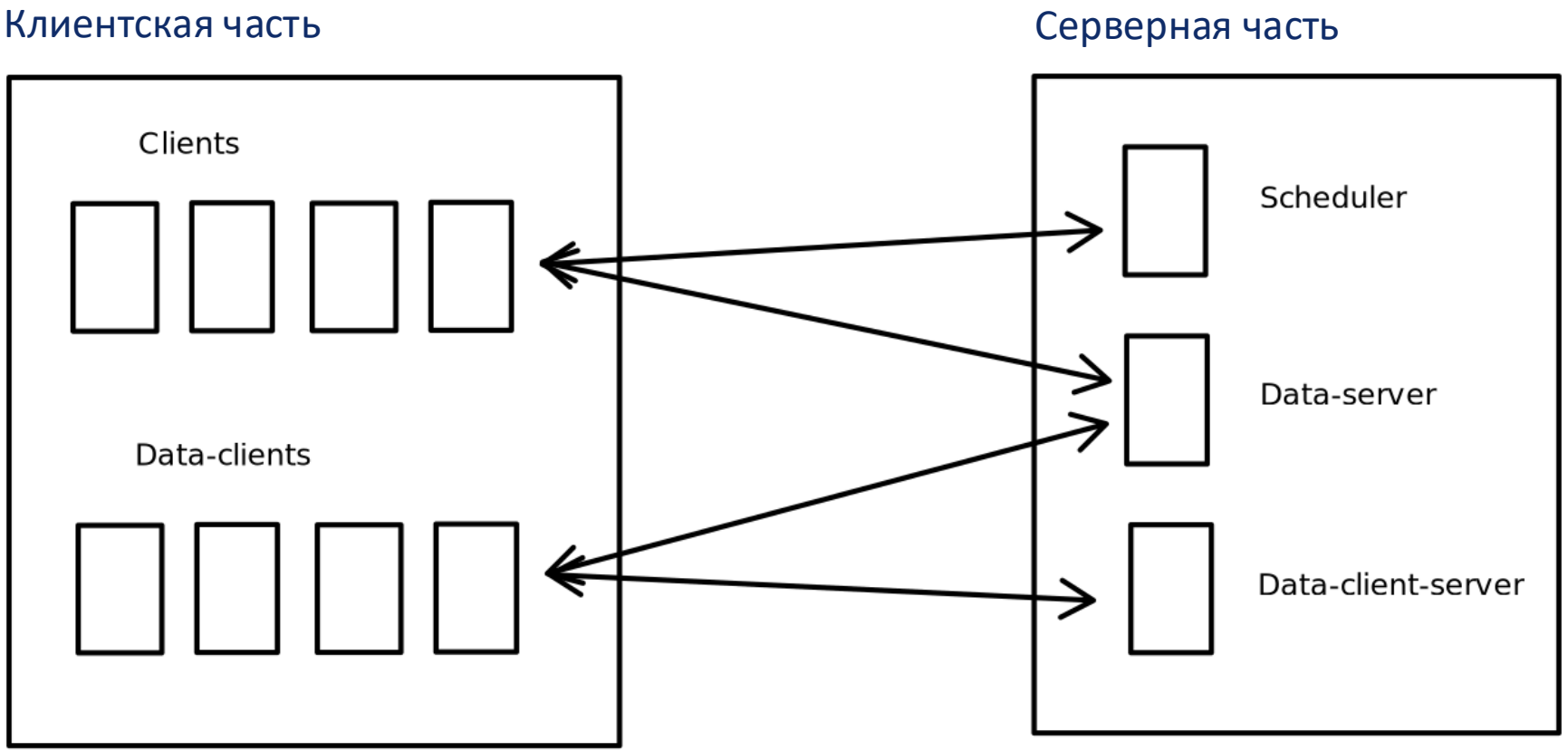
1. https://github.com/arcos-combos/combos/tree/master
2. https://boinc.berkeley.edu/trac/wiki/EmBoinc

Актуальность

- Разумно выбрать ComBoS.
- При его запуске возникли трудности, поэтому код пришлось исправлять.
- Старая и новая версии выдавали разные результаты, поэтому симулятор нужно было верифицировать.
- Запуск экспериментов может помочь выявить, что не хватает симулятору.

https://github.com/Ksenia-C/combos/tree/master

ComBoS


Клиентская часть
Серверная часть
Clients
Data-clients
Scheduler
Data-server
Data-client-server

Запуск ComBoS

C boinc_simulator.c 2

```
3570  int main(int argc, char *argv[])
3837      xbt_free(_pdatabase);
3838      xbt_free(_sserver_info);
3839      xbt_free(_dserver_info);
3840      xbt_free(_dcserver_info);
3841      xbt_free(_dclient_info);
3842      xbt_free(_group_info);
3843      xbt_mutex_destroy(_oclient_mutex);
3844      xbt_mutex_destroy(_dclient_mutex);
3845      xbt_dict_free(&_sscomm);
3846      xbt_dict_free(&_dscomm);
3847
3848      if (res == MSG_OK)
3849          return 0;
3850      else
3851          return 1;
3852  }
3853
```

Весь код в одном файле 4к LoC

```
server_name = workunit->input_files[i];

// BORRAR (esta mal, no generico)
if(i < database->dcreplication){
    int server_number = atoi(server_name+2) - NUMBER_ORDINARY_CLIENTS;

    // printf("resto: %d, server_name: %s, server_number: %d\n", NUMBER_ORDINARY_CLIEN
    //printf("%d\n", _dclient_info[server_number].working);
    if(_dclient_info[server_number].working == 0) continue;
}

dsinput_file_request = xbt_new0(s_dsmessage_t, 1);
dsinput_file_request->type = REQUEST;
dsinput_file_request->answer_mailbox = mailbox;
dsinput_file_request_task = MSG_task_create("ask_work", 0, KB, dsinput_file_request);
server_name = workunit->input_files[i];
MSG_task_send(dsinput_file_request_task, server_name);  // Send input file request
```

Асинхронный код без пояснений

```
46
47-     xbt_mutex_acquire(database->r_mutex);
48
49      while (database->ncurrent_workunits >= MAX_BUFFER && !datab
50-         xbt_cond_wait(database->wg_full, database->r_mutex);
51
52      if (database->wg_end)
```

```
1038
1039+       // ksenia - this is strange - double lock. probably meant
1040+       std::unique_lock lock(*(database->w_mutex));
1041
1042        while (database->ncurrent_workunits >= MAX_BUFFER && !datab
1043+         database->wg_full->wait(lock);
1044
1045        if (database->wg_end)
```

