]Ligatures=TeX,Scale=1

Untitled

Contents

Intro:

Although not strictly *required* to be encrypted, the GDPR strongly recommends encryption of personally identifiable information. Specifically, it is recommended to use a strong encryption standard such as AES. In the past, AES encryption was done via ECB (Electronic Code Book). This meant that identical inputs would produce identical outputs, thus allowing easy indexing and collision detection. This is no longer done, because it allows information to become known about the cleartext, even without decryption. For an example of this, see this stackoverflow answer about the 2013 Adobe password leak.

Current modes of operation (such as CBC) use a random IV (initialization vector) to ensure diffusion. This means that we cannot know if a given ciphertext matches our plaintext without decryption of the ciphertext. Because of this, in order to find a given email address, we must first decrypt half of the data, on average.

In response to this, one stackexchange contributor suggests using a high-collision hash algorithm in order to narrow down which rows could potentially match our cleartext. He helpfully provides the following algorithm:

```
E = Email Address
Choose N s.t. 2^(N+3) <= NUM_USERS < 2^(N+4), N>=0)
H(E) = (Sum each ASCII value of E)%(2^N)
```

The tests

I have prepared several potential implementations of this suggestion, and ran tests on them in order to discover information about performance when inserting and updating data. These implementations will either redo all hashes when a threshold is reached and `N` is changed, or store the value of `N` at the time of the last update to the row.

Generation

Data was created using this C# source code. For convenience's sake, it was also included at the end of this document The original intention was to run 5 iterations each of all combinations of the following datapoints:

- Initial user count
  - 1,000
  - 10,000
  - 100,000
- Users to add or update per test
  - 10
  - 20
  - 30
- Algorithm
  - Control: Using no encryption
  - Control: using a fixed number of buckets
  - Using `SELECT COUNT(*)` for updates to BucketCount, storing BucketExponent in the entity
  - Using a summary table for updates to BucketCount, storing BucketExponent in the entity
  - Using autoincremented row numbers

- Using SELECT COUNT(*) for updates to BucketCount, updating all bucket numbers after threshold is passed

However, due to time constrains I ended the tests prematurely.

List of completed tests

The following tests were completed:

(click to expand)

Algorithm	Test Size	Initial user count
"Control: Using no encryption"	10	1000
"Control: using a fixed number of buckets"	10	1000
"Using SELECT COUNT(*) for updates to BucketCount, storing BucketExponent in the entity"	10	1000
"Using a summary table for updates to BucketCount, storing BucketExponent in the entity"	10	1000
"Using autoincremented row numbers"	10	1000
"Using SELECT COUNT(*) for updates to BucketCount, updating all bucket numbers after threshold is passed"	10	1000
"Control: Using no encryption"	20	1000
"Control: using a fixed number of buckets"	20	1000
"Using SELECT COUNT(*) for updates to BucketCount, storing BucketExponent in the entity"	20	1000
"Using a summary table for updates to BucketCount, storing BucketExponent in the entity"	20	1000
"Using autoincremented row numbers"	20	1000
"Using SELECT COUNT(*) for updates to BucketCount, updating all bucket numbers after threshold is passed"	20	1000
"Control: Using no encryption"	30	1000
"Control: using a fixed number of buckets"	30	1000
"Using SELECT COUNT(*) for updates to BucketCount, storing BucketExponent in the entity"	30	1000
"Using a summary table for updates to BucketCount, storing BucketExponent in the entity"	30	1000
"Using autoincremented row numbers"	30	1000
"Using SELECT COUNT(*) for updates to BucketCount, updating all bucket numbers after threshold is passed"	30	1000
"Control: Using no encryption"	10	10000
"Control: using a fixed number of buckets"	10	10000
"Using SELECT COUNT(*) for updates to BucketCount, storing BucketExponent in the entity"	10	10000
"Using a summary table for updates to BucketCount, storing BucketExponent in the entity"	10	10000
"Using autoincremented row numbers"	10	10000
"Using SELECT COUNT(*) for updates to BucketCount, updating all bucket numbers after threshold is passed"	10	10000
"Control: Using no encryption"	20	10000
"Control: using a fixed number of buckets"	20	10000
"Using SELECT COUNT(*) for updates to BucketCount, storing BucketExponent in the entity"	20	10000
"Using a summary table for updates to BucketCount, storing BucketExponent in the entity"	20	10000
"Using autoincremented row numbers"	20	10000

Algorithm	Test Size	Initial user count
“Using `SELECT COUNT(*)` for updates to BucketCount, updating all bucket numbers after threshold is passed”	20	10000
“Control: Using no encryption”	30	10000
“Control: using a fixed number of buckets”	30	10000
“Using `SELECT COUNT(*)` for updates to BucketCount, storing BucketExponent in the entity”	30	10000
“Using a summary table for updates to BucketCount, storing BucketExponent in the entity”	30	10000
“Using autoincremented row numbers”	30	10000
“Using `SELECT COUNT(*)` for updates to BucketCount, updating all bucket numbers after threshold is passed”	30	10000
“Control: Using no encryption”	10	100000
“Control: using a fixed number of buckets”	10	100000
“Using `SELECT COUNT(*)` for updates to BucketCount, storing BucketExponent in the entity”	10	100000
“Using a summary table for updates to BucketCount, storing BucketExponent in the entity”	10	100000
“Using autoincremented row numbers”	10	100000
“Using `SELECT COUNT(*)` for updates to BucketCount, updating all bucket numbers after threshold is passed”	10	100000
“Control: Using no encryption”	20	100000
“Control: using a fixed number of buckets”	20	100000
“Using `SELECT COUNT(*)` for updates to BucketCount, storing BucketExponent in the entity”	20	100000
“Using a summary table for updates to BucketCount, storing BucketExponent in the entity”	20	100000
“Using autoincremented row numbers”	20	100000
“Using `SELECT COUNT(*)` for updates to BucketCount, updating all bucket numbers after threshold is passed”	20	100000
“Control: Using no encryption”	30	100000
“Control: using a fixed number of buckets”	30	100000
“Using `SELECT COUNT(*)` for updates to BucketCount, storing BucketExponent in the entity”	30	100000
“Using a summary table for updates to BucketCount, storing BucketExponent in the entity”	30	100000
“Using autoincremented row numbers”	30	100000
“Using `SELECT COUNT(*)` for updates to BucketCount, updating all bucket numbers after threshold is passed”	30	100000
“Control: Using no encryption”	10	1000000
“Control: using a fixed number of buckets”	10	1000000
“Using `SELECT COUNT(*)` for updates to BucketCount, storing BucketExponent in the entity”	10	1000000

Results

Summarizing the outcome of these tests yields the following

```
data.table(a$AddUserTests,a$FindAndUpdateUserTests,a$UpdateUserTests) %>%
  summary %>%
  kable(col.names = c("Add User Tests, in tenths of a microsecond","Find and Update User Tests, in tenth
```

Add User Tests, in tenths of a microsecond	Find and Update User Tests, in tenths of a microsecond	Update User Tests, in tenths of a microsecond
Min. : 29238	Min. : 58453	Min. : 63826
1st Qu.: 197630	1st Qu.: 313647	1st Qu.: 287366
Median : 407353	Median : 466563	Median : 468675
Mean : 1023684	Mean : 1073516	Mean : 1079084
3rd Qu.: 953976	3rd Qu.: 1396193	3rd Qu.: 1476060
Max. :117401279	Max. :18636322	Max. :16647301

```
data.table(AddUserTests=a$AddUserTests,FindAndUpdateUserTests=a$FindAndUpdateUserTests,UpdateUserTests=a
  gather() -> gathered
ggplot(gathered, aes(x=gathered$key)) +
  geom_boxplot(aes(y=gathered$value, fill=gathered$key), notch=TRUE, outlier.shape=16, outlier.size=2)+
  scale_y_continuous(trans="log10") +
  scale_fill_brewer(palette="Dark2") +
  labs(y="Time, in tenths of a microsecond", x="Test type") +
  guides(fill="none")
```

```
## Warning: Use of `gathered$value` is discouraged. Use `value` instead.
```

```
## Warning: Use of `gathered$key` is discouraged. Use `key` instead.
## Use of `gathered$key` is discouraged. Use `key` instead.
```

a_files/figure-latex/summaryA-1.pdf

AddUserTests

There is extreme variance in AddUserTests especially, with 2 instances taking as long as 10 seconds. The greatest interquartile range came from updating all buckets after the threshold was passed, though most extreme outliers were in storing the value of N while using select(*) to calculate it.

```
data.table(context=a$Context, tests=a$AddUserTests) %>%
  group_by(context) -> grouped
skim(grouped) %>%
  select(-skim_type, -skim_variable, -n_missing, -complete_rate, -numeric.hist, -numeric.sd) %>%
  rotate_df  %>%
  row_to_names(row_number=1) -> transposed

rownames(transposed) <- c("mean, in tenths of a microsecond","min, in tenths of a microsecond","1st Qu,
kable(transposed)
```

	Control: using a fixed number of buckets	Control: Using no encryption	Using `SELECT COUNT(*)` for updates to BucketCount, storing BucketExponent in the entity	Using `SELECT COUNT(*)` for updates to BucketCount, updating all bucket numbers after threshold is passed	Using a summary table for updates to BucketCount, storing BucketExponent in the entity	Using autoincremented row numbers
mean, in tenths of a microsecond	1492968.8	696863.4	1877877.2	418344.5	716880.3	883798.9
min, in tenths of a microsecond	79493	66429	173935	29238	178676	154126
1st Qu, in tenths of a microsecond	112726.75	97228.25	323748.50	41415.50	329448.50	307333.75
Median, in tenths of a microsecond	279094.5	236444.0	523350.5	143053.0	519436.5	522498.5
3rd Qu., in tenths of a microsecond	913436.2	737637.8	1333478.0	1051074.5	1278552.2	1672075.5
Max, in tenths of a microsecond	117401279	7167681	76707835	1805152	2368564	2289360

```
grouped%>%
  ggplot( aes(x=context)) +
  geom_boxplot(aes(y=tests, fill=context), notch=TRUE, outlier.shape=16, outlier.size=2)+
  scale_y_continuous(trans="log10") +
  scale_fill_brewer(palette="Dark2") +
  labs(y="Time, in tenths of a microsecond", x="Algorithm used") +
  scale_x_discrete(guide=guide_axis(n.dodge=3)) +
  theme(legend.position="bottom", legend.direction = "vertical", legend.title = element_blank(), axis.t
```

a_files/figure-latex/addUserAnalysis-1.pdf

Appendix: Listings

The following are the source code used to generate the data.

