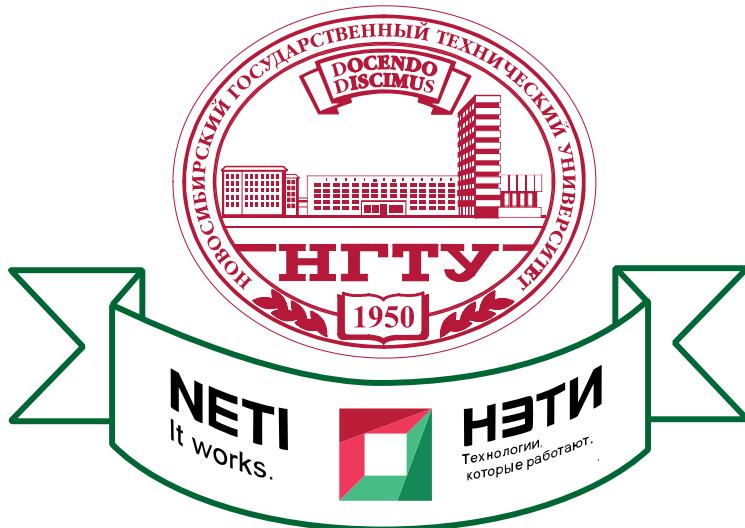


Министерство науки и высшего образования
Российской Федерации

Федеральное государственное бюджетное
образовательное учреждение высшего образования

«НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

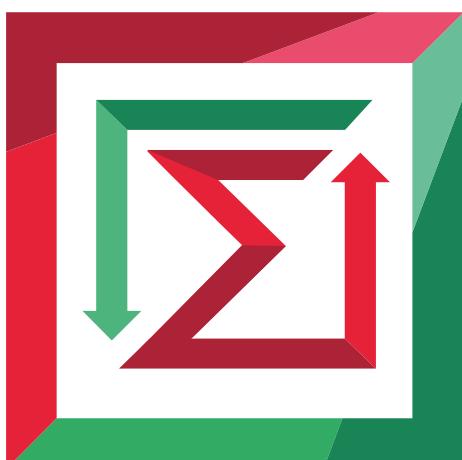


Теоретической и прикладной информатики

Расчетно-графическая задача

по дисциплине «Компьютерное моделирование»

МОДЕЛИРОВАНИЕ ВЫЧИСЛИТЕЛЬНОЙ СИСТЕМЫ



Факультет: ПМИ
Группа: ПМИ-02
Студенты: Сидоров Даниил,
Дюков Богдан
Вариант 6
Преподаватель: Карманов Виталий Сергеевич

Новосибирск

2026

1. Цель работы

Построить и исследовать модель вычислительной системы.

2. Описание системы

Вычислительная система предназначена для решения определенного типа задач и должна отличаться высокой надежностью. В систему входят 3 компьютера: один основной и два резервных. Время безотказной работы компьютера подчиняется экспоненциальному распределению со средним значением 500 часов. Считается, что с помощью системы автоматического переключения вышедший из строя компьютер мгновенно заменяется исправным (при холодном резервировании).