UB в коде

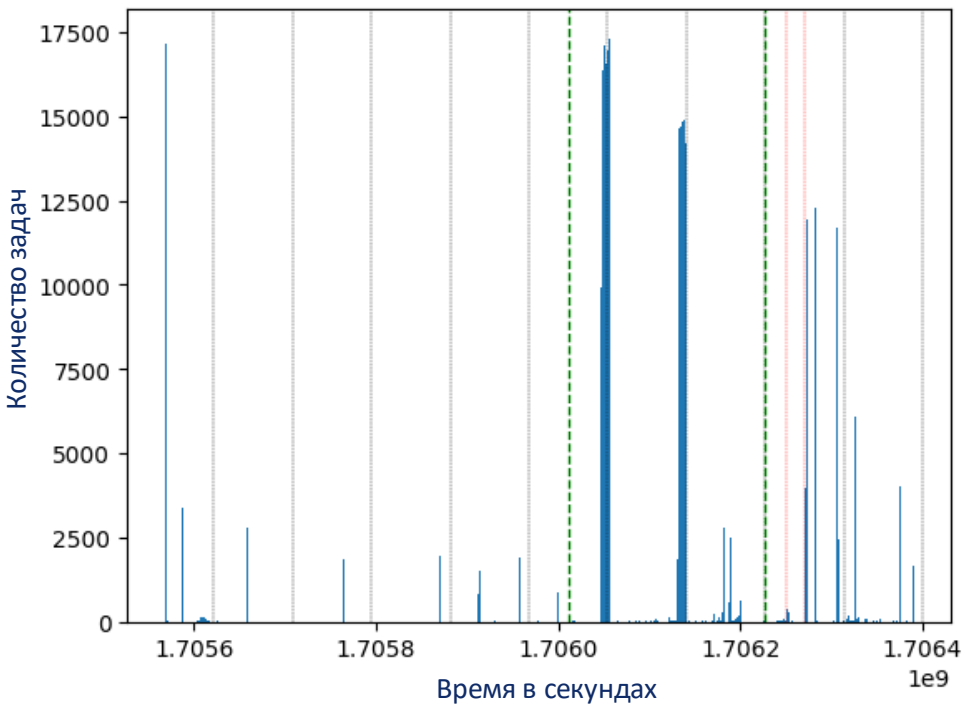
Запуск ComBoS

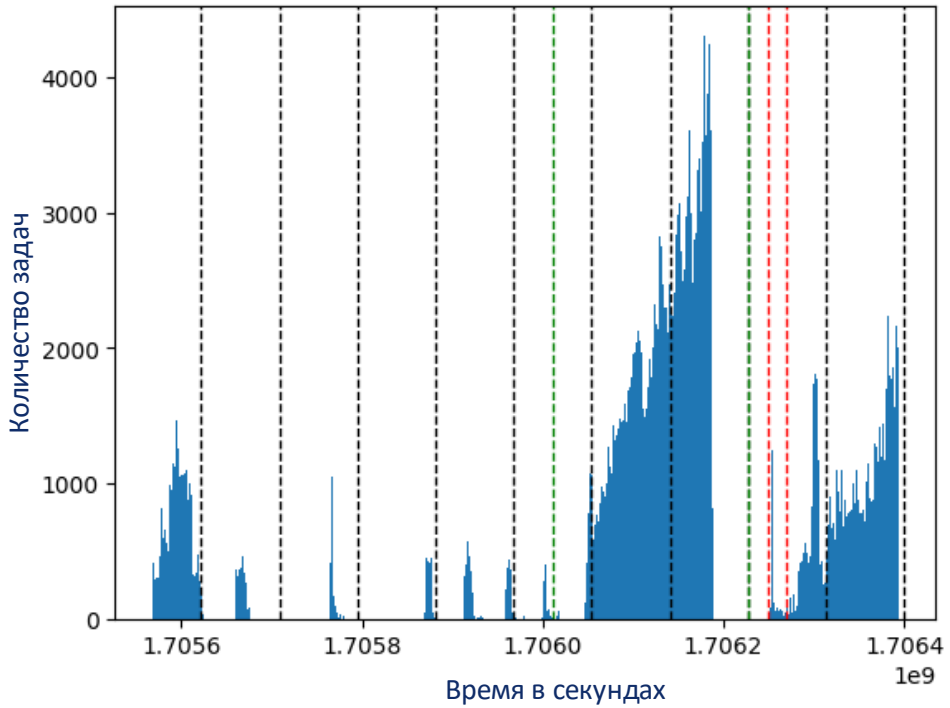
Модификации

- Проект переведен на C++, версия SimGrid - на последнюю,основной файл был разбит на несколько компонент.
- Добавлено поле number_past_through_assimilator в workunit. Был workunit, который был удален до того, как валидатор или ассимилятор закончили с ними работу. В старом коде это была UB.
- Дедлайн выполнения задач сначала рассчитывались как допустимая задержка + время создания, а не допустимая задержка + время отправленния на клиент.
- В случае, если у проекта не было работы для выполнения, клиенты могли заморозиться и никогда не отправить результаты или запросить задачи.
- Возможны два типа хостов - те, которые вычисляют результаты с небольшим количеством ошибок и те, у которых много ошибок. Теперь возможно настроить два кластера с разным количеством ошибочных результатов. Раньше это было невозможно.
- Вместо argc, argv акторам передается конфиг, распарсенный из yaml. Стало удобно добавлять и править параметры в конфигурационном файле

https://github.com/Ksenia-C/combos/tree/master

Верификация ComBoS





Задачи, созданные за неделю измерений (слева), и задачи, присланные с корректным результатом (справа).

RakeSearch: https://rake.boincfast.ru/rakesearch/

Верификация ComBoS

Project scheduling priority

Both scheduling policies involve a notion of **project scheduling priority**, a dynamic quantity that reflects how much processing has been done recently by the project's tasks relative to its resource share.

The scheduling priority of a project P is computed as

```
SP(P) = - REC(P)/resource_share(P)
```

where REC(P) is the amount of processing done recently by P on this host. REC and resource_share are normalized to sum to 1 over all projects.

Документация BOINC

Результаты Верификации ComBoS

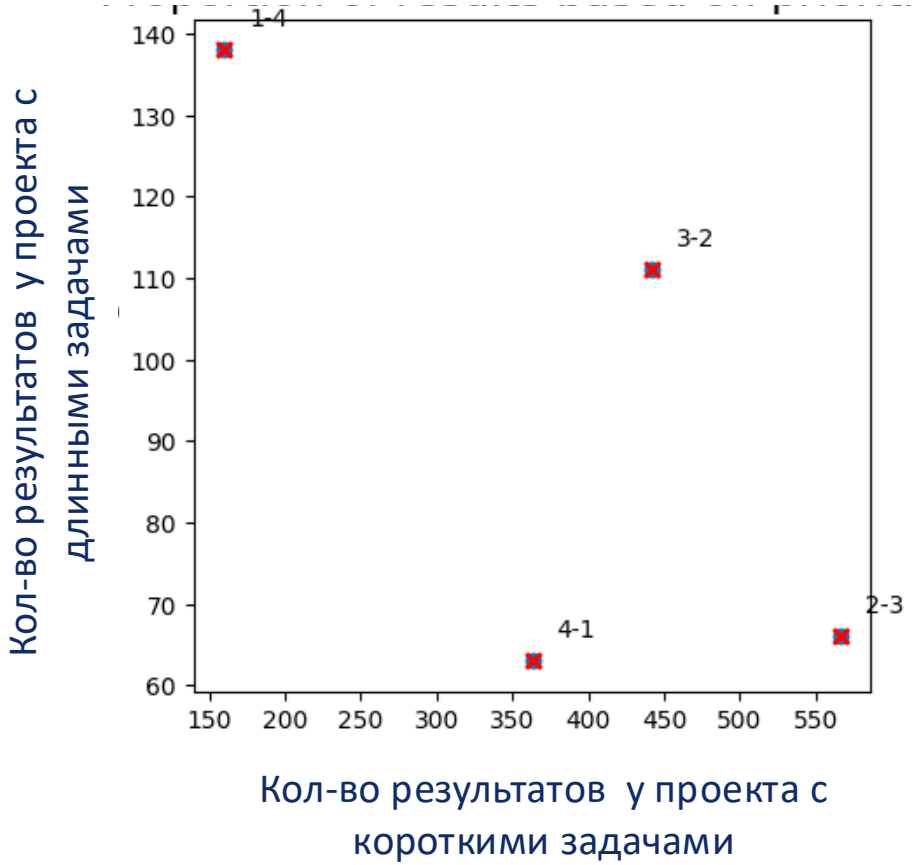
Состояние результата	кол-во результатов в данных	кол-во результатов в симуляции
Валидный	176,620	87,759
Ошибочный	1228	894
Полученный после дедлайна	1051	2

Количество задач с разным статусом в данных по проекту RakeSearch и из симуляции.