Formatting

Given output of Formatting as Parsing.txt, creates well-formed and pretty-printed Json

```
cat parsing.txt | tr "'" '"' | jq > output.json
```

Parsing

Given output of Generation, creates input for Formatting. Run it in bash via `bash awk -f Parsing.awk < output.txt > parsing.txt`

```
BEGIN{
    Context="";
    SeedCount=0;
    TestSize=0;
    ArrNo=0;
    Index=0;
    MaxIndex=0;
    delete Arr;

}

/^Saving/{next;}
/^Changing/{next;}

/^Iteration/{
    if(Index==MaxIndex){
        Index=0;
        ArrNo++;
    }
    Arr[ArrNo,Index] = $(NF - 1);
    Index++;
    next;
}

/^Seeding/{
    SeedCount = $(NF - 1);
    next;
}

/^Adding/{
    MaxIndex = $2;
    next;
}
```

```
NR==1{
    Context = $0;
    print "[";
    next;
}

{
    print "{";
    print "'Context': '" Context "',";
    print "'TestSize': " MaxIndex ",";
    print "'SeedCount' : " SeedCount ",";
    print "'AddUserTests':[";
    print Arr[0,0];
    for( i = 1; i < MaxIndex; i++){
        print "," Arr[0,i];
    }
    print "],";
    print "'FindAndUpdateUserTests':["
    print Arr[1,0];
    for( i = 1; i < MaxIndex; i++){
        print "," Arr[1,i];
    }

    print "],";
    print "'UpdateUserTests':["
    print Arr[2,0];
    for( i = 1; i < MaxIndex; i++){
        print "," Arr[2,i];
    }
    print "]},";

    Context=$0;
    SeedCount=0;
    TestSize=0;
    ArrNo=0;
    Index=0;
    MaxIndex=0;
    delete Arr;
}


END{
    print "{";
    print "'Context': '" Context "',";
    print "'TestSize': " MaxIndex ",";
    print "'SeedCount' : " SeedCount ",";
    print "'AddUserTests':[";
    print Arr[0,0];
    for( i = 1; i < MaxIndex; i++){
        print "," Arr[0,i];
    }
    print "],";
    print "'FindAndUpdateUserTests':["
    print Arr[1,0];
    for( i = 1; i < MaxIndex; i++){
        print "," Arr[1,i];
    }

    print "],";
    print "'UpdateUserTests':["
    print Arr[2,0];
    for( i = 1; i < MaxIndex; i++){
        print "," Arr[2,i];
    }
    print "]}]";
}
```

