

Суперкомпьютерные дни в России

Координированное сохранение контрольных точек
с журналированием передаваемых данных и
асинхронное восстановление расчетов
после отказов

Бондаренко А.А.

Ляхов П.А.

Якобовский М.В

Москва
24 сентября 2018

Цель работы

Разработать принципы сохранения контрольных точек за время, меньшее характерной продолжительности безотказной работы системы, и алгоритмы, обеспечивающие, в случае отказа части оборудования, быстрое автоматическое возобновление расчета на работоспособной части вычислительного поля.

Задачи работы

- Разработать модель исполнения параллельной программы, содержащей различные техники обеспечения отказоустойчивости и выполняемой на вычислительных системах, подверженных частым отказам.
- Разработать алгоритм асинхронного восстановления после отказов, не требующий возврата большинства процессов к последней контрольной точке.

Отказы в высокопроизводительных вычислительных системах

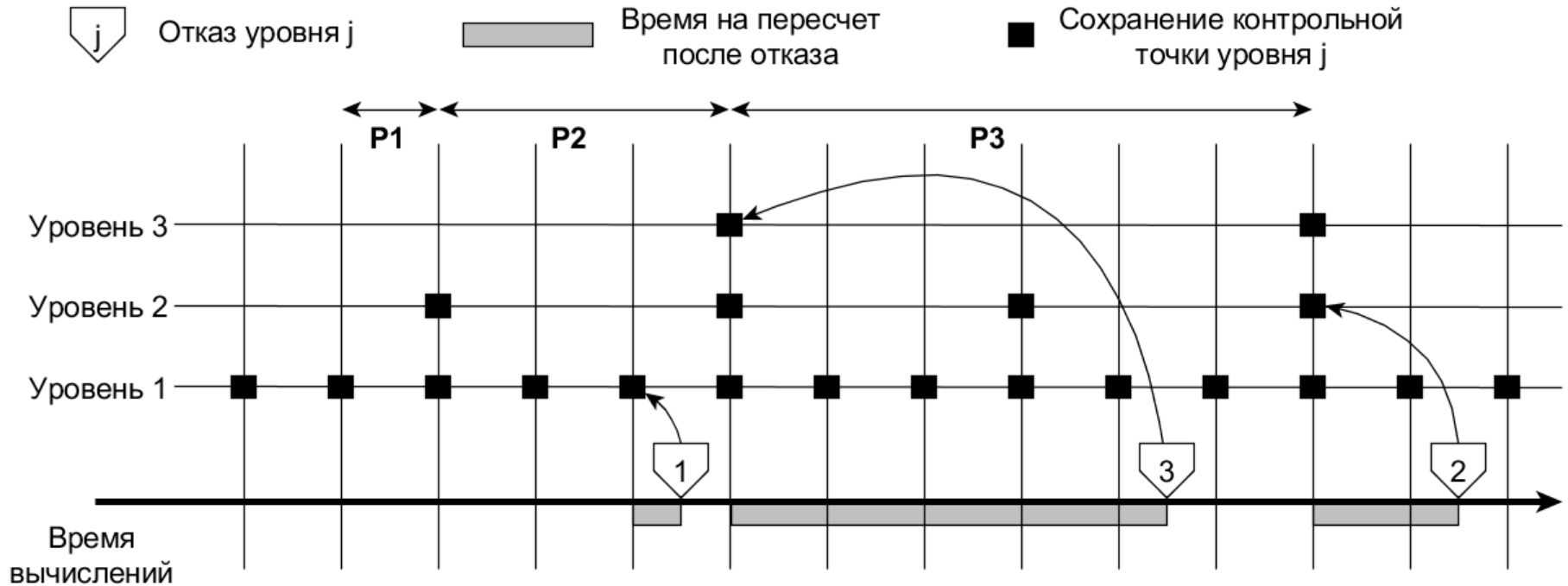
- На данный момент:
 - Среднее время между отказами на высокопроизводительных вычислительных системах составляет менее одного дня
 - В стандарте MPI нет механизмов обеспечения отказоустойчивости
- Специалистами в области HPC утверждается, что на системах уровня Эксафлопс среднее время между отказами будет находиться в диапазоне от 9 часов до 1 часа

1. Bergman K. et al. Exascale computing study: Technology challenges in achieving exascale systems //Defense Advanced Research Projects Agency Information Processing Techniques Office (DARPA IPTO), Tech. Rep. – 2008. – Т. 15.
2. Bland W. et al. Lessons learned implementing user-level failure mitigation in MPICH //2015 15th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid). – IEEE, 2015. – С. 1123-1126.
3. Dongarra J., Herault T., Robert Y. Fault-Tolerance Techniques for High-Performance Computing. Springer, 2015. 320 p. DOI: 10.1007/978-3-319-20943-2
4. Cappello F., Geist A., Gropp W., Kale S., Kramer B., Snir M. Toward Exascale Resilience: 2014 update // Supercomputing frontiers and innovations. 2014. Vol.1, No. 1. P. 1–28.