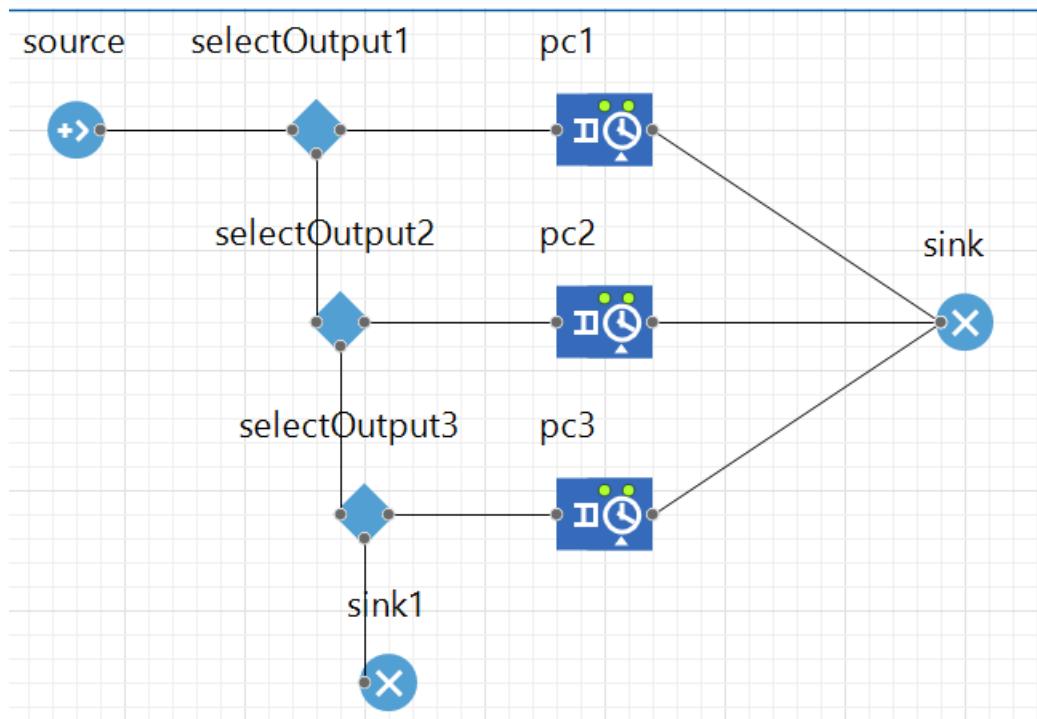
1. Методом имитационного моделирования определить вероятность безотказной работы системы через 100, 200, 300, ..., 1000 часов работы:

- при работе в режиме «холодного» резервирования;
- при работе в режиме «горячего» резервирования;

2. Определить вероятность безотказной работы системы в режиме с восстановлением, если интенсивность восстановления равна 24 часам, а время восстановления подчиняется экспоненциальному распределению.

3. Описание выполненных действий

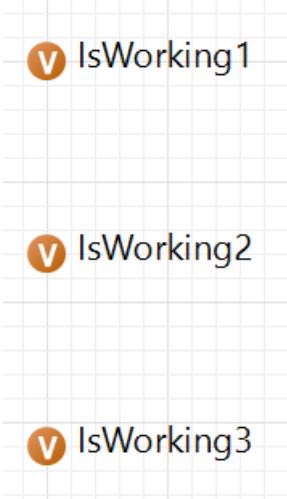
Создадим следующую модель:



Блоки Service представляют собой компьютеры. По условию задачи их 3.

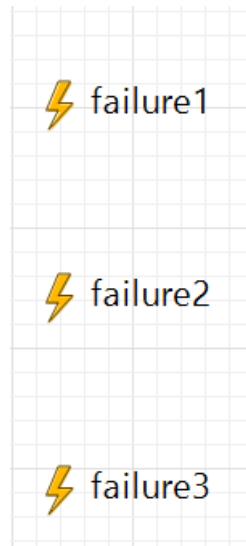
Чтобы наши задачи не приходили к неисправному компьютеру, использованы блоки SelectOutput.

Для работы SelectOutput необходимо было создать переменные, которые бы говорили, исправен компьютер или нет.



Если переменная IsWorking1 имеет значение false, то SelectOutput1 отправляет задачу на SelectOutput2, иначе задачу выполняет первый компьютер. Аналогично работает SelectOutput2. SelectOutput3, в случае неисправности третьего компьютера, отправляет задачу в sink1, но, так как наша система по условию должна быть безотказная, это означает, что система не справилась со своей работой.

Чтобы имитировать поломку компьютера, были использованы блоки Event:



Время безотказной работы компьютера подчиняется экспоненциальному распределению со средним значением 500 часов, значит таймаут выглядит следующим образом:

`exponential(1.0/500.0)`

Когда Event срабатывает, он меняет значение соответствующей переменной IsWorking на false, а также проверяется отказалась наша система или нет. Кроме того, для системы с восстановлением запускается событие “repair”.