Generation

C# code used to create data. Also available at https://pastebin.com/BxwuFrsL

```
using System;
using System.Collections.Generic;
using System.Diagnostics;
using System.Security.Cryptography;
using Microsoft.EntityFrameworkCore;
using Microsoft.EntityFrameworkCore.Storage.ValueConversion;
using static NonCryptographicHelpers;
using static options;

var rand = new Random();
var watch = new Stopwatch();
var fs = new FileStream("output.txt", FileMode.Create);
var sw = new StreamWriter(fs);
Console.SetOut(sw);
for (int w = 0; w < 5; w++)
    for (int exp = 3; exp < 7; exp++)
    {
        for (int numUsers = 10; numUsers < 40; numUsers += 10)
        {
            int numberOfUsers = (int)Math.Pow(10, exp);
            Console.WriteLine("Control: Using no encryption");
            using (var context = new UnencryptedDbContext())
            {
                context.Database.EnsureDeleted();
                context.Database.EnsureCreated();
                Console.WriteLine($"Seeding DB with {numberOfUsers} users ");
                for (int i = 0; i < numberOfUsers; i++)
                {
                    context.Add(new BasicEntity
                    {
                        EmailAddress = $"{rand.Next()}-{rand.Next()}@example.com"
                    });
                }
                Console.WriteLine("Saving changes");
                context.SaveChanges();

                Console.WriteLine($"Adding {numUsers} users individually");
                for (int i = 0; i < numUsers; i++)
                {
                    watch.Restart();
                    watch.Start();
                    context.Add(new BasicEntity
                    {
                        EmailAddress = $"{rand.Next()}-{rand.Next()}@example.com"
                    });
                    context.SaveChanges();
                    watch.Stop();
                    Console.WriteLine($"Iteration {i} took {watch.ElapsedTicks} ticks");
                }
                Console.WriteLine($"Changing {numUsers} users email addresses, twice each");
                var ids = context.BasicEntities.Take(numUsers).Select(_ => _.Id).ToArray();
                for (int j = 0; j < 2; j++)
                    for (int i = 0; i < numUsers; i++)
                    {
                        watch.Restart();
                        watch.Start();
                        var u = context.BasicEntities.Find(ids[i]);
                        u.EmailAddress = $"{rand.Next()}-{rand.Next()}@example.com";
                        context.SaveChanges();
                        watch.Stop();
                        Console.WriteLine($"Iteration {j} on user {i} took {watch.ElapsedTicks} ticks")
                    }
            }
            Console.WriteLine("Control: using a fixed number of buckets");

            using (var context = new BucketedEncryptedDbContext())
            {
                int bucketExponent = GetBucketExponent(numberOfUsers);
                context.Database.EnsureDeleted();
                context.Database.EnsureCreated();
                Console.WriteLine($"Seeding DB with {numberOfUsers} users ");
                for (int i = 0; i < numberOfUsers; i++)
                {
                    context.Add(new BucketedEntity($"{rand.Next()}-{rand.Next()}@example.com", bucketExp
                }
                Console.WriteLine("Saving changes");
                context.SaveChanges();

                Console.WriteLine($"Adding {numUsers} users individually");
                for (int i = 0; i < numUsers; i++)
                {
                    watch.Restart();
                    watch.Start();
                    BucketedEntity entity;
                    bool conflict = false;
                    do
                    {
                        entity = new BucketedEntity($"{rand.Next()}-{rand.Next()}@example.com", bucketE
                        foreach (var other in context.BucketedEntities.Where(_ => _.BucketNo == entity.
                        {
                            if (other.EmailAddress == entity.EmailAddress)
                            {
                                Console.WriteLine("ERROR: CONFLICT. RESTARTING ITERATION.");
                                conflict = true;
                            }

                        }
                    } while (conflict);
                    context.Add(entity);
                    context.SaveChanges();
                    watch.Stop();
                    Console.WriteLine($"Iteration {i} took {watch.ElapsedTicks} ticks");
                }

                Console.WriteLine($"Changing {numUsers} users email addresses, twice each");
                var ids = context.BucketedEntities.Take(numUsers).Select(_ => _.Id).ToArray();
                for (int j = 0; j < 2; j++)
                    for (int i = 0; i < numUsers; i++)
                    {
                        watch.Restart();
                        watch.Start();
                        var u = context.BucketedEntities.Find(ids[i]);

                        BucketedEntity entity;
                        bool conflict = false;
                        do
                        {
                            entity = new BucketedEntity($"{rand.Next()}-{rand.Next()}@example.com", buc
                            foreach (var other in context.BucketedEntities.Where(_ => _.BucketNo == enti
                            {
                                if (other.EmailAddress == entity.EmailAddress)
                                {
                                    Console.WriteLine("ERROR: CONFLICT. RESTARTING ITERATION.");
                                    conflict = true;
                                }

                            }
                        } while (conflict);
                        u.EmailAddress = entity.EmailAddress;
                        u.BucketNo = entity.BucketNo;
                        context.SaveChanges();
                        context.SaveChanges();
                        watch.Stop();
                        Console.WriteLine($"Iteration {j} on user {i} took {watch.ElapsedTicks} ticks")
                    }
            }

            Console.WriteLine("Using `SELECT COUNT(*)` for updates to BucketCount, storing BucketExpone

            using (var context = new BucketExponentStoringDbContext())
            {
                context.Database.EnsureDeleted();
                context.Database.EnsureCreated();
                Console.WriteLine($"Seeding DB with {numberOfUsers} users ");
                for (int i = 0; i < numberOfUsers; i++)
                {
```

```
    context.Add(new BucketExponentStoringBucketedEntity($"{rand.Next()}-{rand.Next()}@e
}
Console.WriteLine("Saving changes");
context.SaveChanges();

Console.WriteLine($"Adding {numUsers} users individually");
for (int i = 0; i < numUsers; i++)
{
    watch.Restart();
    watch.Start();
    BucketExponentStoringBucketedEntity entity;
    bool conflict = false;
    var counts = context.BucketExponentStoringBucketedEntities.Select(_ => _.BucketExpo
    var currentCount = context.BucketExponentStoringBucketedEntities.Count();
    do
    {
        entity = new BucketExponentStoringBucketedEntity($"{rand.Next()}-{rand.Next()}@
        var buckets = counts.Select(_ => new BucketExponentStoringBucketedEntity(entity
        foreach (var b in buckets)
        {
            var others = context.BucketExponentStoringBucketedEntities.Where(_ => _.Buc
            foreach (var o in others)
            {
                if (o.EmailAddress == entity.EmailAddress)
                {
                    conflict = true;
                    break;
                }
            }
            if (conflict) { break; }
        }
    } while (conflict);
    context.Add(entity);
    context.SaveChanges();
    watch.Stop();
    Console.WriteLine($"Iteration {i} took {watch.ElapsedTicks} ticks");
}
Console.WriteLine($"Changing {numUsers} users email addresses, twice each");
var ids = context.BucketExponentStoringBucketedEntities.Take(numUsers).Select(_ => _.Id
for (int j = 0; j < 2; j++)
    for (int i = 0; i < numUsers; i++)
    {
        watch.Restart();
        watch.Start();
        var u = context.BucketExponentStoringBucketedEntities.Find(ids[i]);
        BucketExponentStoringBucketedEntity entity;
        bool conflict = false;
        var counts = context.BucketExponentStoringBucketedEntities.Select(_ => _.Bucket
        var currentCount = context.BucketExponentStoringBucketedEntities.Count();
        do
        {
            entity = new BucketExponentStoringBucketedEntity($"{rand.Next()}-{rand.Next
            var buckets = counts.Select(_ => new BucketExponentStoringBucketedEntity(en
            foreach (var b in buckets)
```

```
                {
                    var others = context.BucketExponentStoringBucketedEntities.Where(_ => _
                    foreach (var o in others)
                    {
                        if (o.EmailAddress == entity.EmailAddress)
                        {
                            conflict = true;
                            break;
                        }
                    }
                    if (conflict) { break; }
                }
            } while (conflict);
            u.BucketExponent = entity.BucketExponent;
            u.BucketNo = entity.BucketNo;
            u.EmailAddress = entity.EmailAddress;
            context.SaveChanges();
            watch.Stop();
            Console.WriteLine($"Iteration {j} on user {i} took {watch.ElapsedTicks} ticks")
        }
}
Console.WriteLine("Using a summary table for updates to BucketCount, storing BucketExponent

using (var context = new KeyValuedBucketExponentStoringDbContext())
{
    context.Database.EnsureDeleted();
    context.Database.EnsureCreated();
    Console.WriteLine($"Seeding DB with {numberOfUsers} users ");
    for (int i = 0; i < numberOfUsers; i++)
    {
        context.Add(new BucketExponentStoringBucketedEntity($"{rand.Next()}-{rand.Next()}@e
    }
    context.KeyValues.Single().value = numberOfUsers;
    Console.WriteLine("Saving changes");
    context.SaveChanges();

    Console.WriteLine($"Adding {numUsers} users individually");
    for (int i = 0; i < numUsers; i++)
    {
        watch.Restart();
        watch.Start();
        BucketExponentStoringBucketedEntity entity;
        bool conflict = false;
        var counts = context.BucketExponentStoringBucketedEntities.Select(_ => _.BucketExpo
        var currentCount = context.KeyValues.Single().value;
        do
        {
            entity = new BucketExponentStoringBucketedEntity($"{rand.Next()}-{rand.Next()}@
            var buckets = counts.Select(_ => new BucketExponentStoringBucketedEntity(entity
            foreach (var b in buckets)
            {
                var others = context.BucketExponentStoringBucketedEntities.Where(_ => _.Buc
                foreach (var o in others)
                {
                    if (o.EmailAddress == entity.EmailAddress)
                    {
                        conflict = true;
                        break;
                    }
                }
                if (conflict) { break; }
            }
        } while (conflict);
        context.Add(entity);
        context.KeyValues.Single().value++;
        context.SaveChanges();
        watch.Stop();
        Console.WriteLine($"Iteration {i} took {watch.ElapsedTicks} ticks");
    }
    Console.WriteLine($"Changing {numUsers} users email addresses, twice each");
    var ids = context.BucketExponentStoringBucketedEntities.Take(numUsers).Select(_ => _.Id
    for (int j = 0; j < 2; j++)
        for (int i = 0; i < numUsers; i++)
        {
            watch.Restart();
            watch.Start();
            var u = context.BucketExponentStoringBucketedEntities.Find(ids[i]);
            BucketExponentStoringBucketedEntity entity;
            bool conflict = false;
            var counts = context.BucketExponentStoringBucketedEntities.Select(_ => _.Bucket
            var currentCount = context.KeyValues.Single().value;
            do
            {
                entity = new BucketExponentStoringBucketedEntity($"{rand.Next()}-{rand.Next
                var buckets = counts.Select(_ => new BucketExponentStoringBucketedEntity(en
                foreach (var b in buckets)
                {
                    var others = context.BucketExponentStoringBucketedEntities.Where(_ => _
                    foreach (var o in others)
                    {
                        if (o.EmailAddress == entity.EmailAddress)
                        {
                            conflict = true;
                            break;
                        }
                    }
                    if (conflict) { break; }
                }
            } while (conflict);
            u.BucketExponent = entity.BucketExponent;
            u.BucketNo = entity.BucketNo;
            u.EmailAddress = entity.EmailAddress;
            context.SaveChanges();
            watch.Stop();
            Console.WriteLine($"Iteration {j} on user {i} took {watch.ElapsedTicks} ticks")
        }
}
```

```
Console.WriteLine("Using autoincremented row numbers");

using (var context = new RowCountedBucketExponentStoringDbContext())