Основные допущения многоуровневого метода

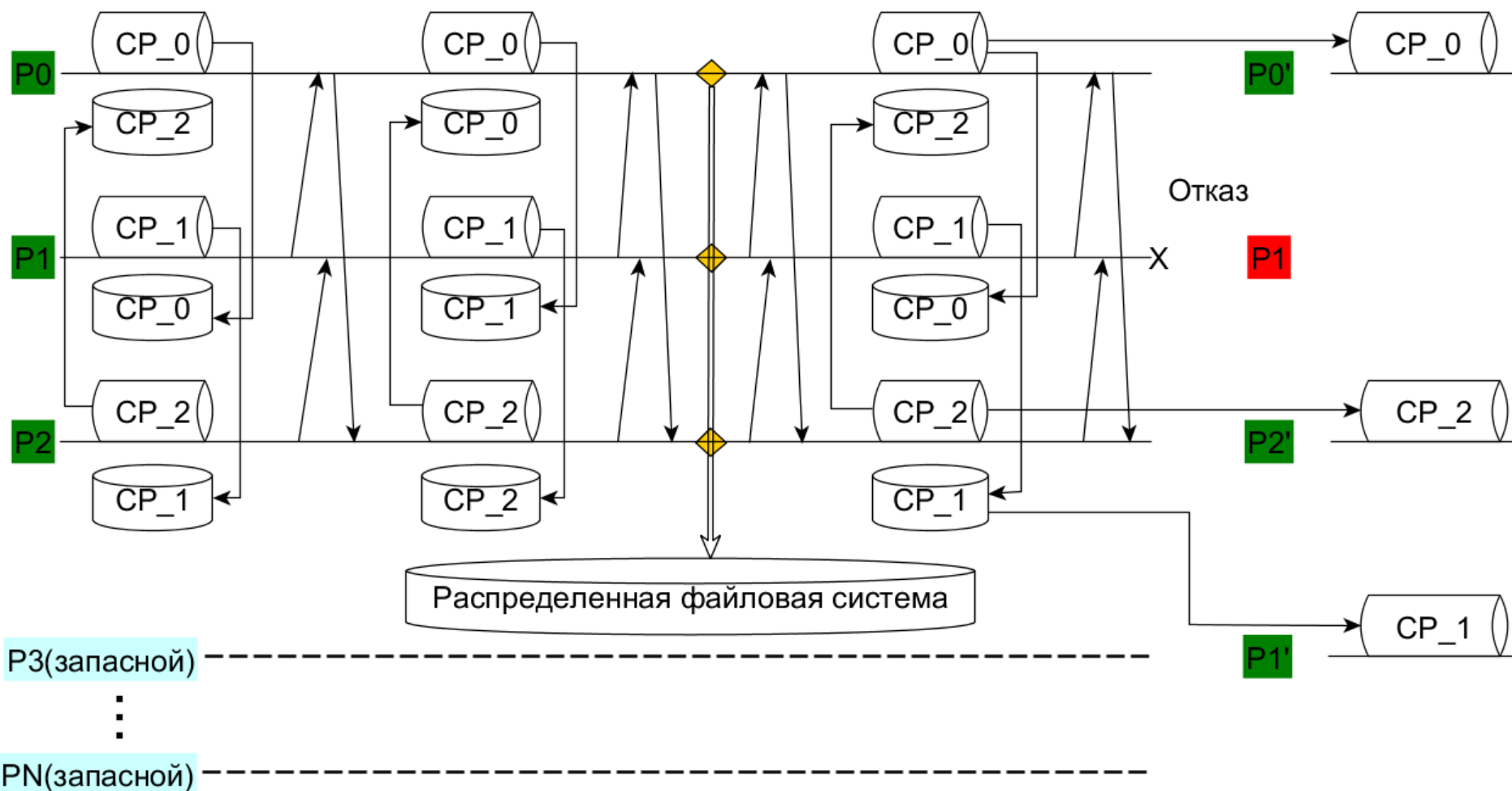
- Время между отказами – случайная величина, имеющая экспоненциальное распределение.
- в системе могут происходить от 1 до k уровней (рассматриваемых типов) отказа;
- наступление отказов на разных уровнях – независимые случайные величины;
- каждый уровень j имеет свою частоту отказа λ_j , свои накладные расходы на сохранение контрольной точки C_j и восстановление из контрольной точки R_j ;