2.035e+8 GigaFLOP в датасете vs 8.767e+7 GigaFLOP в симуляции

Таймлайны (1/3)

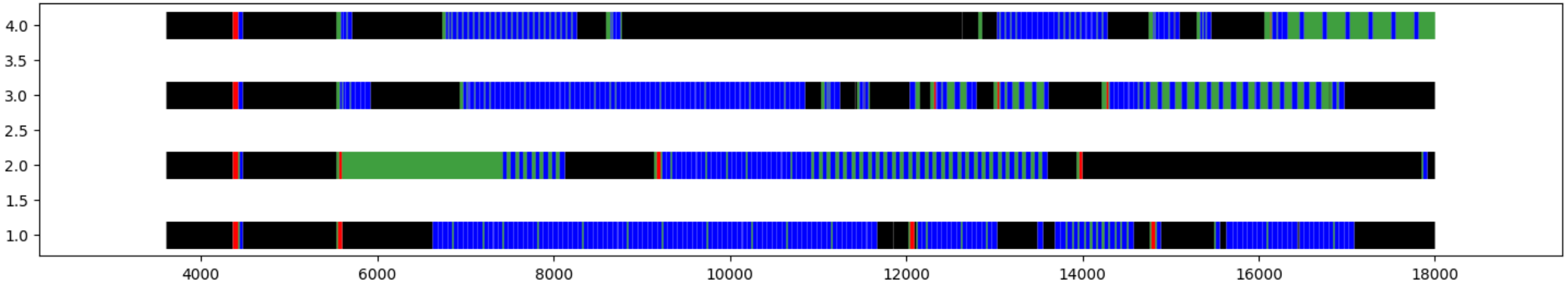
Кол-во результатов в зависимости от приоритета проекта



Количество задач, выполненные для каждого из проекта при разных запусках эксперимента.

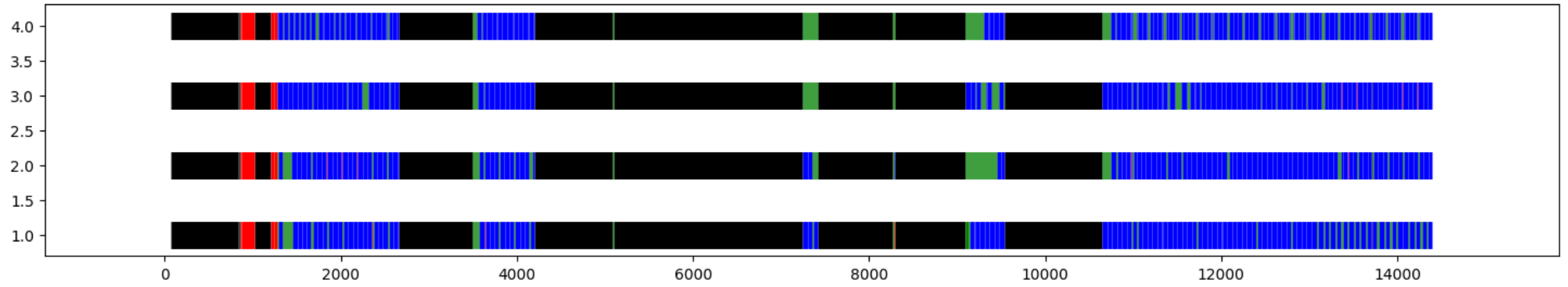
Метки - resource_share проставленные для проектов.
Оси - количество результатов, выполненных для проектов.

Таймлайны (2/3)