⚡ failure1 - Событие

Имя: failure1 Отображать имя Исключить
Видимость: да
Тип события: По таймауту
Режим: "Ручной"
Таймаут: exponential(1.0/500.0) часы

Вести журнал в базе данных
[Вести журнал выполнения модели](#)

▼ Действие

```
IsWorking1 = false;
if (!IsWorking1&!IsWorking2&!IsWorking3)
{
    stop();
}
failure2.restart();
if(withRepair)
{
    repair1.restart();
}
```

► Описание

⚡ failure2 - Событие

Имя: failure2 Отображать имя Исключить

Видимость: да

Тип события: По таймауту

Режим: "Ручной"

Таймаут: exponential(1.0/500.0) часы

Вести журнал в базе данных
[Вести журнал выполнения модели](#)

▼ Действие

```
IsWorking2 = false;
if (!IsWorking1&!IsWorking2&!IsWorking3)
{
    stop();
}
failure3.restart();
if(withRepair)
{
    repair2.restart();
}
```

► Описание

⚡ failure3 - Событие

Имя:	failure3	<input checked="" type="checkbox"/> Отображать имя	<input type="checkbox"/> Исключить
Видимость:	<input checked="" type="radio"/> да		
Тип события:	По таймауту		
Режим:	"Ручной"		
Таймаут:	<input checked="" type="radio"/> exponential(1.0/500.0)		часы
<input checked="" type="checkbox"/> Вести журнал в базе данных Вести журнал выполнения модели			
▼ Действие <pre>IsWorking3 = false; if (!IsWorking1&!IsWorking2&!IsWorking3) { stop(); } if(withRepair) { repair3.restart(); }</pre>			
► Описание			

Для реализации безотказной работы системы в режиме с восстановлением используются блоки Event:



Интенсивность восстановления равна 24 часам, а время восстановления подчиняется экспоненциальному распределению. После восстановления событие меняет значение соответствующей переменной IsWorking на true. В случае холодного резервирования, после ремонта стараемся переключить задачи на основной компьютер и выключить резервные.

⚡ repair1 - Событие

Имя: repair1 Отображать имя Исключить
Видимость: да
Тип события: По таймауту
Режим: "Ручной"
Таймаут:  exponential(1.0/24.0) часы
 Вести журнал в базе данных
[Вести журнал выполнения модели](#)

▼ Действие

```
IsWorking1 = true;
failure1.restart();
if (!withHot)
{
    failure2.reset();
    failure3.reset();
}
```

► Описание

⚡ repair2 - Событие

Имя: repair2 Отображать имя Исключить
Видимость: да
Тип события: По таймауту
Режим: "Ручной"
Таймаут:  exponential(1.0/240.0) секунды
 Вести журнал в базе данных
[Вести журнал выполнения модели](#)

▼ Действие

```
IsWorking2 = true;
failure2.restart();
if (!withHot)
{
    if(IsWorking1 == true)
    {
        failure2.reset();
        failure3.reset();
    } else
    {
        failure1.reset();
        failure3.reset();
    }
}
```

► Описание

⚡ repair3 - Событие

Имя:	repair3	<input checked="" type="checkbox"/> Отображать имя	<input type="checkbox"/> Исключить
Видимость:	да		
Тип события:	По таймауту		
Режим:	"Ручной"		
Таймаут:	exponential(1.0/240.0)	секунды	

Вести журнал в базе данных
[Вести журнал выполнения модели](#)

▼ Действие

```
IsWorking3 = true;
failure3.restart();
if (!withHot)
{
    if(IsWorking1 == true)
    {
        failure2.reset();
        failure3.reset();
    } else if (IsWorking2 == true)
    {
        failure1.reset();
        failure3.reset();
    } else
    {
        failure1.reset();
        failure2.reset();
    }
}
```

► Описание

Так как все события запускаются в ручном режиме, требуется событие, которое запустит другие события при запуске симуляции.



⚡ start - Событие

Имя: Отображать имя Исключить

Видимость: да

Тип события:

Режим:

Использовать модельное время Использовать календарные даты

Время срабатывания (абсолютное):

Время срабатывания:

Вести журнал в базе данных
[Вести журнал выполнения модели](#)

▼ Действие

```
failure1.restart();
if(withHot)
{
    failure2.restart();
    failure3.restart();
}
```

► Описание

Так как наша состоит в определении вероятности безотказной работы системы, нам необходимо получить множество результатов. Чтобы не запускать симуляцию каждый раз, будем записывать время безотказной работы в excel и обнулять наши события при помощи функции stop().

excelFile - Файл Excel

Имя: Отображать имя Исключить

Видимость: да

Файл:

▼ Специфические

- Загружать содержимое при запуске модели
- Сохранять при прекращении работы модели
- Сохранять при сохранении состояния модели

F stop - Функция

Имя: Отображать имя Исключить

Видимость: да

Действие (не возвращает ничего)