Многоуровневое сохранение контрольных точек



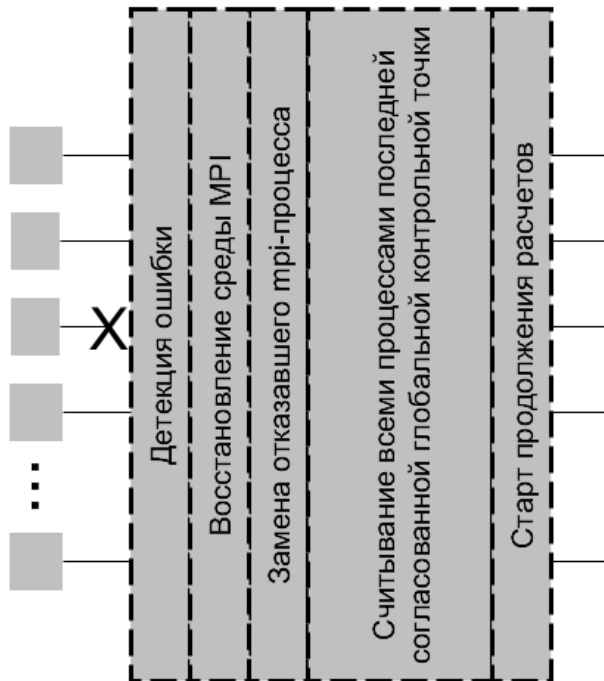
1. Di S. et al. Optimization of multi-level checkpoint model for large scale HPC applications //Parallel and Distributed Processing Symposium, 2014 IEEE 28th International. – IEEE, 2014. – С. 1181-1190.
2. Benoit A. et al. Towards optimal multi-level checkpointing //IEEE Transactions on Computers. – 2017. – Т. 66. – №. 7. – С. 1212-1226.
3. Di S. et al. Toward an optimal online checkpoint solution under a two-level HPC checkpoint model //IEEE Transactions on Parallel and Distributed Systems. – 2017. – Т. 28. – №. 1. – С. 244-259.