{
    context.Database.EnsureDeleted();
    context.Database.EnsureCreated();
    Console.WriteLine($"Seeding DB with {numberOfUsers} users ");
    for (int i = 0; i < numberOfUsers; i++)
    {
        context.Add(new RowCountedBucketExponentStoringBucketedEntity($"{rand.Next()}-{rand
    }
    Console.WriteLine("Saving changes");
    context.SaveChanges();

    Console.WriteLine($"Adding {numUsers} users individually");
    for (int i = 0; i < numUsers; i++)
    {
        watch.Restart();
        watch.Start();
        RowCountedBucketExponentStoringBucketedEntity entity;
        bool conflict = false;
        var counts = context.RowCountedBucketExponentStoringBucketedEntities.Select(_ => _.
        var currentCount = context.RowCountedBucketExponentStoringBucketedEntities.Max(_ =>
        do
        {
            entity = new RowCountedBucketExponentStoringBucketedEntity($"{rand.Next()}-{ran
            var buckets = counts.Select(_ => new RowCountedBucketExponentStoringBucketedEnt
            foreach (var b in buckets)
            {
                var others = context.RowCountedBucketExponentStoringBucketedEntities.Where(
                foreach (var o in others)
                {
                    if (o.EmailAddress == entity.EmailAddress)
                    {
                        conflict = true;
                        break;
                    }
                }
                if (conflict) { break; }
            }
        } while (conflict);
        context.Add(entity);
        context.SaveChanges();
        watch.Stop();
        Console.WriteLine($"Iteration {i} took {watch.ElapsedTicks} ticks");
    }
    Console.WriteLine($"Changing {numUsers} users email addresses, twice each");
    var ids = context.RowCountedBucketExponentStoringBucketedEntities.Take(numUsers).Select
    for (int j = 0; j < 2; j++)
        for (int i = 0; i < numUsers; i++)
        {
            watch.Restart();
            watch.Start();
            var u = context.RowCountedBucketExponentStoringBucketedEntities.Find(ids[i]);
            RowCountedBucketExponentStoringBucketedEntity entity;
            bool conflict = false;
            var currentCount = context.RowCountedBucketExponentStoringBucketedEntities.Max(
            var counts = context.RowCountedBucketExponentStoringBucketedEntities.Select(_ =
            do
            {
                entity = new RowCountedBucketExponentStoringBucketedEntity($"{rand.Next()}-
                var buckets = counts.Select(_ => new RowCountedBucketExponentStoringBucketed
                foreach (var b in buckets)
                {
                    var others = context.RowCountedBucketExponentStoringBucketedEntities.Wh
                    foreach (var o in others)
                    {
                        if (o.EmailAddress == entity.EmailAddress)
                        {
                            conflict = true;
                            break;
                        }
                    }
                    if (conflict) { break; }
                }
            } while (conflict);
            u.BucketExponent = entity.BucketExponent;
            u.BucketNo = entity.BucketNo;
            u.EmailAddress = entity.EmailAddress;
            context.SaveChanges();
            watch.Stop();
            Console.WriteLine($"Iteration {j} on user {i} took {watch.ElapsedTicks} ticks")
        }
}

Console.WriteLine("Using `SELECT COUNT(*)` for updates to BucketCount, updating all bucket 

using (var context = new TimestampedDbContext())
{
    Console.WriteLine($"Seeding DB with {numberOfUsers} users ");
    context.Database.EnsureDeleted();
    context.Database.EnsureCreated();
    for (int i = 0; i < numberOfUsers; i++)
    {

        int bucketExponent = GetBucketExponent(numberOfUsers);
        context.Add(new TimestampedBucketedEntity($"{rand.Next()}-{rand.Next()}@example.com
        context.SaveChanges();
    }
    Console.WriteLine("Saving changes");
    context.SaveChanges();

    Console.WriteLine($"Adding {numUsers} users individually");
    for (int i = 0; i < numUsers; i++)
    {
        watch.Restart();
        watch.Start();
        TimestampedBucketedEntity entity;
        bool conflict = false;
        var currentCount = context.TimestampedBucketedEntity.Count();
        int bucketExponent = GetBucketExponent(currentCount + 1);
        if (GetBucketExponent(currentCount) != GetBucketExponent(currentCount - 1))
        {
            foreach (var v in context.TimestampedBucketedEntity)
            {
                v.EmailAddress = v.EmailAddress;
                v.BucketNo = new BucketedEntity(v.EmailAddress, bucketExponent).BucketNo;
                v.EmailUpdated = DateTime.UtcNow;
            }
        }
        do
        {
            entity = new TimestampedBucketedEntity($"{rand.Next()}-{rand.Next()}@example.co
            foreach (var other in context.TimestampedBucketedEntity.Where(_ => _.BucketNo =
            {
                if (other.EmailAddress == entity.EmailAddress)
                {
                    Console.WriteLine("ERROR: CONFLICT. RESTARTING ITERATION.");
                    conflict = true;
                }

            }
        } while (conflict);
        context.SaveChanges();
        watch.Stop();
        Console.WriteLine($"Iteration {i} took {watch.ElapsedTicks} ticks");
    }


    Console.WriteLine($"Changing {numUsers} users email addresses, twice each");
    var ids = context.TimestampedBucketedEntity.Take(numUsers).Select(_ => _.Id).ToArray();
    for (int j = 0; j < 2; j++)
        for (int i = 0; i < numUsers; i++)
        {
            watch.Restart();
            watch.Start();
            var currentCount = context.TimestampedBucketedEntity.Count();
            var bucketExponent = GetBucketExponent(currentCount);
            var u = context.TimestampedBucketedEntity.Find(ids[i]);

            TimestampedBucketedEntity entity;
            bool conflict = false;
            do
            {
                entity = new TimestampedBucketedEntity($"{rand.Next()}-{rand.Next()}@exampl
                foreach (var other in context.TimestampedBucketedEntity.Where(_ => _.Bucket
                {
                    if (other.EmailAddress == entity.EmailAddress)
                    {
                        Console.WriteLine("ERROR: CONFLICT. RESTARTING ITERATION.");
                        conflict = true;
```

```
                                }

                            }
                        } while (conflict);
                        u.EmailAddress = entity.EmailAddress;
                        u.BucketNo = entity.BucketNo;
                        context.SaveChanges();
                        context.SaveChanges();
                        watch.Stop();
                        Console.WriteLine($"Iteration {j} on user {i} took {watch.ElapsedTicks} ticks")
                    }
            }
        }
    }
sw.Close();

/* RowCountedTimestampedDbContext
using (var context = new RowCountedTimestampedDbContext())
{
    context.Database.EnsureDeleted();
    context.Database.EnsureCreated();
    for (int i = 0; i < 333; i++)
    {
        int count = 0;
        try{
            count = context.RowCountedTimestampedBucketedEntity.Select(_ => _.EntryNo).Max();
        }catch{}
        int bucketExponent = GetBucketExponent(count);
        if(GetBucketExponent(count) != GetBucketExponent(count - 1)){
            foreach(var v in context.RowCountedTimestampedBucketedEntity){
                v.EmailAddress = v.EmailAddress;
                v.BucketNo = new BucketedEntity(v.EmailAddress, bucketExponent).BucketNo;
                v.EmailUpdated = DateTime.UtcNow;
            }
        }
        context.Add(new RowCountedTimestampedBucketedEntity($"{i}_{rand.Next()}@example.com", bucketExp
        context.SaveChanges();
    }
}*/


/* save to TimestampedDbContext
using (var context = new TimestampedDbContext())
{
    context.Database.EnsureDeleted();
    context.Database.EnsureCreated();
    for (int i = 0; i < 333; i++)
    {
        int bucketExponent = GetBucketExponent(context.TimestampedBucketedEntity.Count());
        var count = context.TimestampedBucketedEntity.Count();
        if(GetBucketExponent(count) != GetBucketExponent(count - 1)){
            foreach(var v in context.TimestampedBucketedEntity){
                v.EmailAddress = v.EmailAddress;
                v.BucketNo = new BucketedEntity(v.EmailAddress, bucketExponent).BucketNo;
                v.EmailUpdated = DateTime.UtcNow;
            }
        }
        context.Add(new TimestampedBucketedEntity($"{i}_{rand.Next()}@example.com", bucketExponent));
        context.SaveChanges();
    }
}
*/

/* Fetch from bucketexponentstoringdbcontext

using (var context = new BucketExponentStoringDbContext())
using (var context2 = new BucketExponentStoringDbContext())
using (var context3 = new BucketExponentStoringDbContext())
{
    var exponents = context3.BucketExponentStoringBucketedEntities.Select(_ => _.BucketExponent).Distin
    foreach (var v in context.BucketExponentStoringBucketedEntities)
    {
        var bucketNumbers = exponents.Select(_ => new BucketedEntity(v.EmailAddress, _).BucketNo);
        Console.WriteLine($"EmailAddress: {v.EmailAddress}");
        Console.WriteLine($"BucketNo: {v.BucketNo}");
        Console.WriteLine($"Others: {context2.BucketExponentStoringBucketedEntities.Where(_ => bucketNu

    }
}*/

// DbContexts
public static class options
{
    public static string conn = "server=localhost;port=3306;database=tmp;uid=devuser;pwd=Pa55w0rd!";
    public static ServerVersion srvvrs = ServerVersion.Parse("8.0.27-mysql");
}
public class UnencryptedDbContext : DbContext
{
    public DbSet<BasicEntity> BasicEntities { get; set; }
    public UnencryptedDbContext() : base() { }

    protected override void OnConfiguring(DbContextOptionsBuilder options) => options.UseMySql(conn, srv

    protected override void OnModelCreating(ModelBuilder modelBuilder)
    {
        modelBuilder.Entity<BasicEntity>(e =>
        {
            e.HasIndex(e => e.EmailAddress)
            .IsUnique();
            e.Property(e => e.EmailAddress)
            .IsRequired();
        }
        );
    }
}
public class BasicEncryptedDbContext : DbContext
{
    public DbSet<BasicEntity> BasicEntities { get; set; }
    public BasicEncryptedDbContext() : base() { }
    protected override void OnConfiguring(DbContextOptionsBuilder options) => options.UseMySql(conn, sr

    protected override void OnModelCreating(ModelBuilder modelBuilder)
    {
        modelBuilder.Entity<BasicEntity>(e =>
        {
            e.HasIndex(e => e.EmailAddress)
            .IsUnique();
            e.Property(e => e.EmailAddress)
            .IsRequired().HasConversion<PersonalDataConverter>();
        });
    }
}

public class BucketedEncryptedDbContext : DbContext
{
    public DbSet<BucketedEntity> BucketedEntities { get; set; }
    public BucketedEncryptedDbContext() : base() { }

    protected override void OnConfiguring(DbContextOptionsBuilder options) => options.UseMySql(conn, sr

    protected override void OnModelCreating(ModelBuilder modelBuilder)
    {
        modelBuilder.Entity<BucketedEntity>(e =>
        {
            e.HasIndex(e => e.EmailAddress)
            .IsUnique();
            e.HasIndex(e => e.BucketNo);
            e.Property(e => e.EmailAddress)
            .IsRequired().HasConversion<PersonalDataConverter>();
        }
        );
    }
}


public class RowCountedBucketExponentStoringDbContext : DbContext
{
    public DbSet<RowCountedBucketExponentStoringBucketedEntity> RowCountedBucketExponentStoringBucketed
    public RowCountedBucketExponentStoringDbContext() : base() { }

    protected override void OnConfiguring(DbContextOptionsBuilder options) => options.UseMySql(conn, sr
    protected override void OnModelCreating(ModelBuilder modelBuilder)
    {
        modelBuilder.Entity<RowCountedBucketExponentStoringBucketedEntity>(e =>
        {
            e.HasIndex(e => e.EmailAddress)
            .IsUnique();
            e.HasIndex(e => e.BucketNo);
            e.HasIndex(e => e.BucketExponent);

            e.HasAlternateKey(e => e.EntryNo);
            e.Property(e => e.EntryNo).ValueGeneratedOnAdd();
            e.Property(e => e.EmailAddress)
            .IsRequired().HasConversion<PersonalDataConverter>();
        }
        );
    }
}

public class RowCountedTimestampedDbContext : DbContext
{
    public DbSet<RowCountedTimestampedBucketedEntity> RowCountedTimestampedBucketedEntity { get; set; }
    public RowCountedTimestampedDbContext() : base() { }

    protected override void OnConfiguring(DbContextOptionsBuilder options) => options.UseMySql(conn, sr
    protected override void OnModelCreating(ModelBuilder modelBuilder)
    {
        modelBuilder.Entity<RowCountedTimestampedBucketedEntity>(e =>
        {
            e.HasIndex(e => e.EmailAddress)
            .IsUnique();

            e.HasAlternateKey(e => e.EntryNo);
            e.Property(e => e.EntryNo).ValueGeneratedOnAdd(); e.HasIndex(e => e.BucketNo);
            e.Property(e => e.EmailUpdated).ValueGeneratedOnAddOrUpdate();
            e.Property(e => e.EmailAddress)
            .IsRequired().HasConversion<PersonalDataConverter>();