Возвращает значение

Аргументы

Тело функции

```
excelFile.setCellValue((time()/3600)-lastTime, "Лист1", m, 1);
if(m < n)
{
    IsWorking1 = true;
    IsWorking2 = true;
    IsWorking3 = true;
    lastTime = time()/3600;
    m=m+1;
    failure1.restart();
    if(withHot)
    {
        failure2.restart();
        failure3.restart();
    }
    if(withRepair)
    {
        repair1.reset();
        repair2.reset();
        repair3.reset();
    }
}
else
{
    stopSimulation();
}
```

Специфические

Уровень доступа:

Единицы измерения (система динамики):

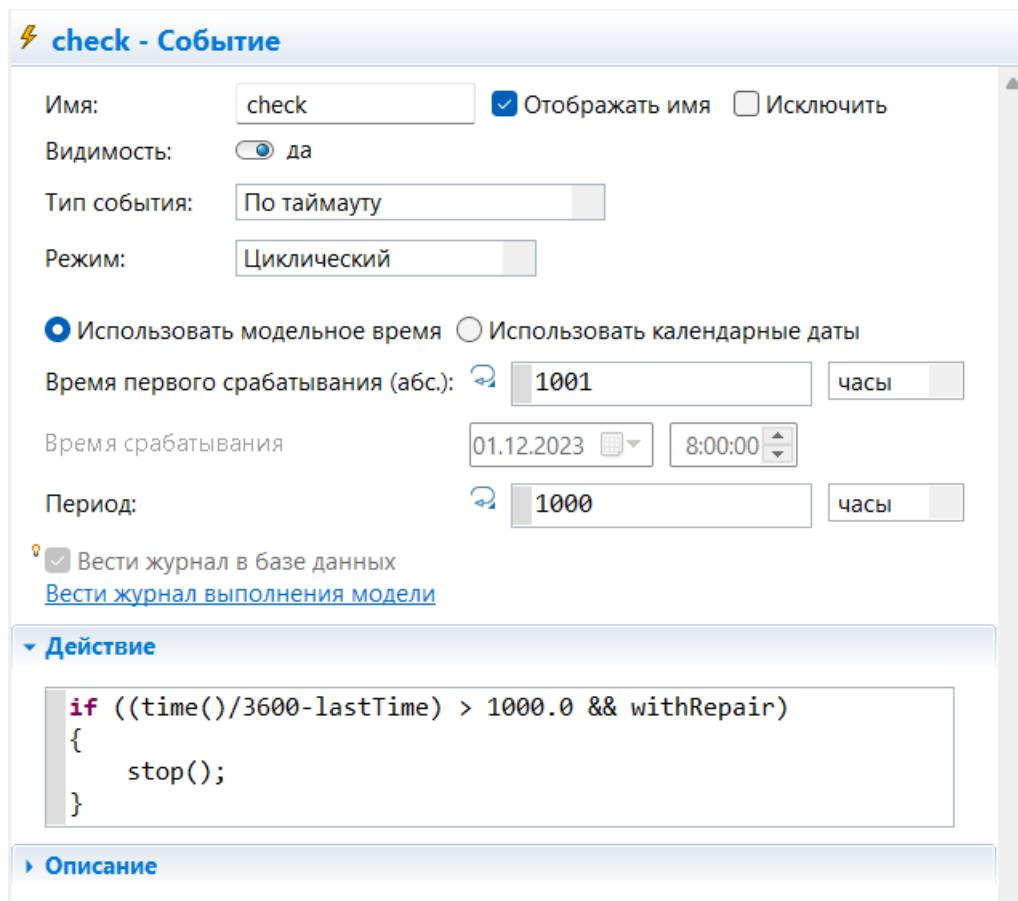
Количество циклов указывается в переменной n, текущий цикл в m, последнее время безотказной работы в lastTime.