Стандартная стратегия восстановления

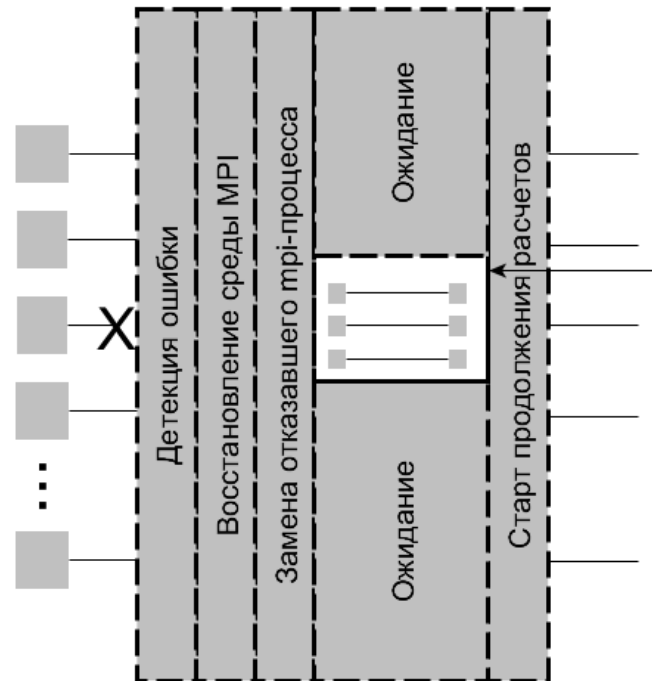


Этапы восстановления расчета после отказа

Стандартное
восстановление

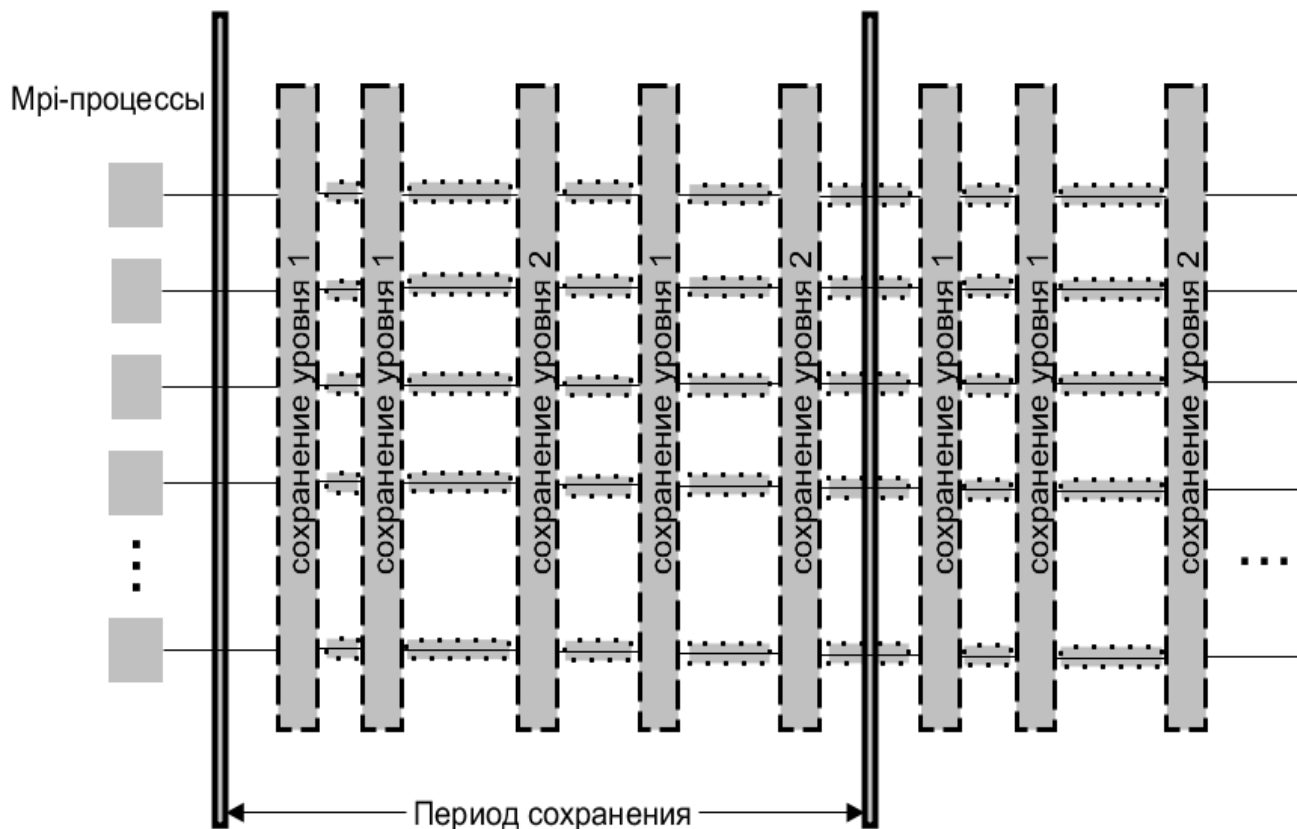


Асинхронное
восстановление



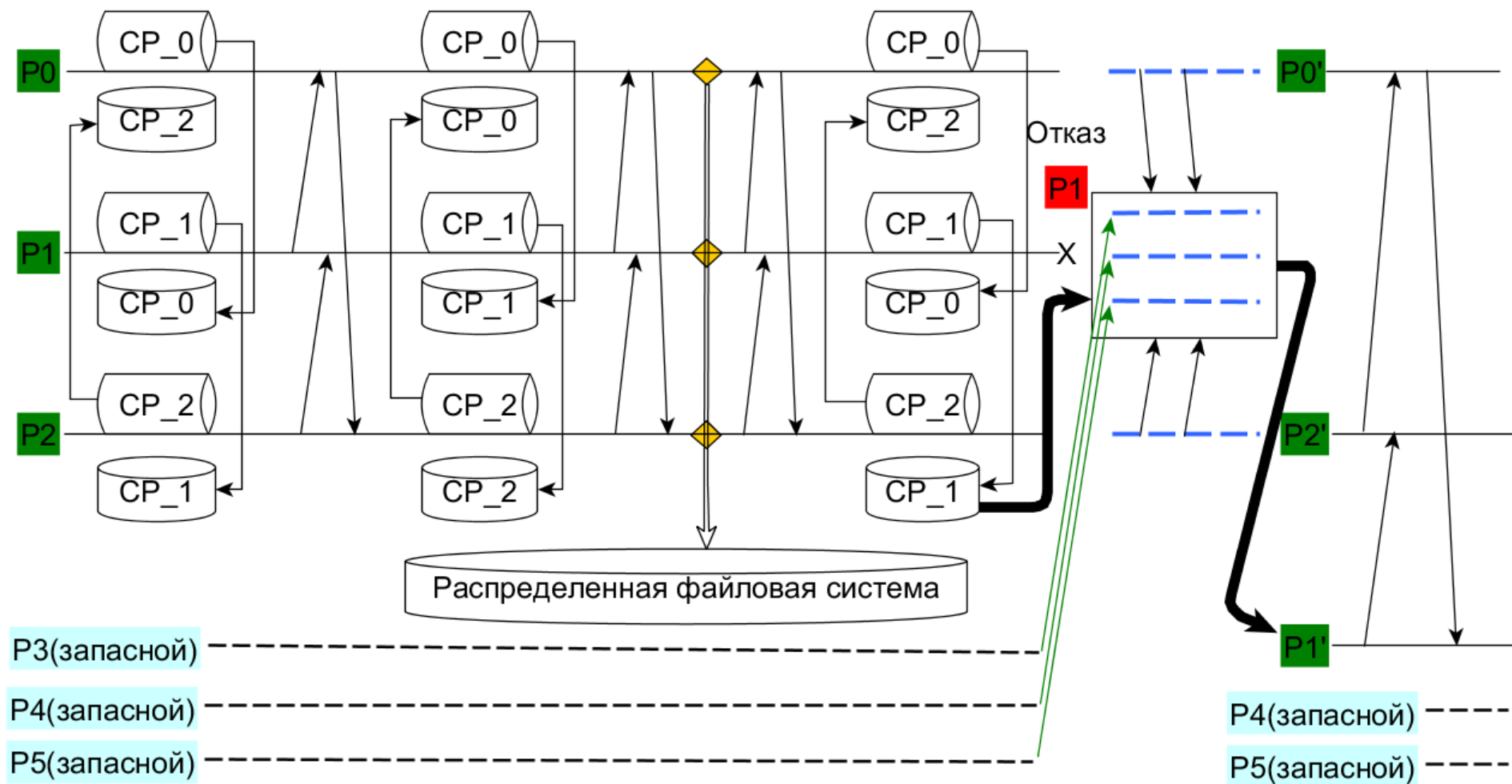
расчет данных
резервными
процессами от
последнего
сохранения до
момента отказа

Сохранение контрольных точек с журналированием передаваемых данных



Замечание: В данной работе для получения предварительных оценок полагаем, что накладные расходы на журналирование (logging) равны 0.

Асинхронное восстановление



Моделирование исполнения программ

Итерационный метод оценки накладных расходов

- 2 времени счета « T_{comp} », « T_{cur} » для фиксации выполнения полезной работы и наступления событий (отказа, сохранения)
- работа продолжается, пока не будет выполнен базовый расчет (время полезной работы) T_{comp}
- наступление ошибок и времени сохранения определяется по текущему времени T_{cur}

$iter \leftarrow 1$

while $T_{comp} < T_{max}$ **do**

$TFS \leftarrow \text{get time for saves } (iter, T_{cur}, C_i)$

$LOF \leftarrow \text{get level of failure } (iter, T_{cur})$

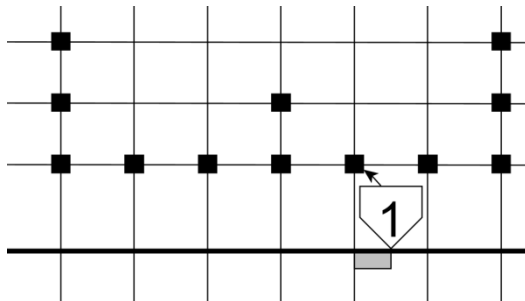
$T_{comp} \leftarrow T_{comp} + \text{get comp difference } (LOF)$