        }
        );
    }
}

public class RowCountedBucketedDbContext : DbContext
{
    public DbSet<RowCountedBucketedEntity> RowCountedBucketedEntities { get; set; }
    public RowCountedBucketedDbContext() : base() { }

    protected override void OnConfiguring(DbContextOptionsBuilder options) => options.UseMySql(conn, sr

    protected override void OnModelCreating(ModelBuilder modelBuilder)
    {
        modelBuilder.Entity<RowCountedBucketedEntity>(e =>
        {
            e.HasIndex(e => e.EmailAddress)
            .IsUnique();
            e.HasIndex(e => e.BucketNo);
            e.HasAlternateKey(e => e.EntryNo);
            e.Property(e => e.EntryNo).ValueGeneratedOnAdd();
            e.Property(e => e.EmailAddress)
            .IsRequired().HasConversion<PersonalDataConverter>();
        }
        );
    }
}
```

```
public class BucketExponentStoringDbContext : DbContext
{
    public DbSet<BucketExponentStoringBucketedEntity> BucketExponentStoringBucketedEntities { get; set;
    public BucketExponentStoringDbContext() : base() { }

    protected override void OnConfiguring(DbContextOptionsBuilder options) => options.UseMySql(conn, srv
    protected override void OnModelCreating(ModelBuilder modelBuilder)
    {
        modelBuilder.Entity<BucketExponentStoringBucketedEntity>(e =>
        {
            e.HasIndex(e => e.EmailAddress)
            .IsUnique();
            e.HasIndex(e => e.BucketNo);
            e.HasIndex(e => e.BucketExponent);
            e.Property(e => e.EmailAddress)
            .IsRequired().HasConversion<PersonalDataConverter>();
        }
        );
    }
}

public class TimestampedDbContext : DbContext
{
    public DbSet<TimestampedBucketedEntity> TimestampedBucketedEntity { get; set; }
    public TimestampedDbContext() : base() { }

    protected override void OnConfiguring(DbContextOptionsBuilder options) => options.UseMySql(conn, srv
    protected override void OnModelCreating(ModelBuilder modelBuilder)
    {
        modelBuilder.Entity<TimestampedBucketedEntity>(e =>
        {
            e.HasIndex(e => e.EmailAddress)
            .IsUnique();
            e.HasIndex(e => e.BucketNo);
            e.Property(e => e.EmailUpdated).ValueGeneratedOnAddOrUpdate();
            e.Property(e => e.EmailAddress)
            .IsRequired().HasConversion<PersonalDataConverter>();
        }
        );
    }
}


public class KeyValuedBucketExponentStoringDbContext : DbContext
{
    public DbSet<BucketExponentStoringBucketedEntity> BucketExponentStoringBucketedEntities { get; set;
    public DbSet<KeyValues> KeyValues { get; set; }
    public KeyValuedBucketExponentStoringDbContext() : base() { }

    protected override void OnConfiguring(DbContextOptionsBuilder options) => options.UseMySql(conn, srv
    protected override void OnModelCreating(ModelBuilder modelBuilder)
    {
        modelBuilder.Entity<KeyValues>(e =>
        {
            e.HasKey(k => k.Key);
            e.HasData(new KeyValues
            {
                Key = "UserCount",
                value = 0
            });
        });
        modelBuilder.Entity<BucketExponentStoringBucketedEntity>(e =>
        {
            e.HasIndex(e => e.EmailAddress)
            .IsUnique();
            e.HasIndex(e => e.BucketNo);
            e.HasIndex(e => e.BucketExponent);
            e.Property(e => e.EmailAddress)
            .IsRequired().HasConversion<PersonalDataConverter>();
        }
        );
    }
}

public class KeyValuedTimestampedDbContext : DbContext
{
    public DbSet<TimestampedBucketedEntity> TimestampedBucketedEntity { get; set; }

    public DbSet<KeyValues> KeyValues { get; set; }
    public KeyValuedTimestampedDbContext() : base() { }

    protected override void OnConfiguring(DbContextOptionsBuilder options) => options.UseMySql(conn, sr
    protected override void OnModelCreating(ModelBuilder modelBuilder)
    {
        modelBuilder.Entity<KeyValues>(e =>
        {
            e.HasKey(k => k.Key);
            e.HasData(new KeyValues
            {
                Key = "UserCount",
                value = 0
            });
        });
        modelBuilder.Entity<TimestampedBucketedEntity>(e =>
        {
            e.HasIndex(e => e.EmailAddress)
            .IsUnique();
            e.HasIndex(e => e.BucketNo);
            e.Property(e => e.EmailUpdated).ValueGeneratedOnAddOrUpdate();
            e.Property(e => e.EmailAddress)
            .IsRequired().HasConversion<PersonalDataConverter>();
        }
        );
    }
}

public class KeyValuedBucketedDbContext : DbContext
{
    public DbSet<BucketedEntity> BucketedEntities { get; set; }
    public KeyValuedBucketedDbContext() : base() { }

    public DbSet<KeyValues> KeyValues { get; set; }
    protected override void OnConfiguring(DbContextOptionsBuilder options) => options.UseMySql(conn, sr

    protected override void OnModelCreating(ModelBuilder modelBuilder)
    {
        modelBuilder.Entity<KeyValues>(e =>
        {
            e.HasKey(k => k.Key);
            e.HasData(new KeyValues
            {
                Key = "UserCount",
                value = 0
            });
        });
        modelBuilder.Entity<BucketedEntity>(e =>
        {
            e.HasIndex(e => e.EmailAddress)
            .IsUnique();
            e.HasIndex(e => e.BucketNo);
            e.Property(e => e.EmailAddress)
            .IsRequired().HasConversion<PersonalDataConverter>();
        }
        );
    }
}

// Entities
public class BasicEntity
{
    public Guid Id { get; set; }
    public string EmailAddress { get; set; } = "example@example.com";
}

public class TimestampedBucketedEntity : BucketedEntity
{
    public TimestampedBucketedEntity() { }
    public TimestampedBucketedEntity(string emailAddress, int bucketExponent) : base(emailAddress, bucke
    public DateTime EmailUpdated { get; set; } = DateTime.UtcNow;

}

public class BucketExponentStoringBucketedEntity : BucketedEntity
{

    public BucketExponentStoringBucketedEntity() { }
    public BucketExponentStoringBucketedEntity(string emailAddress, int bucketExponent) : base(emailAddr
    public int BucketExponent { get; set; }
}

public class BucketedEntity
{
    public BucketedEntity() { }
    public BucketedEntity(string emailAddress, int bucketExponent)
    {
        BucketNo = 0;
        foreach (char c in emailAddress.ToLowerInvariant())
        {
            BucketNo += (int)c;
        }
        BucketNo %= (int)Math.Pow(2, bucketExponent);
        EmailAddress = emailAddress;
    }
    public Guid Id { get; set; }
    public string EmailAddress { get; set; } = "example@example.com";
    public int BucketNo { get; set; } = 0;
}

public class RowCountedBasicEntity
{
    public int EntryNo { get; set; } = 0;
    public Guid Id { get; set; }
    public string EmailAddress { get; set; } = "example@example.com";
}

public class RowCountedTimestampedBucketedEntity : RowCountedBucketedEntity
{
    public RowCountedTimestampedBucketedEntity() { }
    public RowCountedTimestampedBucketedEntity(string emailAddress, int bucketExponent) : base(emailAdd
    public DateTime EmailUpdated { get; set; } = DateTime.UtcNow;

}

public class RowCountedBucketExponentStoringBucketedEntity : RowCountedBucketedEntity
{

    public RowCountedBucketExponentStoringBucketedEntity() { }
    public RowCountedBucketExponentStoringBucketedEntity(string emailAddress, int bucketExponent) : bas
    public int BucketExponent { get; set; }
}

public class RowCountedBucketedEntity
{
    public RowCountedBucketedEntity() { }
    public RowCountedBucketedEntity(string emailAddress, int bucketExponent)
    {
        BucketNo = 0;
        foreach (char c in emailAddress.ToLowerInvariant())
        {
            BucketNo += (int)c;
        }
        BucketNo %= (int)Math.Pow(2, bucketExponent);
        EmailAddress = emailAddress;
    }
    public Guid Id { get; set; }
    public string EmailAddress { get; set; } = "example@example.com";
    public int BucketNo { get; set; } = 0;
    public int EntryNo { get; set; } = 0;
}

public class KeyValues
{
    public string Key { get; set; }
    public int value { get; set; }
}

// Conversions
internal static class TupleExtensions
{
    public static string ToBase64String(this (string s, byte[] b) input)
    {
        return Convert.ToBase64String(input.b) + " " + input.s;
    }
}
internal static class ArrayExtensions
{
    public static string AesDecrypt(this string[] arr)
    {
        return AesEncryptionHelper.AesDecrypt(arr[1], Key.key, Convert
            .FromBase64String(arr[0]));
    }
}
public class PersonalDataConverter : ValueConverter<string, string>
{
    private const StringSplitOptions Sso =
        StringSplitOptions.TrimEntries | StringSplitOptions.RemoveEmptyEntries;
    public PersonalDataConverter() : base(
        cleartext => (AesEncryptionHelper.AesEncrypt(cleartext, Key.key, null))
            .ToBase64String(),
        ciphertext => ciphertext
            .Split(" ", Sso)
            .AesDecrypt()
      , default
    )
    { }
}

// non cryptographic hash functions
public static class NonCryptographicHelpers
{
    public static int GetBucketExponent(int userCount)
    {
        if (userCount < 1) { return 1; }
        return Math.Max((int)Math.Floor(Math.Log2(userCount)) - 3, 1);
    }
}

// Encryption
public static class Key
{
    public static byte[] key = {
        (byte) 33,
        (byte) 40,
        (byte) 212,
        (byte) 209,
        (byte) 219,
        (byte) 205,
        (byte) 88,
        (byte) 100,
        (byte) 20,
        (byte) 23,
        (byte) 131,
        (byte) 149,
        (byte) 104,
        (byte) 200,
        (byte) 215,
        (byte) 17,
        (byte) 36,
        (byte) 102,
        (byte) 106,
        (byte) 19,
        (byte) 165,
        (byte) 234,
        (byte) 163,
        (byte) 139,
        (byte) 133,
        (byte) 63,
        (byte) 139,
        (byte) 249,
        (byte) 224,
        (byte) 41,
        (byte) 186,
        (byte) 209,
    };
}


public static class AesEncryptionHelper
{
    /// <summary>
    /// Encrypt the given secret using AES
    /// </summary>
    /// <param name="secret">plaintext to encrypt</param>
    /// <param name="key">The secret key to use to encrypt</param>
    /// <param name="IV">Optional initialization vector to use</param>
    /// <returns>A tuple containing the base64 encoded, encrypted ciphertext, and the initialization ve
    /// <exception cref="ArgumentException">Key or IV is incorrect length</exception>
    public static (string ciphertext, byte[] IV) AesEncrypt(string secret, in byte[] key, in byte[]? IV
    {
        using (var aes = Aes.Create())
        {
            if (key.Length != aes.Key.Length)
            {
                throw new ArgumentException("key length incorrect");
            }
            if (IV != null && IV.Length != aes.IV.Length)
            {
                throw new ArgumentException("IV length incorrect");
            }
            aes.Key = key;
            if (IV != null)
            {
                aes.IV = IV;
            }
            var encryptor = aes.CreateEncryptor(aes.Key, aes.IV);
            using (MemoryStream msEncrypt = new MemoryStream())
            {
                using (CryptoStream csEncrypt = new CryptoStream(msEncrypt, encryptor, CryptoStreamMode
                {
                    using (StreamWriter swEncrypt = new StreamWriter(csEncrypt))
                    {
                        swEncrypt.Write(secret);
                    }
                    return (Convert.ToBase64String(msEncrypt.ToArray()), aes.IV);
                }
            }
        }
    }
    /// <summary>
    /// Decrypt the result of <see cref="AesEncrypt"/>
    /// </summary>
    /// <param name="secret">The base64 encoded, aes encrypted ciphertext</param>
    /// <param name="key">The secret key used to encrypt the secret</param>
    /// <param name="IV">The initialization vector used to encrypt the secret</param>
    /// <returns>The decrypted plaintext</returns>
    /// <exception cref="CryptographicException">Key or IV is incorrect length</exception>
    /// <exception cref="ArgumentNullException"/>
    /// <exception cref="FormatException">Secret is not a valid base 64 string </exception>
    public static string AesDecrypt(string secret, in byte[] key, in byte[] IV)
    {
        using (Aes aesAlg = Aes.Create())
        {
            aesAlg.Key = key;
            aesAlg.IV = IV;

            // Create a decryptor to perform the stream transform.
            ICryptoTransform decryptor = aesAlg.CreateDecryptor(aesAlg.Key, aesAlg.IV);

            // Create the streams used for decryption.
            using (MemoryStream msDecrypt = new MemoryStream(Convert.FromBase64String(secret)))
            {
                using (CryptoStream csDecrypt = new CryptoStream(msDecrypt, decryptor, CryptoStreamMode
                {
                    using (StreamReader srDecrypt = new StreamReader(csDecrypt))
                    {

                        // Read the decrypted bytes from the decrypting stream
                        // and place them in a string.
```

```
                        return srDecrypt.ReadToEnd();
                    }
                }
            }
        }
    }
}
```