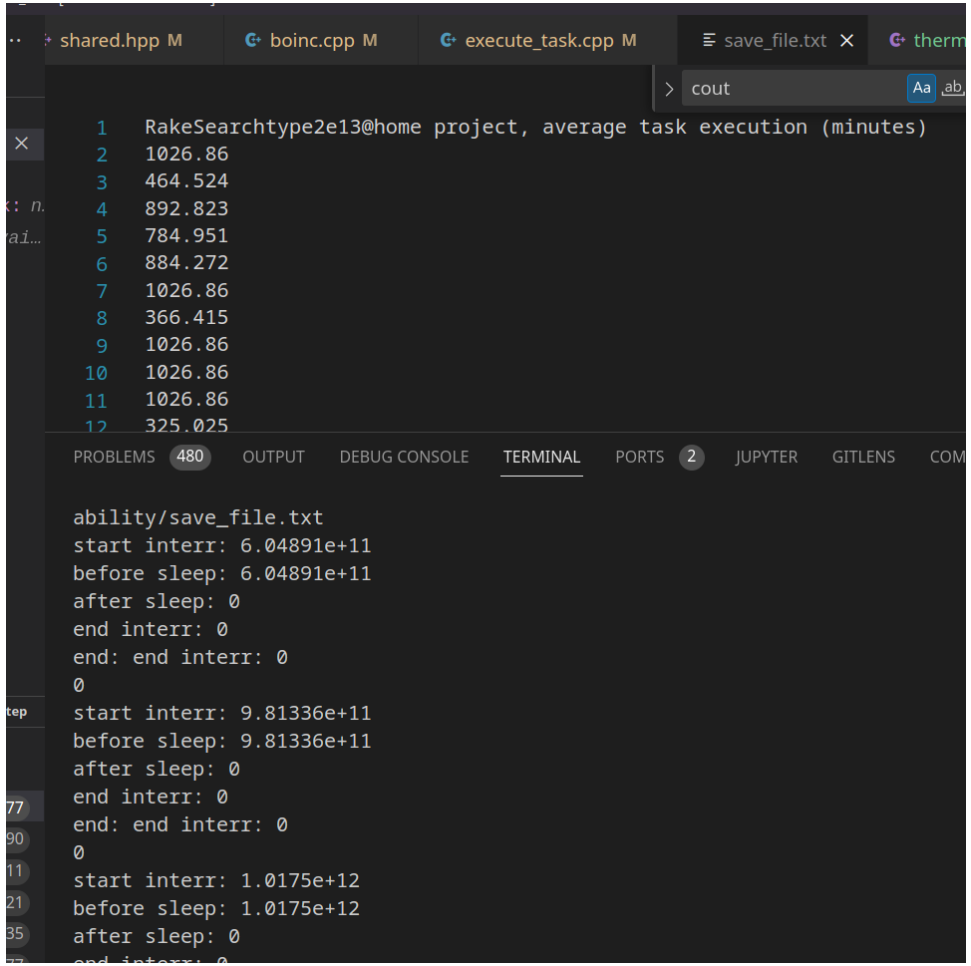


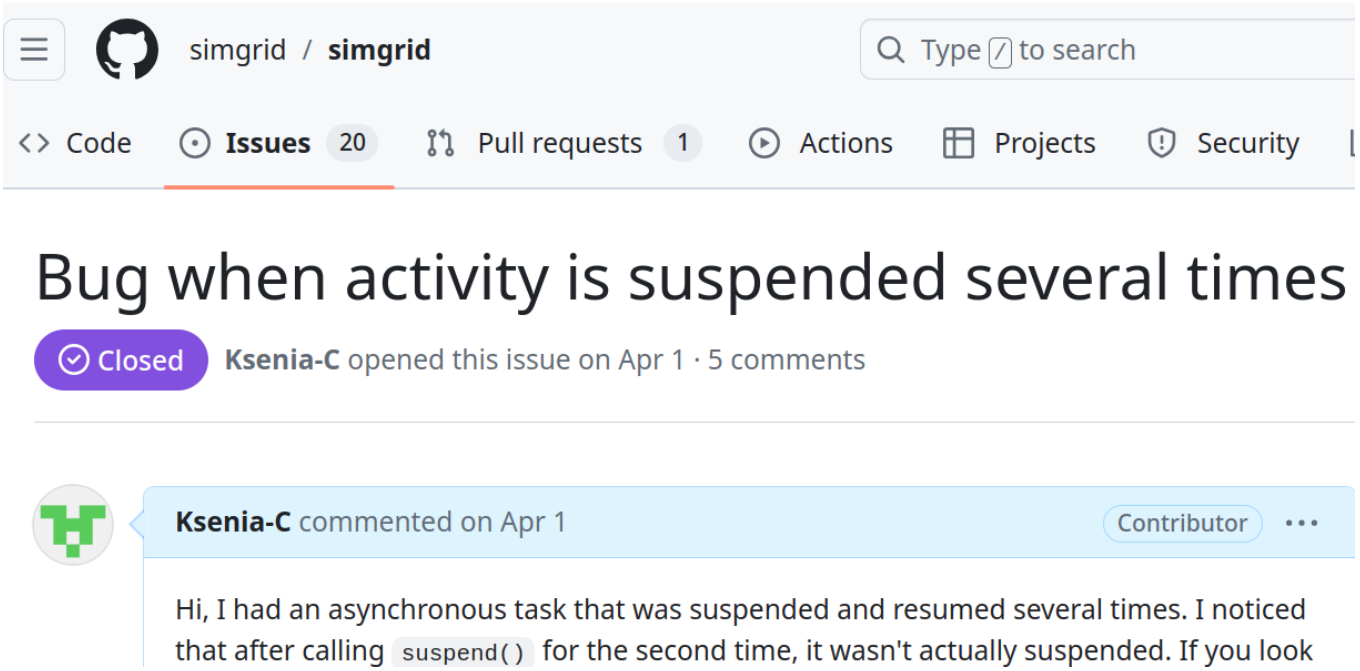
Черный – периоды недоступности
Красный – периоды простоя
Зеленый – период выполнения более быстрых задач
Синий – периоды выполнения более долгих задач

Таймлайны (3/3)



Метрики


shared.hpp M
boinc.cpp M
execute_task.cpp M
save_file.txt
cout
RakeSearchtype2e13@home project, average task execution (minutes)
1026.86
464.524
892.823
784.951
884.272
1026.86
366.415
1026.86
1026.86
1026.86
325.025
PROBLEMS 480
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS 2
JUPYTER
GITLENS
ability/save_file.txt
start interr: 6.04891e+11
before sleep: 6.04891e+11
after sleep: 0
end interr: 0
end: end interr: 0
0
start interr: 9.81336e+11
before sleep: 9.81336e+11
after sleep: 0
end interr: 0
end: end interr: 0
0
start interr: 1.0175e+12
before sleep: 1.0175e+12
after sleep: 0


simgrid / simgrid
Type / to search
Code
Issues 20
Pull requests 1
Actions
Projects
Security
Bug when activity is suspended several times
Closed
Ksenia-C opened this issue on Apr 1 · 5 comments
Ksenia-C commented on Apr 1
Contributor
Hi, I had an asynchronous task that was suspended and resumed several times. I noticed that after calling suspend() for the second time, it wasn't actually suspended. If you look

Результаты Верификации ComBoS

Состояние результата	кол-во результатов в данных	кол-во результатов в симуляции
Валидный	176,620	87,759
Ошибочный	1228	894
Полученный после дедлайна	1051	2

Количество задач с разным статусом в данных по проекту RakeSearch и из симуляции.

2.035e+8 GigaFLOP в датасете vs 8.767e+7 GigaFLOP в симуляции

Верификация одного хоста

ID	кол-во CPU	мощность CPU в GigaFLOPs
A	4	3.576
B	32	6.836

Хосты для верификации из RakeSearch.

	A		B	
метрика	данные	симуляция	данные	симуляция
50% времени выполнения	348.83	275	184.22	142
# завершенных	434	350	12304	746
GigaFLOPs	575.47e+3	341.71e+3	15.20e+6	0.73e+6

Сравнение параметров из датасета и в симуляции.

Верификация одного хоста

ID	кол-во CPU	мощность CPU в GigaFLOPs
A	4	3.576
B	32	6.836

Хосты для верификации из RakeSearch.

	A		B	
метрика	данные	симуляция	данные	симуляция
50% времени выполнения	348.83	275	184.22	142
# завершенных	434	350	12304	746 → 23718
GigaFLOPs	575.47e+3	341.71e+3	15.20e+6	0.73e+6 → 23.16e+6.

Сравнение параметров из датасета и в симуляции.

Относительная работа

TC_{real} - сколько работы выполнил хост

τ - сколько времени работал хост

$ncpus$ - сколько cpu заявлено в базе данных

$HP_{df} = \frac{TC_{real}}{\tau \cdot ncpus}$ - фактическая мощность одного CPU

HP_{db} - мощность одного CPU из базы данных

TC_{sim} - сколько работы выполнил хост в симуляции

$coef_{slow\_down} = \frac{HP_{db}}{HP_{df}}$

$coef_{flops} = \frac{TC_{sim}}{TC_{real}}$

Относительная работа

хост	$coef_{slow\_down}$	время симуляции	#рез-тов в датасете	#рез-тов в симуляции	$coef_{flops}$
1	1.243	20	2659	3797	1.138
2	1.694	18	8934	13298	1.497
3	2.084	37	1176	2426	1.908
4	2.108	38	2215	5574	1.922
5	2.251	38	1282	2605	2.083
6	2.267	37	2296	8578	2.056
7	2.373	38	2608	6040	2.157
8	2.434	38	4730	16436	2.229
9	2.461	37	1333	4588	2.256
10	2.540	38	2109	6105	2.306
11	2.726	38	1442	3856	2.513
12	3.337	38	2249	9707	3.030
13	4.619	37	1798	8743	4.196
14	6.924	37	1450	11904	6.246
15	8.016	37	1261	8214	7.177
16	27.093	38	1037	5511	24.775

Результаты эксперимента и вычисление коэффициентов.

$$coef_{slow\_down} = \frac{HP_{db}}{HP_{df}} = \frac{HP_{db} \cdot ncpus \cdot \tau}{TC_{real}} \approx coef_{flops} = \frac{TC_{sim}}{TC_{real}}, \quad HP_{db} \cdot ncpus \cdot \tau \approx TC_{sim}$$

Ускорение завершения эксперимента (стратегии)

- **base** - код без изменений, задачи отсылаются, если их запрашивают, независимо от кластера.
- **v1** - задачи отправляются первый раз не зависимо от кластера. Если задача завершилась с ошибкой, она отправляется снова только на хороший.
- **v2** - до определенного процента выполнения эксперимента задачи отправляются всем, после - только хорошим.
- **v3** - задачи отправляются независимо от кластера. Если задач на выполнение больше нет (были разданы всем хотя по одному инстансу), а хороший хост запрашивает работу, то задача, отправленная на плохой хост, но еще не полученная, реплицируется и отдается в ответ на запрос. В итоге дожидается ответ от любого из хоста.
- **v4** - создается больше реплик еще не выполненных задач ближе к концу эксперимента.