$T_{cur} \leftarrow T_{cur} + \text{get cur difference } (LOF)$

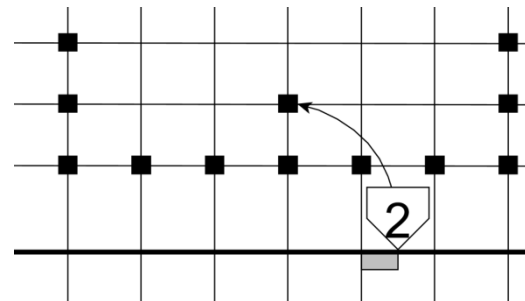
$iter \leftarrow iter + \text{get iter difference } (LOF)$

end while

return T_{cur}

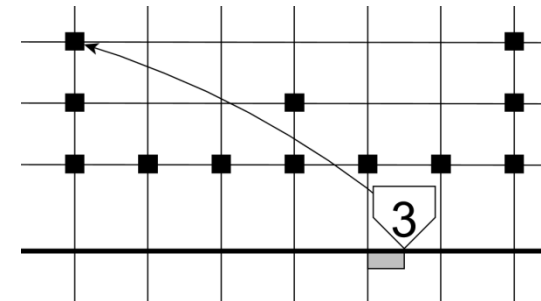


Нет отказа



Случай 1

Случай 2



Случай 3

$\text{get comp difference } ()$

Δt

0

$-\Delta t$

$-4\Delta t$

$\text{get cur difference } ()$

$\Delta t + TFS,$

+«Серое время»

+«Серое время»

+«Серое время»

$\text{get iter difference } ()$

1

0

-1

-4

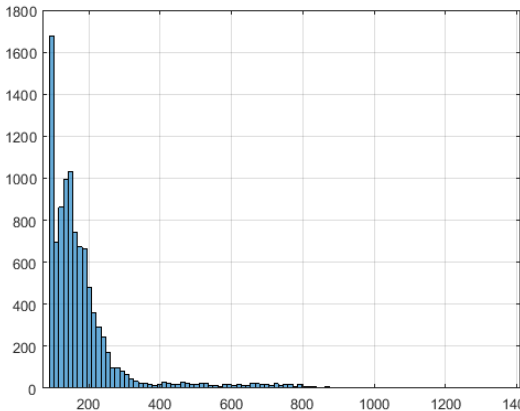
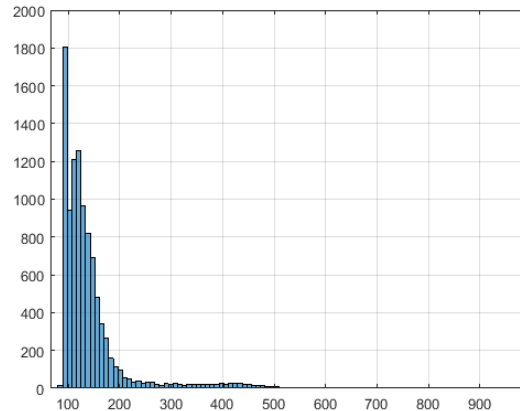
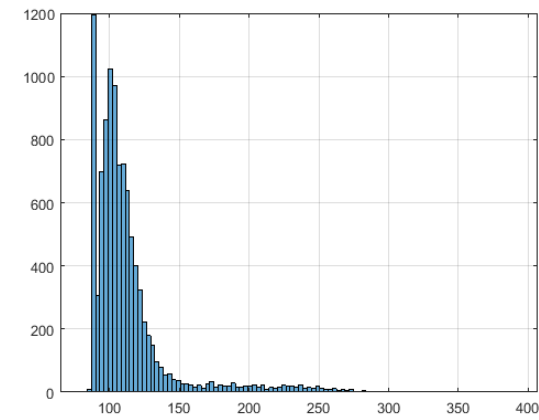
10

Описание вычислительного эксперимента

- Время до наступления следующего отказа (этого уровня) является случайной величиной, которая подчиняется экспоненциальному распределению
- В работе рассматривается двухуровневая стратегия сохранения (в память соседнему процессу, в РФС)
- Тестовая задача - расчет распределения тепла в тонкой пластине
- Рассматриваются 3 алгоритма, с координированным сохранением и
 - ❖ со стандартным восстановлением;
 - ❖ журналированием, с асинхронным восстановлением, `recalc=2`;
 - ❖ журналированием, с асинхронным восстановлением, `recalc=5`;
- Разработаны программы, реализующие моделирование исполнения параллельных программ и оценивающие накладные расходы итерационным методом
- Параллельные программы, в которых отказ происходит с помощью `raise(SIGKILL)`, обработка отказа происходит с помощью функционала ULFM (`MPIX_Comm_revoke(comm)`, `MPIX_Comm_shrink`, ...) (fault-tolerance.org)