Appendix: Raw data

The following are raw data. This can also be found on the provided pastebin links.

C# output

This was the result of allowing Generation to run for 3 days. It can also be found (here)[https://pastebin.com/z5X5Tv61]

```
Control: Using no encryption
Seeding DB with 1000 users
Saving changes
Adding 10 users individually
Iteration 0 took 125915 ticks
Iteration 1 took 109700 ticks
Iteration 2 took 87624 ticks
Iteration 3 took 96303 ticks
Iteration 4 took 110093 ticks
Iteration 5 took 97157 ticks
Iteration 6 took 102967 ticks
Iteration 7 took 105412 ticks
Iteration 8 took 139721 ticks
Iteration 9 took 89647 ticks
Changing 10 users email addresses, twice each
Iteration 0 on user 0 took 192229 ticks
Iteration 0 on user 1 took 121345 ticks
Iteration 0 on user 2 took 104096 ticks
Iteration 0 on user 3 took 88876 ticks
Iteration 0 on user 4 took 98232 ticks
Iteration 0 on user 5 took 100892 ticks
Iteration 0 on user 6 took 120133 ticks
Iteration 0 on user 7 took 90541 ticks
Iteration 0 on user 8 took 100625 ticks
Iteration 0 on user 9 took 86574 ticks
Iteration 1 on user 0 took 115270 ticks
Iteration 1 on user 1 took 87651 ticks
Iteration 1 on user 2 took 97321 ticks
Iteration 1 on user 3 took 98133 ticks
Iteration 1 on user 4 took 97710 ticks
Iteration 1 on user 5 took 91371 ticks
Iteration 1 on user 6 took 108267 ticks
Iteration 1 on user 7 took 87039 ticks
Iteration 1 on user 8 took 108001 ticks
Iteration 1 on user 9 took 111011 ticks
Control: using a fixed number of buckets
Seeding DB with 1000 users
Saving changes
```

```
Adding 10 users individually
Iteration 0 took 387184 ticks
Iteration 1 took 123588 ticks
Iteration 2 took 145833 ticks
Iteration 3 took 125489 ticks
Iteration 4 took 107759 ticks
Iteration 5 took 97789 ticks
Iteration 6 took 116862 ticks
Iteration 7 took 111355 ticks
Iteration 8 took 155429 ticks
Iteration 9 took 106439 ticks
Changing 10 users email addresses, twice each
Iteration 0 on user 0 took 140736 ticks
Iteration 0 on user 1 took 146311 ticks
Iteration 0 on user 2 took 141659 ticks
Iteration 0 on user 3 took 174228 ticks
Iteration 0 on user 4 took 139698 ticks
Iteration 0 on user 5 took 112702 ticks
Iteration 0 on user 6 took 128414 ticks
Iteration 0 on user 7 took 107844 ticks
Iteration 0 on user 8 took 106583 ticks
Iteration 0 on user 9 took 106817 ticks
Iteration 1 on user 0 took 100089 ticks
Iteration 1 on user 1 took 97202 ticks
Iteration 1 on user 2 took 96354 ticks
Iteration 1 on user 3 took 102101 ticks
Iteration 1 on user 4 took 110859 ticks
Iteration 1 on user 5 took 121846 ticks
Iteration 1 on user 6 took 124438 ticks
Iteration 1 on user 7 took 108929 ticks
Iteration 1 on user 8 took 105626 ticks
Iteration 1 on user 9 took 94361 ticks
Using ‘SELECT COUNT(*)‘ for updates to BucketCount, storing BucketExponent in the entity
Seeding DB with 1000 users
Saving changes
Adding 10 users individually
Iteration 0 took 400691 ticks
Iteration 1 took 181629 ticks
Iteration 2 took 178578 ticks
Iteration 3 took 325571 ticks
Iteration 4 took 277091 ticks
Iteration 5 took 213632 ticks
Iteration 6 took 219604 ticks
Iteration 7 took 230411 ticks
Iteration 8 took 232851 ticks
Iteration 9 took 249286 ticks
Changing 10 users email addresses, twice each
Iteration 0 on user 0 took 270336 ticks
Iteration 0 on user 1 took 249826 ticks
Iteration 0 on user 2 took 277684 ticks
Iteration 0 on user 3 took 258408 ticks
Iteration 0 on user 4 took 260940 ticks
Iteration 0 on user 5 took 270210 ticks
Iteration 0 on user 6 took 260713 ticks
Iteration 0 on user 7 took 262322 ticks
Iteration 0 on user 8 took 263575 ticks
Iteration 0 on user 9 took 281274 ticks
Iteration 1 on user 0 took 269038 ticks
Iteration 1 on user 1 took 265077 ticks
Iteration 1 on user 2 took 270030 ticks
Iteration 1 on user 3 took 247109 ticks
Iteration 1 on user 4 took 242536 ticks
Iteration 1 on user 5 took 252842 ticks
Iteration 1 on user 6 took 249719 ticks
Iteration 1 on user 7 took 276344 ticks
Iteration 1 on user 8 took 273388 ticks
Iteration 1 on user 9 took 269117 ticks
Using a summary table for updates to BucketCount, storing BucketExponent in the entity
Seeding DB with 1000 users
Saving changes
Adding 10 users individually
Iteration 0 took 266057 ticks
Iteration 1 took 208420 ticks
Iteration 2 took 178676 ticks
Iteration 3 took 193731 ticks
Iteration 4 took 208675 ticks
Iteration 5 took 213283 ticks
Iteration 6 took 258683 ticks
Iteration 7 took 259354 ticks
Iteration 8 took 244300 ticks
Iteration 9 took 238287 ticks
Changing 10 users email addresses, twice each
Iteration 0 on user 0 took 267937 ticks
Iteration 0 on user 1 took 277018 ticks
Iteration 0 on user 2 took 292599 ticks
Iteration 0 on user 3 took 279157 ticks
Iteration 0 on user 4 took 251162 ticks
Iteration 0 on user 5 took 250267 ticks
Iteration 0 on user 6 took 247666 ticks
Iteration 0 on user 7 took 253423 ticks
Iteration 0 on user 8 took 253339 ticks
Iteration 0 on user 9 took 269060 ticks
Iteration 1 on user 0 took 266536 ticks
Iteration 1 on user 1 took 294245 ticks
Iteration 1 on user 2 took 258685 ticks
Iteration 1 on user 3 took 258493 ticks
Iteration 1 on user 4 took 251259 ticks
Iteration 1 on user 5 took 248941 ticks
Iteration 1 on user 6 took 246978 ticks
Iteration 1 on user 7 took 249891 ticks
Iteration 1 on user 8 took 272175 ticks
Iteration 1 on user 9 took 255044 ticks
Using autoincremented row numbers
Seeding DB with 1000 users
Saving changes
Adding 10 users individually
Iteration 0 took 223925 ticks
Iteration 1 took 170681 ticks
Iteration 2 took 170198 ticks
Iteration 3 took 203296 ticks
Iteration 4 took 220807 ticks
Iteration 5 took 216432 ticks
Iteration 6 took 241818 ticks
Iteration 7 took 232538 ticks
Iteration 8 took 224783 ticks
Iteration 9 took 246420 ticks
Changing 10 users email addresses, twice each
Iteration 0 on user 0 took 252342 ticks
Iteration 0 on user 1 took 261797 ticks
Iteration 0 on user 2 took 289953 ticks
Iteration 0 on user 3 took 270112 ticks
Iteration 0 on user 4 took 268567 ticks
Iteration 0 on user 5 took 273613 ticks
Iteration 0 on user 6 took 264937 ticks
Iteration 0 on user 7 took 261278 ticks
Iteration 0 on user 8 took 246931 ticks
Iteration 0 on user 9 took 230709 ticks
Iteration 1 on user 0 took 238089 ticks
Iteration 1 on user 1 took 252896 ticks
Iteration 1 on user 2 took 259011 ticks
Iteration 1 on user 3 took 270015 ticks
Iteration 1 on user 4 took 267015 ticks
Iteration 1 on user 5 took 263204 ticks
Iteration 1 on user 6 took 271006 ticks
Iteration 1 on user 7 took 252629 ticks
Iteration 1 on user 8 took 252524 ticks
Iteration 1 on user 9 took 259072 ticks
Using ‘SELECT COUNT(*)‘ for updates to BucketCount, updating all bucket numbers after threshold is pass
Seeding DB with 1000 users
Saving changes
Adding 10 users individually
Iteration 0 took 67273 ticks
Iteration 1 took 34529 ticks
Iteration 2 took 45961 ticks
Iteration 3 took 38198 ticks
Iteration 4 took 34042 ticks
Iteration 5 took 32439 ticks
Iteration 6 took 31803 ticks
Iteration 7 took 39787 ticks
Iteration 8 took 35189 ticks
Iteration 9 took 33348 ticks
Changing 10 users email addresses, twice each
Iteration 0 on user 0 took 134930 ticks
Iteration 0 on user 1 took 134884 ticks
Iteration 0 on user 2 took 137315 ticks
Iteration 0 on user 3 took 127422 ticks
Iteration 0 on user 4 took 123143 ticks
Iteration 0 on user 5 took 138347 ticks
Iteration 0 on user 6 took 146856 ticks
Iteration 0 on user 7 took 113937 ticks
Iteration 0 on user 8 took 160258 ticks
Iteration 0 on user 9 took 141649 ticks
Iteration 1 on user 0 took 140440 ticks
Iteration 1 on user 1 took 129264 ticks
Iteration 1 on user 2 took 144075 ticks
Iteration 1 on user 3 took 132971 ticks
Iteration 1 on user 4 took 121115 ticks
Iteration 1 on user 5 took 112827 ticks
Iteration 1 on user 6 took 109957 ticks
Iteration 1 on user 7 took 115862 ticks
Iteration 1 on user 8 took 123194 ticks
Iteration 1 on user 9 took 124116 ticks
Control: Using no encryption
Seeding DB with 1000 users
Saving changes
Adding 20 users individually
Iteration 0 took 102048 ticks
Iteration 1 took 79765 ticks
Iteration 2 took 90511 ticks
Iteration 3 took 96056 ticks
Iteration 4 took 92279 ticks
Iteration 5 took 84078 ticks
Iteration 6 took 134905 ticks
Iteration 7 took 142946 ticks
Iteration 8 took 87196 ticks
Iteration 9 took 86810 ticks
Iteration 10 took 77642 ticks
Iteration 11 took 81453 ticks
Iteration 12 took 68965 ticks
Iteration 13 took 77494 ticks
Iteration 14 took 78211 ticks
Iteration 15 took 90474 ticks
Iteration 16 took 96168 ticks
Iteration 17 took 69546 ticks
Iteration 18 took 77884 ticks
Iteration 19 took 66429 ticks
Changing 20 users email addresses, twice each
Iteration 0 on user 0 took 81919 ticks
Iteration 0 on user 1 took 59407 ticks
Iteration 0 on user 2 took 64863 ticks
Iteration 0 on user 3 took 67176 ticks
Iteration 0 on user 4 took 64392 ticks
Iteration 0 on user 5 took 69668 ticks
Iteration 0 on user 6 took 62392 ticks
Iteration 0 on user 7 took 58927 ticks
Iteration 0 on user 8 took 81152 ticks
Iteration 0 on user 9 took 73265 ticks
Iteration 0 on user 10 took 85632 ticks
Iteration 0 on user 11 took 109036 ticks
Iteration 0 on user 12 took 98112 ticks
Iteration 0 on user 13 took 85337 ticks
Iteration 0 on user 14 took 77434 ticks
Iteration 0 on user 15 took 84450 ticks
Iteration 0 on user 16 took 85414 ticks
Iteration 0 on user 17 took 77449 ticks
Iteration 0 on user 18 took 96957 ticks
Iteration 0 on user 19 took 90788 ticks
Iteration 1 on user 0 took 93046 ticks
Iteration 1 on user 1 took 82428 ticks
Iteration 1 on user 2 took 74593 ticks
Iteration 1 on user 3 took 86866 ticks
Iteration 1 on user 4 took 66720 ticks
Iteration 1 on user 5 took 74017 ticks
Iteration 1 on user 6 took 90969 ticks
Iteration 1 on user 7 took 83512 ticks
Iteration 1 on user 8 took 74535 ticks
Iteration 1 on user 9 took 88217 ticks
Iteration 1 on user 10 took 94600 ticks
Iteration 1 on user 11 took 92074 ticks
Iteration 1 on user 12 took 102055 ticks
Iteration 1 on user 13 took 94006 ticks
Iteration 1 on user 14 took 102528 ticks
Iteration 1 on user 15 took 98799 ticks
Iteration 1 on user 16 took 98918 ticks
Iteration 1 on user 17 took 92451 ticks
Iteration 1 on user 18 took 81218 ticks
Iteration 1 on user 19 took 63826 ticks
Control: using a fixed number of buckets
Seeding DB with 1000 users
Saving changes
Adding 20 users individually
Iteration 0 took 98299 ticks
Iteration 1 took 85442 ticks
Iteration 2 took 88366 ticks
Iteration 3 took 94688 ticks
Iteration 4 took 97916 ticks
Iteration 5 took 94108 ticks
Iteration 6 took 102121 ticks
Iteration 7 took 120411 ticks
Iteration 8 took 87906 ticks
Iteration 9 took 84495 ticks
Iteration 10 took 97974 ticks
Iteration 11 took 91976 ticks
Iteration 12 took 100875 ticks
Iteration 13 took 109937 ticks
Iteration 14 took 107985 ticks
Iteration 15 took 107730 ticks
Iteration 16 took 105837 ticks
Iteration 17 took 89417 ticks
Iteration 18 took 83785 ticks