Ускорение завершения эксперимента (результаты)

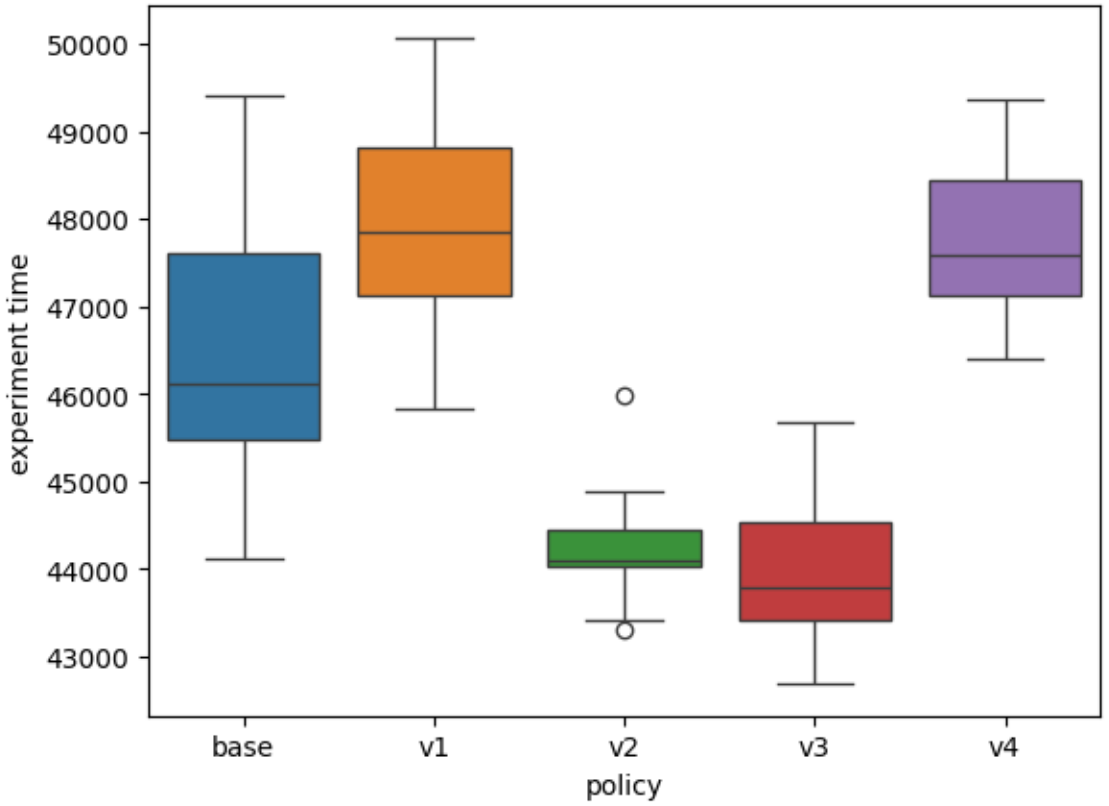
policy	10% валидных рез-тов	20%	40%	60%	80%
base	61 584 (17.1h)	52 913 (14.7h)	45 647 (12.7h)	41 048 (11.4h)	**35 351 (9.8h)**
v1	55 546 (15.4h)	52 913 (14.7h)	47 012 (13.1h)	43 672 (12.1h)	37 168 (10.3h)
v2 (50%)	56 068 (15.6h)	52 913 (14.7h)	48 288 (13.4h)	46 109 (12.8h)	40 256 (11.2h)
v2 (80%)	**54 583 (15.2h)**	**50 280 (14.0h)**	**43 672 (12.1h)**	**39 724 (11.0h)**	**34 978 (9.7h)**
v2 (90%)	**54 583 (15.2h)**	52 744 (14.7h)	45 647 (12.7h)	40 256 (11.2h)	37 521 (10.4h)
v3	**54 583 (15.2h)**	**51 315 (14.3h)**	**43 907 (12.2h)**	**39 227 (10.9h)**	35 847 (10.0h)
v4	58 081 (16.1h)	53 560 (14.9h)	47 303 (13.1h)	41 843 (11.6h)	37 966 (10.5h)

Результаты эксперимента с хостами, порождающими большое число некорректных результатов.

policy	20%	40%	60%	80%
base	369 473 (102.6h)	309 795 (86.1h)	**271 142 (75.3h)**	**242 976 (67.5h)**
v1	377 583 (104.9h)	341 557 (94.9h)	299 308 (83.1h)	249 491 (69.3h)
v2 (80%)	357 050 (99.2h)	315 734 (87.7h)	281 859 (78.3h)	243 609 (67.7h)
v2 (90%)	**352 790 (98.0h)**	**304 963 (84.7h)**	**271 151 (75.3h)**	245 762 (68.3h)
v3	**352 790 (98.0h)**	**301 384 (83.7h)**	**266 448 (74.0h)**	**241 117 (67.0h)**

Результаты эксперимента с хостами, порождающими большое число некорректных результатов (большая продолжительность задач).

Ускорение завершения эксперимента (результаты)



Распределения времен завершения набора задач для разных политик

Заключение

- Код симулятора стал проще для понимания и более корректным.
- Были найдены и устранены неочевидные ошибки в симуляторе, которые негативно сказывались на его работу и проведение экспериментов (например, вычисление дедлайна задачи).
- Появилась возможность валидировать симуляции с помощью таймлайнов и метрик, иллюстрирующие внутренную работу симуляции.
- Симулятор был провалидирован на данных с проекта RakeSearch.
- С помощью модифицированного симулятора проведено исследование по ускорению завершения эксперимента.
- Код доступен по https://github.com/Ksenia-C/combos

Спасибо за внимание!

Благодарности:

- Команде RakeSearch за предоставленные данные
- Команде ComBos за открытый код
- Команде SimGrid за приятную работу с ними
- Работникам библиотеки ВШЭ за хорошее рабочее место

Запуск ComBoS

```
double time1 = MSG_get_clock();

err = MSG_task_execute(task->msg_task);
printf("time for execution at %f for %f\n", time1, MSG_get_clock() - time1);
```