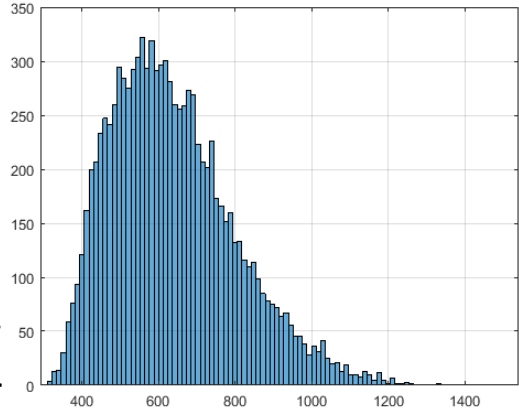
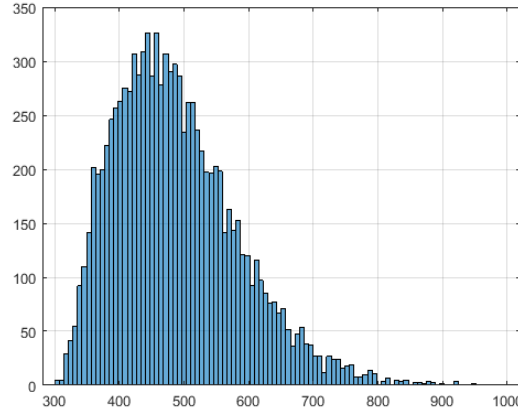
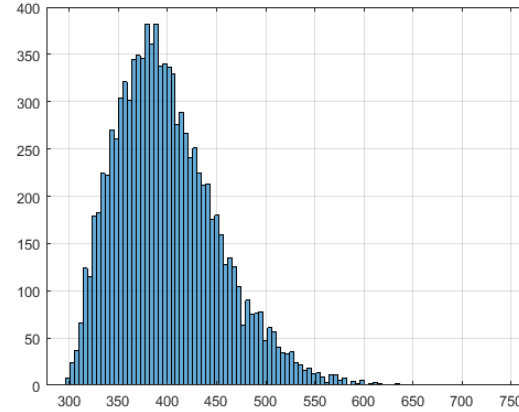
Накладные расходы при разных стратегиях восстановления

Параметры $C = [1 \ 6]$, $R = [0.5 \ 4]$, $T_6 = 3600$, $MTBF = [1800 \ 36000]$

	Стандартная	Асинхронная, resalc=2	Асинхронная, resalc=5
	$\bar{t} = 187, \sigma = 130$ $t_{min} = 90, t_{max} = 1391$ $\bar{n}_1 = 2.1 \ \bar{n}_2 = 0.1,$	$\bar{t} = 142, \sigma = 72$ $t_{min} = 85, t_{max} = 975$ $\bar{n}_1 = 2.1 \ \bar{n}_2 = 0.1, \Delta T = 45(24\%)$	$\bar{t} = 113, \sigma = 30$ $t_{min} = 85, t_{max} = 387$ $\bar{n}_1 = 2.1 \ \bar{n}_2 = 0.1, \Delta T = 74(40\%)$
Итерационный, $N = 10000$			
ULFM, $N = 3$	$\bar{t} = 146$ $\bar{n}_1 = 2, \bar{n}_2 = 0$	$\bar{t} = 121$ $\bar{n}_1 = 2, \bar{n}_2 = 0, \Delta T = 25(17\%)$	$\bar{t} = 77$ $\bar{n}_1 = 2, \bar{n}_2 = 0, \Delta T = 69(47\%)$

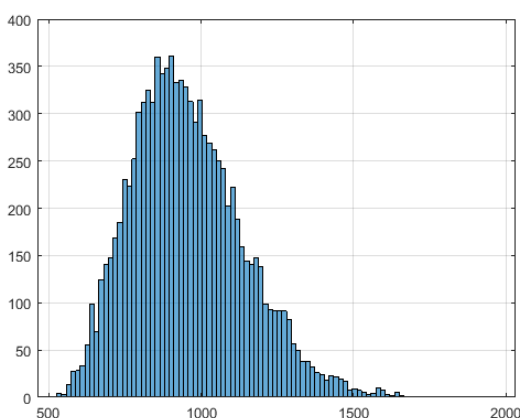
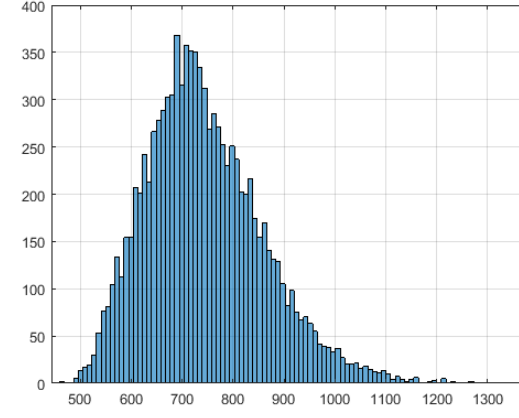
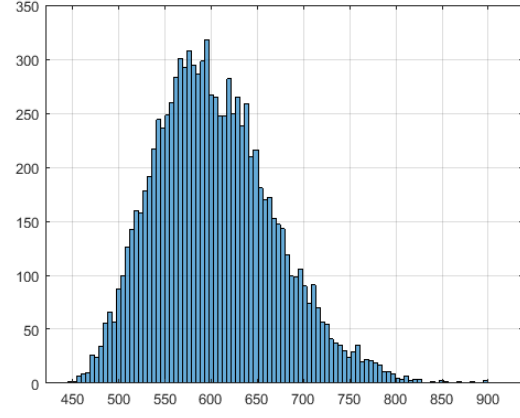
Накладные расходы при разных стратегиях восстановления