Iteration 19 took 110233 ticks
Changing 20 users email addresses, twice each
Iteration 0 on user 0 took 110263 ticks
Iteration 0 on user 1 took 100791 ticks
Iteration 0 on user 2 took 95227 ticks
Iteration 0 on user 3 took 91681 ticks
Iteration 0 on user 4 took 101439 ticks
Iteration 0 on user 5 took 97214 ticks
Iteration 0 on user 6 took 222532 ticks
Iteration 0 on user 7 took 110712 ticks
Iteration 0 on user 8 took 112253 ticks
Iteration 0 on user 9 took 94362 ticks
Iteration 0 on user 10 took 95015 ticks
Iteration 0 on user 11 took 85965 ticks
Iteration 0 on user 12 took 99243 ticks
Iteration 0 on user 13 took 84526 ticks
Iteration 0 on user 14 took 110316 ticks
Iteration 0 on user 15 took 93306 ticks
Iteration 0 on user 16 took 87633 ticks
Iteration 0 on user 17 took 99122 ticks
Iteration 0 on user 18 took 98055 ticks
Iteration 0 on user 19 took 93200 ticks
Iteration 1 on user 0 took 91376 ticks
Iteration 1 on user 1 took 130653 ticks
Iteration 1 on user 2 took 105125 ticks
Iteration 1 on user 3 took 122158 ticks
Iteration 1 on user 4 took 130953 ticks
Iteration 1 on user 5 took 107435 ticks
Iteration 1 on user 6 took 110699 ticks
Iteration 1 on user 7 took 114817 ticks
Iteration 1 on user 8 took 110464 ticks
Iteration 1 on user 9 took 121475 ticks
Iteration 1 on user 10 took 91102 ticks
Iteration 1 on user 11 took 110287 ticks
Iteration 1 on user 12 took 99555 ticks
Iteration 1 on user 13 took 111306 ticks
Iteration 1 on user 14 took 111283 ticks
Iteration 1 on user 15 took 102663 ticks
Iteration 1 on user 16 took 95852 ticks
Iteration 1 on user 17 took 103838 ticks
Iteration 1 on user 18 took 130393 ticks
Iteration 1 on user 19 took 123397 ticks
Using ‘SELECT COUNT(*)‘ for updates to BucketCount, storing BucketExponent in the entity
Seeding DB with 1000 users
Saving changes
Adding 20 users individually
Iteration 0 took 231521 ticks
Iteration 1 took 216606 ticks
Iteration 2 took 215943 ticks
Iteration 3 took 185591 ticks
Iteration 4 took 191287 ticks
Iteration 5 took 197354 ticks
Iteration 6 took 207217 ticks
Iteration 7 took 215933 ticks
Iteration 8 took 217363 ticks
Iteration 9 took 236370 ticks
Iteration 10 took 246014 ticks
Iteration 11 took 253276 ticks
Iteration 12 took 267834 ticks
Iteration 13 took 279297 ticks
Iteration 14 took 291885 ticks
Iteration 15 took 286623 ticks
Iteration 16 took 323141 ticks
Iteration 17 took 309019 ticks
Iteration 18 took 310627 ticks
Iteration 19 took 308702 ticks
Changing 20 users email addresses, twice each
Iteration 0 on user 0 took 328794 ticks
Iteration 0 on user 1 took 378254 ticks
Iteration 0 on user 2 took 381903 ticks
Iteration 0 on user 3 took 374731 ticks
Iteration 0 on user 4 took 343219 ticks
Iteration 0 on user 5 took 370549 ticks
Iteration 0 on user 6 took 338194 ticks
Iteration 0 on user 7 took 327484 ticks
Iteration 0 on user 8 took 343987 ticks
Iteration 0 on user 9 took 367552 ticks
Iteration 0 on user 10 took 350309 ticks
Iteration 0 on user 11 took 372116 ticks
Iteration 0 on user 12 took 349846 ticks
Iteration 0 on user 13 took 355176 ticks
Iteration 0 on user 14 took 360848 ticks
Iteration 0 on user 15 took 320541 ticks
Iteration 0 on user 16 took 327422 ticks
Iteration 0 on user 17 took 349412 ticks
Iteration 0 on user 18 took 363672 ticks
Iteration 0 on user 19 took 381508 ticks
Iteration 1 on user 0 took 396138 ticks
Iteration 1 on user 1 took 347374 ticks
Iteration 1 on user 2 took 357834 ticks
Iteration 1 on user 3 took 325260 ticks
Iteration 1 on user 4 took 391736 ticks
Iteration 1 on user 5 took 373131 ticks
Iteration 1 on user 6 took 366369 ticks
Iteration 1 on user 7 took 344527 ticks
Iteration 1 on user 8 took 358639 ticks
Iteration 1 on user 9 took 356688 ticks
Iteration 1 on user 10 took 341857 ticks
Iteration 1 on user 11 took 345675 ticks
Iteration 1 on user 12 took 337638 ticks
Iteration 1 on user 13 took 315677 ticks
Iteration 1 on user 14 took 343424 ticks
Iteration 1 on user 15 took 360157 ticks
Iteration 1 on user 16 took 360297 ticks
Iteration 1 on user 17 took 344439 ticks
Iteration 1 on user 18 took 364705 ticks
Iteration 1 on user 19 took 366212 ticks
Using a summary table for updates to BucketCount, storing BucketExponent in the entity
Seeding DB with 1000 users
Saving changes
Adding 20 users individually
Iteration 0 took 197575 ticks
Iteration 1 took 219961 ticks
Iteration 2 took 207896 ticks
Iteration 3 took 213221 ticks
Iteration 4 took 231721 ticks
Iteration 5 took 223388 ticks
Iteration 6 took 238625 ticks
Iteration 7 took 246049 ticks
Iteration 8 took 252885 ticks
Iteration 9 took 265719 ticks
Iteration 10 took 257550 ticks
Iteration 11 took 252336 ticks
Iteration 12 took 269592 ticks
Iteration 13 took 284268 ticks
Iteration 14 took 303186 ticks
Iteration 15 took 320063 ticks
Iteration 16 took 327020 ticks
Iteration 17 took 330970 ticks
Iteration 18 took 352937 ticks
Iteration 19 took 337840 ticks
Changing 20 users email addresses, twice each
Iteration 0 on user 0 took 313005 ticks
Iteration 0 on user 1 took 351944 ticks
Iteration 0 on user 2 took 338690 ticks
Iteration 0 on user 3 took 342038 ticks
Iteration 0 on user 4 took 354094 ticks
Iteration 0 on user 5 took 361101 ticks
Iteration 0 on user 6 took 352632 ticks
Iteration 0 on user 7 took 345737 ticks
Iteration 0 on user 8 took 347548 ticks
Iteration 0 on user 9 took 319723 ticks
Iteration 0 on user 10 took 329992 ticks
Iteration 0 on user 11 took 336822 ticks
Iteration 0 on user 12 took 343282 ticks
Iteration 0 on user 13 took 347227 ticks
Iteration 0 on user 14 took 350066 ticks
Iteration 0 on user 15 took 372130 ticks
Iteration 0 on user 16 took 357067 ticks
Iteration 0 on user 17 took 321230 ticks
Iteration 0 on user 18 took 319006 ticks
Iteration 0 on user 19 took 338406 ticks
Iteration 1 on user 0 took 339486 ticks
Iteration 1 on user 1 took 344906 ticks
Iteration 1 on user 2 took 337174 ticks
Iteration 1 on user 3 took 337069 ticks
Iteration 1 on user 4 took 349268 ticks
Iteration 1 on user 5 took 323144 ticks
Iteration 1 on user 6 took 325960 ticks
Iteration 1 on user 7 took 327756 ticks
Iteration 1 on user 8 took 332195 ticks
Iteration 1 on user 9 took 339165 ticks
Iteration 1 on user 10 took 436477 ticks
Iteration 1 on user 11 took 352482 ticks
Iteration 1 on user 12 took 353570 ticks
Iteration 1 on user 13 took 332876 ticks
Iteration 1 on user 14 took 314325 ticks
Iteration 1 on user 15 took 349220 ticks
Iteration 1 on user 16 took 345906 ticks
Iteration 1 on user 17 took 348043 ticks
Iteration 1 on user 18 took 348904 ticks
Iteration 1 on user 19 took 331638 ticks
```

```
Using autoincremented row numbers
Seeding DB with 1000 users
Saving changes
Adding 20 users individually
Iteration 0 took 165229 ticks
Iteration 1 took 154126 ticks
Iteration 2 took 167335 ticks
Iteration 3 took 195688 ticks
Iteration 4 took 212095 ticks
Iteration 5 took 211416 ticks
Iteration 6 took 218852 ticks
Iteration 7 took 220734 ticks
Iteration 8 took 229382 ticks
Iteration 9 took 243858 ticks
Iteration 10 took 254888 ticks
Iteration 11 took 274532 ticks
Iteration 12 took 278978 ticks
Iteration 13 took 276047 ticks
Iteration 14 took 272249 ticks
Iteration 15 took 289927 ticks
Iteration 16 took 312141 ticks
Iteration 17 took 298676 ticks
Iteration 18 took 303853 ticks
Iteration 19 took 317614 ticks
Changing 20 users email addresses, twice each
Iteration 0 on user 0 took 333143 ticks
Iteration 0 on user 1 took 375286 ticks
Iteration 0 on user 2 took 344971 ticks
Iteration 0 on user 3 took 330827 ticks
Iteration 0 on user 4 took 332933 ticks
Iteration 0 on user 5 took 347115 ticks
Iteration 0 on user 6 took 354462 ticks
Iteration 0 on user 7 took 353142 ticks
Iteration 0 on user 8 took 360634 ticks
Iteration 0 on user 9 took 352254 ticks
Iteration 0 on user 10 took 342486 ticks
Iteration 0 on user 11 took 325645 ticks
Iteration 0 on user 12 took 334837 ticks
Iteration 0 on user 13 took 372966 ticks
Iteration 0 on user 14 took 375800 ticks
Iteration 0 on user 15 took 370132 ticks
Iteration 0 on user 16 took 354513 ticks
Iteration 0 on user 17 took 331766 ticks
Iteration 0 on user 18 took 377899 ticks
Iteration 0 on user 19 took 346608 ticks
Iteration 1 on user 0 took 324214 ticks
Iteration 1 on user 1 took 334095 ticks
Iteration 1 on user 2 took 366119 ticks
Iteration 1 on user 3 took 357072 ticks
Iteration 1 on user 4 took 368873 ticks
Iteration 1 on user 5 took 465252 ticks
Iteration 1 on user 6 took 364157 ticks
Iteration 1 on user 7 took 352727 ticks
Iteration 1 on user 8 took 327712 ticks
Iteration 1 on user 9 took 316888 ticks
Iteration 1 on user 10 took 330295 ticks
Iteration 1 on user 11 took 353040 ticks
Iteration 1 on user 12 took 333435 ticks
Iteration 1 on user 13 took 378317 ticks
Iteration 1 on user 14 took 365022 ticks
Iteration 1 on user 15 took 369566 ticks
Iteration 1 on user 16 took 346503 ticks
Iteration 1 on user 17 took 320028 ticks
Iteration 1 on user 18 took 326708 ticks
Iteration 1 on user 19 took 322994 ticks
Using ‘SELECT COUNT(*)‘ for updates to BucketCount, updating all bucket numbers after threshold is passe
Seeding DB with 1000 users
Saving changes
Adding 20 users individually
Iteration 0 took 41042 ticks
Iteration 1 took 43823 ticks
Iteration 2 took 61234 ticks
Iteration 3 took 37340 ticks
Iteration 4 took 36427 ticks
Iteration 5 took 33058 ticks
Iteration 6 took 63264 ticks
Iteration 7 took 43109 ticks
Iteration 8 took 40487 ticks
Iteration 9 took 57112 ticks
Iteration 10 took 40283 ticks
Iteration 11 took 54857 ticks
Iteration 12 took 36397 ticks
Iteration 13 took 50099 ticks
Iteration 14 took 34550 ticks
Iteration 15 took 34053 ticks
Iteration 16 took 38604 ticks
Iteration 17 took 52671 ticks
Iteration 18 took 41540 ticks
Iteration 19 took 35862 ticks
Changing 20 users email addresses, twice each
Iteration 0 on user 0 took 119011 ticks
Iteration 0 on user 1 took 149837 ticks
Iteration 0 on user 2 took 118310 ticks
Iteration 0 on user 3 took 134781 ticks
Iteration 0 on user 4 took 136532 ticks
Iteration 0 on user 5 took 129900 ticks
Iteration 0 on user 6 took 133216 ticks
Iteration 0 on user 7 took 149717 ticks
Iteration 0 on user 8 took 151614 ticks
Iteration 0 on user 9 took 137248 ticks
Iteration 0 on user 10 took 118428 ticks
Iteration 0 on user 11 took 143567 ticks
Iteration 0 on user 12 took 122045 ticks
Iteration 0 on user 13 took 101092 ticks
Iteration 0 on user 14 took 104882 ticks
Iteration 0 on user 15 took 103384 ticks
Iteration 0 on user 16 took 105522 ticks
Iteration 0 on user 17 took 100325 ticks
Iteration 0 on user 18 took 102258 ticks
Iteration 0 on user 19 took 105211 ticks
Iteration 1 on user 0 took 102869 ticks
Iteration 1 on user 1 took 106381 ticks
Iteration 1 on user 2 took 94411 ticks
Iteration 1 on user 3 took 96952 ticks
Iteration 1 on user 4 took 145092 ticks
Iteration 1 on user 5 took 204043 ticks
Iteration 1 on user 6 took 134266 ticks
Iteration 1 on user 7 took 126916 ticks
Iteration 1 on user 8 took 135773 ticks