Jun 27, 2023

agiersch

v3.34

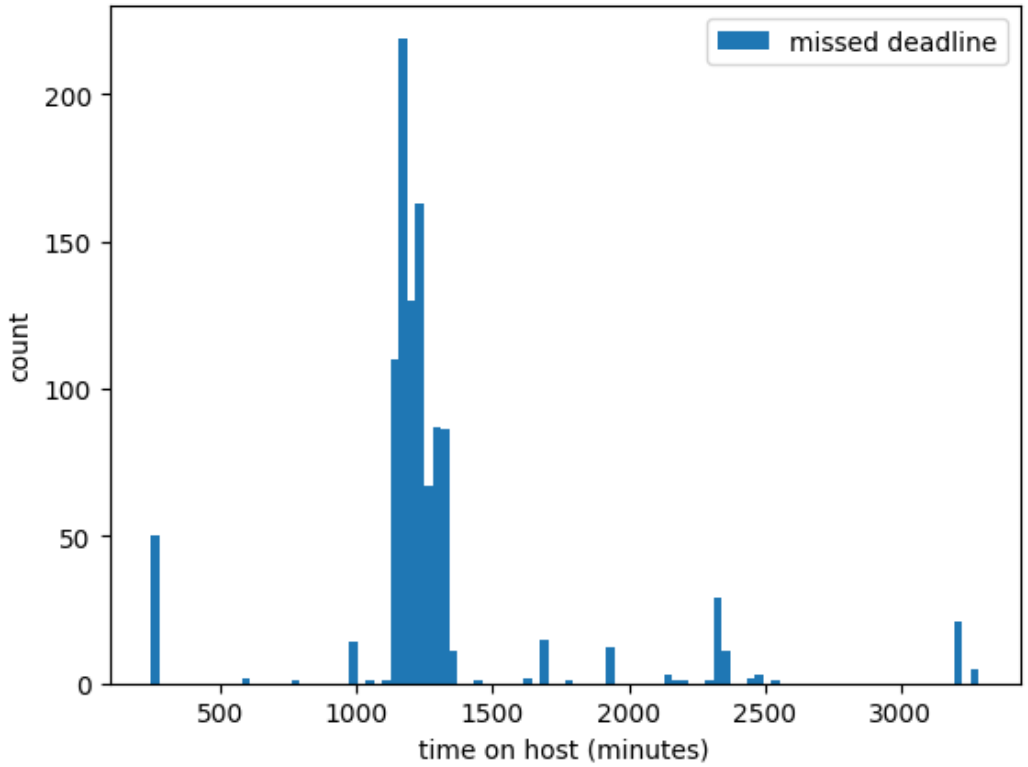
036c801

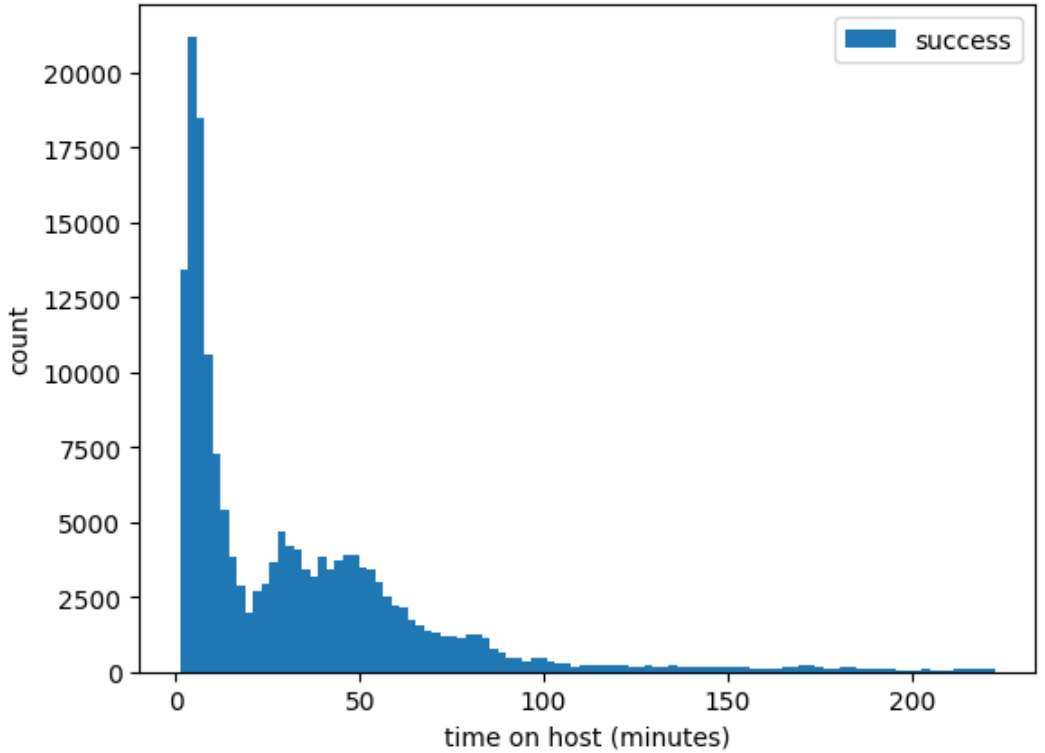
Compare

Save the planet, skip a release: 3.33 was due 6 months ago, so skip directly to 3.34.

- MSG and Java are gone (EOL was scheduled for 2020), move to C++17 and drop 32bits support.