Параметры $C = [1 \ 6]$, $R = [0.5 \ 4]$, $T_0 = 3600$, $MTBF = [360 \ 1800]$

	Стандартная	Асинхронная, resalc=2	Асинхронная, resalc=5
	$\bar{t} = 638, \sigma = 164$ $t_{min} = 312, t_{max} = 1520$ $\bar{n}_1 = 11.8 \ \bar{n}_2 = 2.4,$	$\bar{t} = 488, \sigma = 97$ $t_{min} = 301, t_{max} = 1001$ $\bar{n}_1 = 11.4 \ \bar{n}_2 = 2.3, \Delta T = 150(24\%)$	$\bar{t} = 400, \sigma = 53$ $t_{min} = 297, t_{max} = 744$ $\bar{n}_1 = 11.1 \ \bar{n}_2 = 2.2, \Delta T = 238(37\%)$
Итерационный, $N = 10000$			
ULFM, $N = 3$	$\bar{t} = 714$ $\bar{n}_1 = 14, \bar{n}_2 = 4$	$\bar{t} = 626$ $\bar{n}_1 = 14, \bar{n}_2 = 4, \Delta T = 88(12\%)$	$\bar{t} = 620$ $\bar{n}_1 = 14, \bar{n}_2 = 4, \Delta T = 94(13\%)$

Накладные расходы при разных стратегиях восстановления

Параметры $C = [1 \ 6]$, $R = [0.5 \ 4]$, $T_0 = 3600$, $MTBF = [180 \ 900]$

	Стандартная	Асинхронная, resalc=2	Асинхронная, resalc=5
	$\bar{t} = 955, \sigma = 184$ $t_{min} = 483, t_{max} = 2012$ $\bar{n}_1 = 25.3 \ \bar{n}_2 = 5.1,$	$\bar{t} = 744, \sigma = 116$ $t_{min} = 446, t_{max} = 1363$ $\bar{n}_1 = 24.2 \ \bar{n}_2 = 4.9, \Delta T = 211(22\%)$	$\bar{t} = 604, \sigma = 65$ $t_{min} = 441, t_{max} = 918$ $\bar{n}_1 = 23.3 \ \bar{n}_2 = 4.7, \Delta T = 351(37\%)$
Итерационный, $N = 10000$			
ULFM, $N = 3$	$\bar{t} = 1051$ $\bar{n}_1 = 24, \bar{n}_2 = 6$	$\bar{t} = 926$ $\bar{n}_1 = 23, \bar{n}_2 = 6, \Delta T = 125(12\%)$	$\bar{t} = 730$ $\bar{n}_1 = 21, \bar{n}_2 = 6, \Delta T = 321(31\%)$

Заключение

Результаты, полученные итерационным методом оценки накладных расходов, показывают, что для асинхронного метода восстановления:

- применение 2 процессов для пересчета, позволяет сократить накладные расходы на 20 % и более,
- применение 5 процессов для пересчета, позволяет сократить накладные расходы на 35 % и более.

Вычислительный эксперимент на тестовой задаче с осуществлением отказа `mpi`-процессов с помощью функции `raise(SIGKILL)` и последующим восстановлением параллельного приложения с помощью функций `ULFM` показал, что для асинхронного метода восстановления:

- применение 2 процессов для пересчета, позволяет в некоторых случаях сократить накладные расходы до 17 %, но не менее чем на 12%,
- применение 5 процессов для пересчета, позволяет в некоторых случаях сократить накладные расходы до 47 %, но не менее чем на 13%.

Спасибо за внимание

bondaleksey@gmail.com Бондаренко А.А.

pavel.lyakhov@phystech.edu Ляхов П.А.

lira@imamod.ru Якобовский М.В