v n

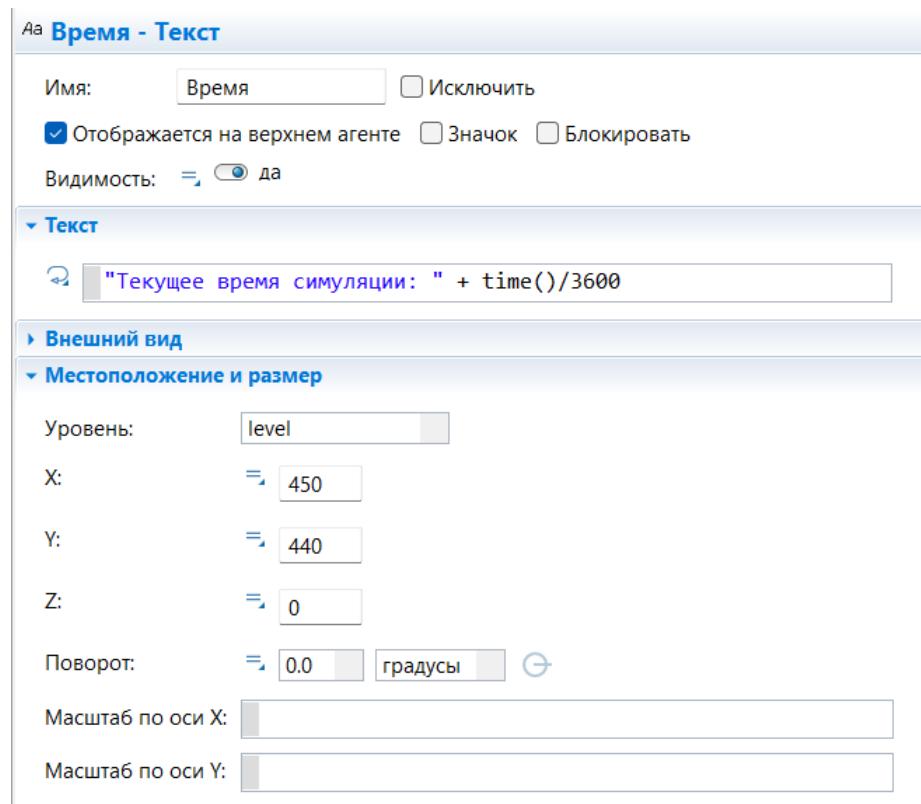
v m

v lastTime

Система с восстановлением имеет шанс проработать огромное количество времени, поэтому добавим событие, которое будет завершать работу системы, если она отработает срок свыше 2000 часов.



Используем динамический текст, чтобы видеть текущее время симуляции и время работы системы.



Настройка элемента "Время1 - Текст":

Имя: Время1 Исключить
 Отображается на верхнем агенте Значок Блокировать
 Видимость: = да

Текст: "Текущее время работы: " + (time() / 3600 - lastTime)