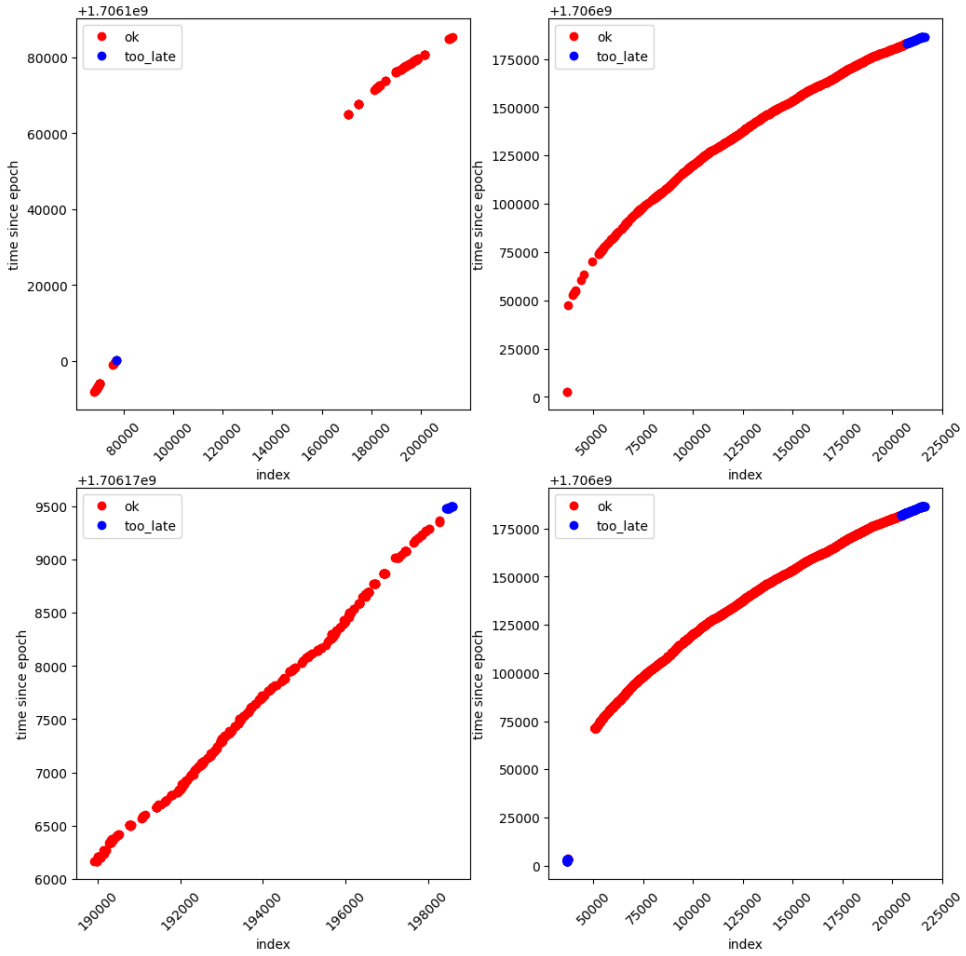
Результаты пришедшие слишком поздно





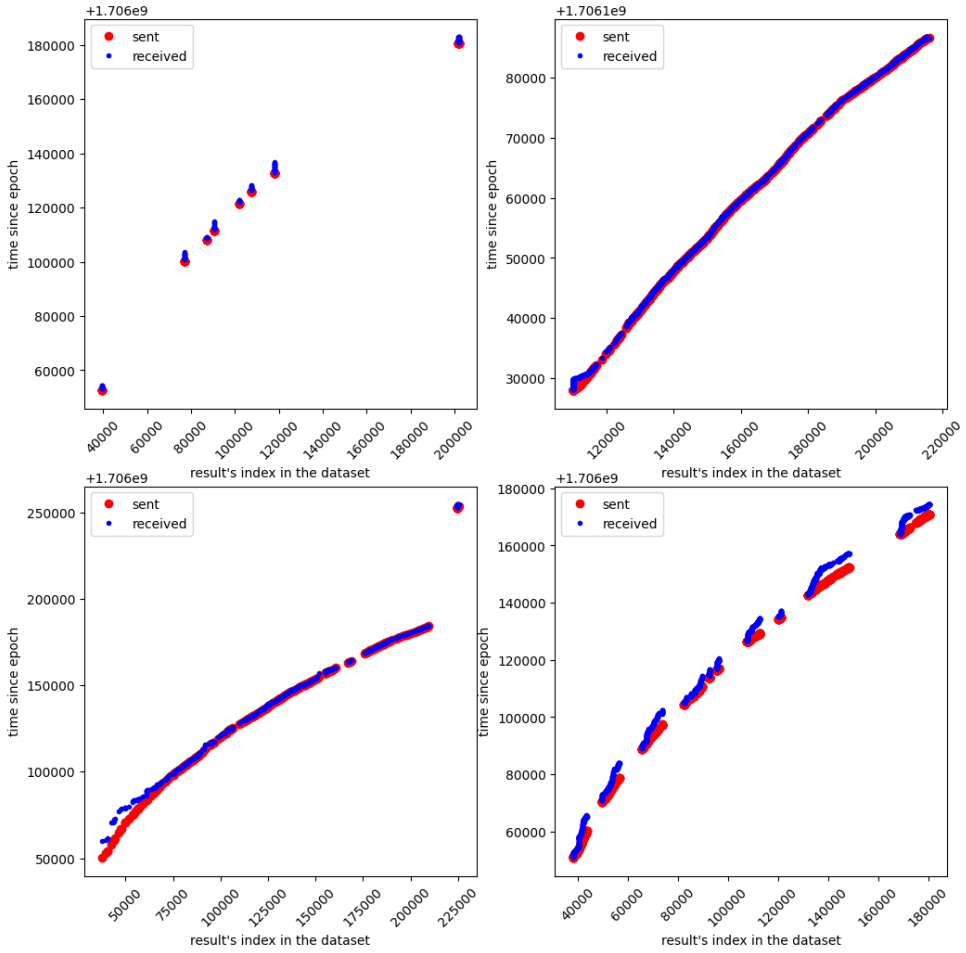
Гистограммы времени нахождения задач на хосте.

Результаты пришедшие слишком поздно



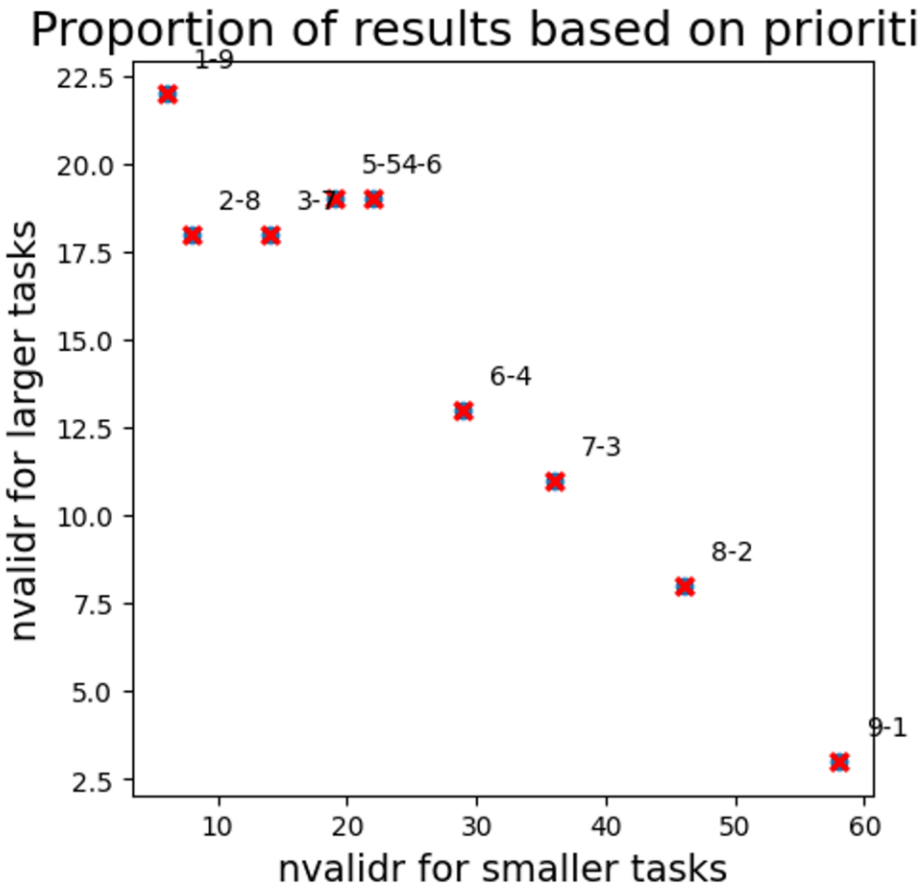
Время получения на клиенте корректных задач и задач, отправленных слишком поздно, для выделенных хостов.

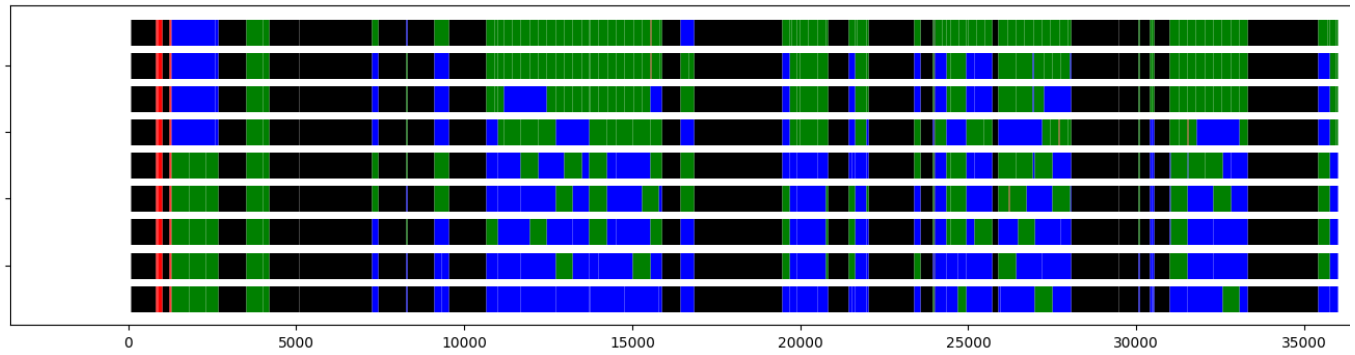
Результаты пришедшие слишком поздно



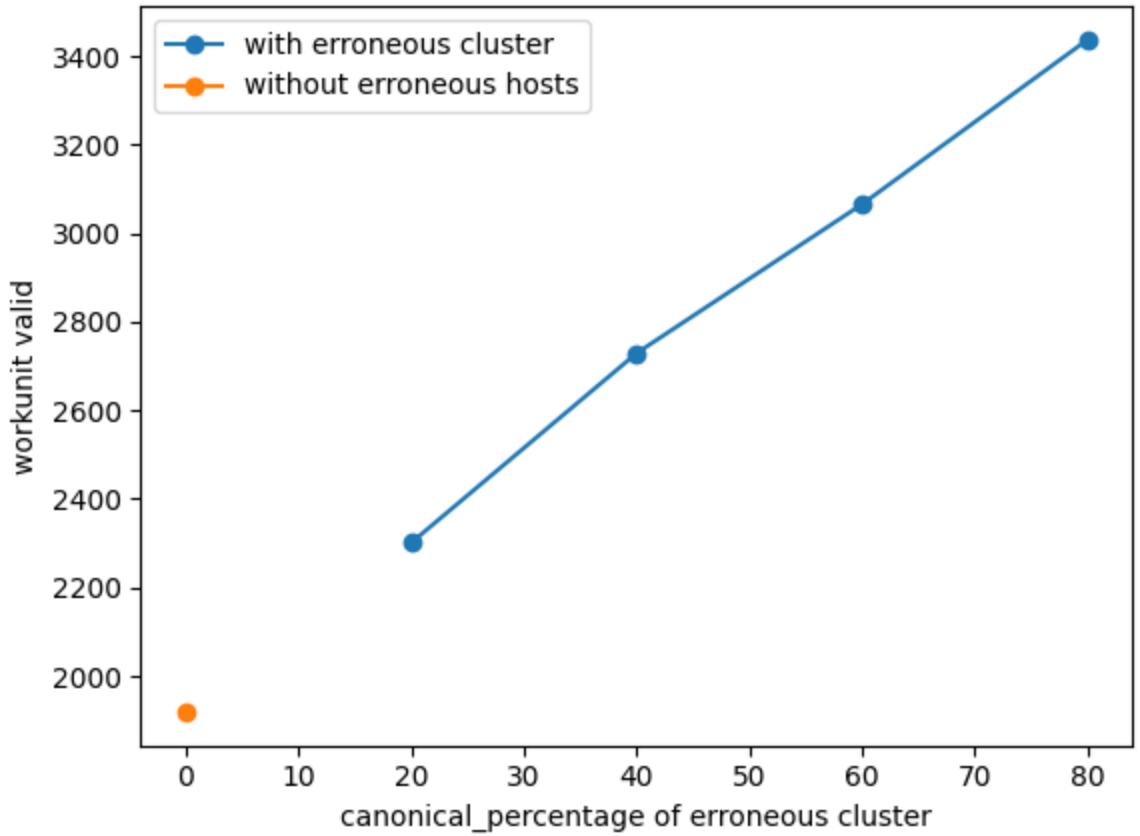
Время получения на клиенте задач (красный) и время их отправки на сервер (синий).

resource_share и таймлайны





Ускорение завершения эксперимента

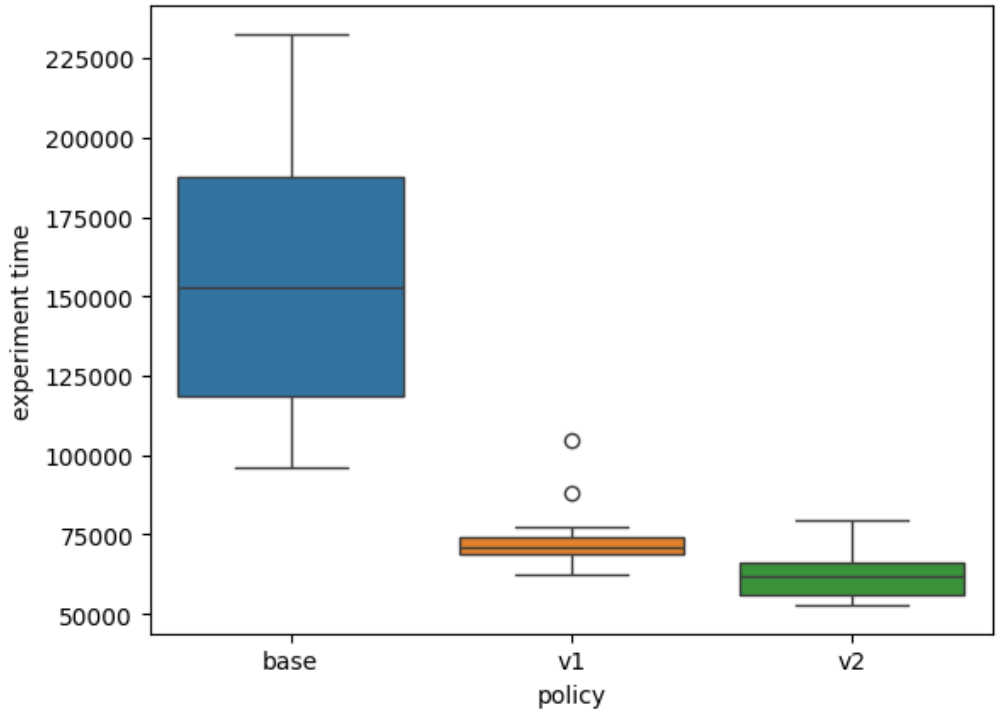


Количество выполненных задач при разных параметров второго кластера, и с его отключением.

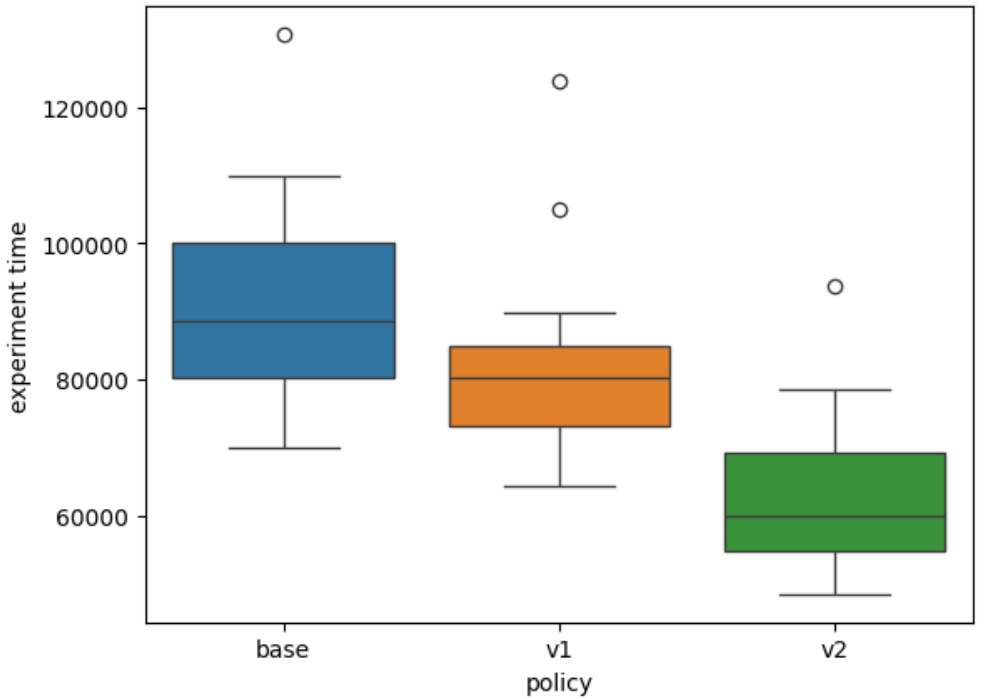
Ускорение завершения эксперимента

- **base** - код без изменений, задачи отсылаются, если их запрашивают, независимо от кластера.
- **v1** - не отсылать плохим хостам задачи.
- **v2** - аналогично **v3** предыдущего пункта - если все задачи разосланы, а хорошие хосты запрашивают еще - прореплицировать незавершенные задачи с плохих хостов.

Ускорение завершения эксперимента



cluster	avalability parameters	non-avalability parameters
1	gamma(0.357, 43.652)	exponential (0.615)
2	exponential (0.857)	exponential (0.15)



cluster	avalability parameters	non-avalability parameters
1	gamma(0.357, 43.652)	exponential (0.615)
2	exponential (0.9)	exponential (0.35)