Внешний вид

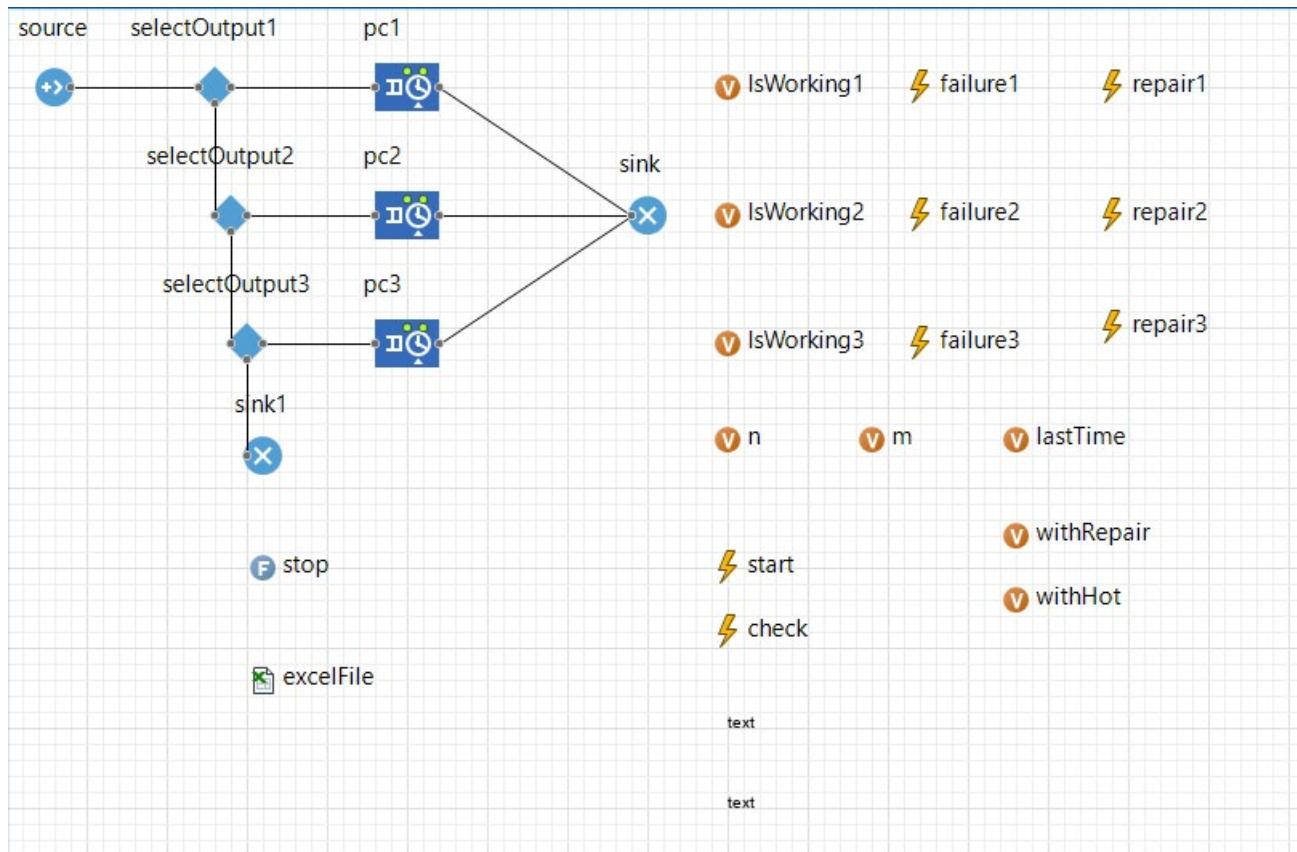
Местоположение и размер

- Уровень: level
- X: 450
- Y: 490
- Z: 0
- Поворот: 0.0 градусы
- Масштаб по оси X:
- Масштаб по оси Y:

Специфические

Описание

В итоге наша модель имеет следующий вид:



4. Полученные результаты и их анализ

Вычислим вероятность безотказной работы системы в режиме холодного резервирования при n=1000:

Через 100 часов	Через 200 часов	Через 300 часов	Через 400 часов	Через 500 часов	Через 600 часов	Через 700 часов	Через 800 часов	Через 900 часов	Через 1000 часов
0,999	0,993	0,987	0,962	0,93	0,899	0,858	0,807	0,764	0,705

Вычислим вероятность безотказной работы системы в режиме горячего резервирования при n=1000. Различие между холодным и горячим резервированием заключается в том, что при холодном работает только один компьютер, другие выключены, а при горячем работают все компьютеры, но выполняет задачи только один. Значит при горячем резервировании все компьютеры будут иметь возможность выйти из строя, даже если не получают задачи.

Через 100 часов	Через 200 часов	Через 300 часов	Через 400 часов	Через 500 часов	Через 600 часов	Через 700 часов	Через 800 часов	Через 900 часов	Через 1000 часов
0,997	0,966	0,912	0,839	0,753	0,688	0,594	0,523	0,446	0,379

Вычислим вероятность безотказной работы системы в режиме холодного резервирования с восстановлением при n=1000:

Через 100 часов	Через 200 часов	Через 300 часов	Через 400 часов	Через 500 часов	Через 600 часов	Через 700 часов	Через 800 часов	Через 900 часов	Через 1000 часов
1	1	1	1	1	1	1	0.998	0.998	0.998

Вычислим вероятность безотказной работы системы в режиме горячего резервирования с восстановлением при n=1000:

Через 100 часов	Через 200 часов	Через 300 часов	Через 400 часов	Через 500 часов	Через 600 часов	Через 700 часов	Через 800 часов	Через 900 часов	Через 1000 часов
1	0,999	0,997	0,997	0,996	0,995	0,995	0,994	0,993	0,993

5. Вывод

Видим, что вероятность безотказной работы при холодном резервировании выше, чем при горячем. Это логично, ведь преимущество горячего резервирования заключается в моментальном переключении, а не надежности. Так как у нас по условию при холодном резервировании неисправный компьютер мгновенно заменяется исправным, горячее резервирование не имеет смысла.

Вероятность безотказной работы вычислительной системы с восстановлением близка к 1. Это получается из-за того, что время ремонта значительно меньше времени безотказной работы($\text{exponential}(1.0/500.0) > \text{exponential}(1.0/24.0)$). Если увеличивать время ремонта, то вероятность безотказной работы будет уменьшаться.