Iteration 1 on user 9 took 139431 ticks
Iteration 1 on user 10 took 126315 ticks
Iteration 1 on user 11 took 164945 ticks
Iteration 1 on user 12 took 133351 ticks
Iteration 1 on user 13 took 141000 ticks
Iteration 1 on user 14 took 135688 ticks
Iteration 1 on user 15 took 119556 ticks
Iteration 1 on user 16 took 128672 ticks
Iteration 1 on user 17 took 149521 ticks
Iteration 1 on user 18 took 126161 ticks
Iteration 1 on user 19 took 126590 ticks
Control: Using no encryption
Seeding DB with 1000 users
Saving changes
Adding 30 users individually
Iteration 0 took 78259 ticks
Iteration 1 took 79127 ticks
Iteration 2 took 70860 ticks
Iteration 3 took 78099 ticks
Iteration 4 took 71211 ticks
Iteration 5 took 77751 ticks
Iteration 6 took 82066 ticks
Iteration 7 took 66822 ticks
Iteration 8 took 104424 ticks
Iteration 9 took 97442 ticks
Iteration 10 took 104700 ticks
Iteration 11 took 91511 ticks
Iteration 12 took 88133 ticks
Iteration 13 took 80899 ticks
Iteration 14 took 77904 ticks
Iteration 15 took 73126 ticks
Iteration 16 took 83651 ticks
Iteration 17 took 78278 ticks
Iteration 18 took 69023 ticks
Iteration 19 took 76146 ticks
Iteration 20 took 89265 ticks
Iteration 21 took 95590 ticks
Iteration 22 took 96352 ticks
Iteration 23 took 89319 ticks
Iteration 24 took 87433 ticks
Iteration 25 took 87794 ticks
Iteration 26 took 70903 ticks
Iteration 27 took 79293 ticks
Iteration 28 took 90571 ticks
Iteration 29 took 80814 ticks
Changing 30 users email addresses, twice each
Iteration 0 on user 0 took 95246 ticks
Iteration 0 on user 1 took 99786 ticks
Iteration 0 on user 2 took 69172 ticks
Iteration 0 on user 3 took 70004 ticks
Iteration 0 on user 4 took 68036 ticks
Iteration 0 on user 5 took 66335 ticks
Iteration 0 on user 6 took 68857 ticks
Iteration 0 on user 7 took 63716 ticks
Iteration 0 on user 8 took 58453 ticks
Iteration 0 on user 9 took 59593 ticks
Iteration 0 on user 10 took 72820 ticks
Iteration 0 on user 11 took 59472 ticks
Iteration 0 on user 12 took 73930 ticks
Iteration 0 on user 13 took 78254 ticks
Iteration 0 on user 14 took 82674 ticks
Iteration 0 on user 15 took 65929 ticks
Iteration 0 on user 16 took 60931 ticks
Iteration 0 on user 17 took 81875 ticks
Iteration 0 on user 18 took 68828 ticks
Iteration 0 on user 19 took 68618 ticks
Iteration 0 on user 20 took 76766 ticks
Iteration 0 on user 21 took 91133 ticks
Iteration 0 on user 22 took 88368 ticks
Iteration 0 on user 23 took 103410 ticks
Iteration 0 on user 24 took 68646 ticks
Iteration 0 on user 25 took 81036 ticks
Iteration 0 on user 26 took 63766 ticks
Iteration 0 on user 27 took 76100 ticks
Iteration 0 on user 28 took 86116 ticks
Iteration 0 on user 29 took 71180 ticks
Iteration 1 on user 0 took 69559 ticks
Iteration 1 on user 1 took 89874 ticks
Iteration 1 on user 2 took 94716 ticks
Iteration 1 on user 3 took 83702 ticks
Iteration 1 on user 4 took 79320 ticks
Iteration 1 on user 5 took 67768 ticks
Iteration 1 on user 6 took 77542 ticks
Iteration 1 on user 7 took 74097 ticks
Iteration 1 on user 8 took 67042 ticks
Iteration 1 on user 9 took 99365 ticks
Iteration 1 on user 10 took 76782 ticks
Iteration 1 on user 11 took 94490 ticks
Iteration 1 on user 12 took 88649 ticks
Iteration 1 on user 13 took 105359 ticks
Iteration 1 on user 14 took 98151 ticks
Iteration 1 on user 15 took 100761 ticks
Iteration 1 on user 16 took 101475 ticks
Iteration 1 on user 17 took 103651 ticks
Iteration 1 on user 18 took 81661 ticks
Iteration 1 on user 19 took 71066 ticks
Iteration 1 on user 20 took 77389 ticks
Iteration 1 on user 21 took 76554 ticks
Iteration 1 on user 22 took 89098 ticks
Iteration 1 on user 23 took 87852 ticks
Iteration 1 on user 24 took 95662 ticks
Iteration 1 on user 25 took 72279 ticks
Iteration 1 on user 26 took 82398 ticks
Iteration 1 on user 27 took 83349 ticks
Iteration 1 on user 28 took 78744 ticks
Iteration 1 on user 29 took 91827 ticks
Control: using a fixed number of buckets
Seeding DB with 1000 users
Saving changes
Adding 30 users individually
Iteration 0 took 143568 ticks
Iteration 1 took 115438 ticks
Iteration 2 took 126894 ticks
Iteration 3 took 85813 ticks
Iteration 4 took 93387 ticks
Iteration 5 took 85128 ticks
Iteration 6 took 102187 ticks
Iteration 7 took 103198 ticks
Iteration 8 took 151743 ticks
Iteration 9 took 89008 ticks
Iteration 10 took 93293 ticks
Iteration 11 took 91899 ticks
Iteration 12 took 88943 ticks
Iteration 13 took 85849 ticks
Iteration 14 took 89991 ticks
Iteration 15 took 97049 ticks
Iteration 16 took 103007 ticks
Iteration 17 took 114622 ticks
Iteration 18 took 94736 ticks
Iteration 19 took 106159 ticks
Iteration 20 took 97498 ticks
Iteration 21 took 112095 ticks
Iteration 22 took 87211 ticks
Iteration 23 took 79493 ticks
Iteration 24 took 96239 ticks
Iteration 25 took 99243 ticks
Iteration 26 took 92516 ticks
Iteration 27 took 110818 ticks
Iteration 28 took 80981 ticks
Iteration 29 took 79616 ticks
Changing 30 users email addresses, twice each
Iteration 0 on user 0 took 93874 ticks
Iteration 0 on user 1 took 88890 ticks
Iteration 0 on user 2 took 82586 ticks
Iteration 0 on user 3 took 86321 ticks
Iteration 0 on user 4 took 98553 ticks
Iteration 0 on user 5 took 104282 ticks
Iteration 0 on user 6 took 101162 ticks
Iteration 0 on user 7 took 100766 ticks
Iteration 0 on user 8 took 107648 ticks
Iteration 0 on user 9 took 126410 ticks
Iteration 0 on user 10 took 99040 ticks
Iteration 0 on user 11 took 90167 ticks
Iteration 0 on user 12 took 101245 ticks
Iteration 0 on user 13 took 88123 ticks
Iteration 0 on user 14 took 102940 ticks
Iteration 0 on user 15 took 111412 ticks
Iteration 0 on user 16 took 102166 ticks
Iteration 0 on user 17 took 107210 ticks
Iteration 0 on user 18 took 114814 ticks
Iteration 0 on user 19 took 99656 ticks
Iteration 0 on user 20 took 108881 ticks
Iteration 0 on user 21 took 120324 ticks
Iteration 0 on user 22 took 110074 ticks
Iteration 0 on user 23 took 103493 ticks
Iteration 0 on user 24 took 147124 ticks
Iteration 0 on user 25 took 114095 ticks
Iteration 0 on user 26 took 122008 ticks
Iteration 0 on user 27 took 116967 ticks
Iteration 0 on user 28 took 94728 ticks
Iteration 0 on user 29 took 93988 ticks
Iteration 1 on user 0 took 107260 ticks
Iteration 1 on user 1 took 115093 ticks
Iteration 1 on user 2 took 106019 ticks
Iteration 1 on user 3 took 99001 ticks
Iteration 1 on user 4 took 107461 ticks
Iteration 1 on user 5 took 114260 ticks
Iteration 1 on user 6 took 105380 ticks
Iteration 1 on user 7 took 107340 ticks
Iteration 1 on user 8 took 127429 ticks
Iteration 1 on user 9 took 117159 ticks
Iteration 1 on user 10 took 112135 ticks
Iteration 1 on user 11 took 95151 ticks
Iteration 1 on user 12 took 100139 ticks
Iteration 1 on user 13 took 102777 ticks
Iteration 1 on user 14 took 104729 ticks
Iteration 1 on user 15 took 110059 ticks
Iteration 1 on user 16 took 106514 ticks
Iteration 1 on user 17 took 106371 ticks
Iteration 1 on user 18 took 99968 ticks
Iteration 1 on user 19 took 91756 ticks
Iteration 1 on user 20 took 152230 ticks
Iteration 1 on user 21 took 123801 ticks
Iteration 1 on user 22 took 120114 ticks
Iteration 1 on user 23 took 102089 ticks
Iteration 1 on user 24 took 105343 ticks
Iteration 1 on user 25 took 105955 ticks
Iteration 1 on user 26 took 98128 ticks
Iteration 1 on user 27 took 96454 ticks
Iteration 1 on user 28 took 100330 ticks
Iteration 1 on user 29 took 135328 ticks
Using ‘SELECT COUNT(*)‘ for updates to BucketCount, storing BucketExponent in the entity
Seeding DB with 1000 users
Saving changes
Adding 30 users individually
```

```
Iteration 0 took 186664 ticks
Iteration 1 took 183612 ticks
Iteration 2 took 173935 ticks
Iteration 3 took 186729 ticks
Iteration 4 took 195353 ticks
Iteration 5 took 199596 ticks
Iteration 6 took 207363 ticks
Iteration 7 took 220455 ticks
Iteration 8 took 224294 ticks
Iteration 9 took 236374 ticks
Iteration 10 took 238975 ticks
Iteration 11 took 263691 ticks
Iteration 12 took 253124 ticks
Iteration 13 took 246123 ticks
Iteration 14 took 261771 ticks
Iteration 15 took 278987 ticks
Iteration 16 took 299355 ticks
Iteration 17 took 311666 ticks
Iteration 18 took 313343 ticks
Iteration 19 took 309410 ticks
Iteration 20 took 332918 ticks
Iteration 21 took 340997 ticks
Iteration 22 took 441698 ticks
Iteration 23 took 355895 ticks
Iteration 24 took 366713 ticks
Iteration 25 took 385144 ticks
Iteration 26 took 379414 ticks
Iteration 27 took 392746 ticks
Iteration 28 took 432807 ticks
Iteration 29 took 406623 ticks
Changing 30 users email addresses, twice each
Iteration 0 on user 0 took 401932 ticks
Iteration 0 on user 1 took 442367 ticks
Iteration 0 on user 2 took 442020 ticks
Iteration 0 on user 3 took 422227 ticks
Iteration 0 on user 4 took 429425 ticks
Iteration 0 on user 5 took 427482 ticks
Iteration 0 on user 6 took 421885 ticks
Iteration 0 on user 7 took 421977 ticks
Iteration 0 on user 8 took 465670 ticks
Iteration 0 on user 9 took 446581 ticks
Iteration 0 on user 10 took 425814 ticks
Iteration 0 on user 11 took 430246 ticks
Iteration 0 on user 12 took 440289 ticks
Iteration 0 on user 13 took 413156 ticks
Iteration 0 on user 14 took 421418 ticks
Iteration 0 on user 15 took 424993 ticks
Iteration 0 on user 16 took 431172 ticks
Iteration 0 on user 17 took 437791 ticks
Iteration 0 on user 18 took 430546 ticks
Iteration 0 on user 19 took 406979 ticks
Iteration 0 on user 20 took 418955 ticks
Iteration 0 on user 21 took 461111 ticks
Iteration 0 on user 22 took 434035 ticks
Iteration 0 on user 23 took 418714 ticks
Iteration 0 on user 24 took 443424 ticks
Iteration 0 on user 25 took 419051 ticks
Iteration 0 on user 26 took 408992 ticks
Iteration 0 on user 27 took 417197 ticks
Iteration 0 on user 28 took 414389 ticks
Iteration 0 on user 29 took 423906 ticks
Iteration 1 on user 0 took 422652 ticks
Iteration 1 on user 1 took 421046 ticks
Iteration 1 on user 2 took 409871 ticks
Iteration 1 on user 3 took 421906 ticks
Iteration 1 on user 4 took 441410 ticks
Iteration 1 on user 5 took 435933 ticks
Iteration 1 on user 6 took 435410 ticks
Iteration 1 on user 7 took 427016 ticks
Iteration 1 on user 8 took 428091 ticks
Iteration 1 on user 9 took 421156 ticks
Iteration 1 on user 10 took 400914 ticks
Iteration 1 on user 11 took 420785 ticks
Iteration 1 on user 12 took 431259 ticks
Iteration 1 on user 13 took 416598 ticks
Iteration 1 on user 14 took 421315 ticks
Iteration 1 on user 15 took 431253 ticks
Iteration 1 on user 16 took 412191 ticks
Iteration 1 on user 17 took 492582 ticks
Iteration 1 on user 18 took 540019 ticks
Iteration 1 on user 19 took 526973 ticks
Iteration 1 on user 20 took 416374 ticks
Iteration 1 on user 21 took 407955 ticks
Iteration 1 on user 22 took 419032 ticks
Iteration 1 on user 23 took 388512 ticks
Iteration 1 on user 24 took 419927 ticks
Iteration 1 on user 25 took 461032 ticks
Iteration 1 on user 26 took 430591 ticks
Iteration 1 on user 27 took 432283 ticks
Iteration 1 on user 28 took 429345 ticks
Iteration 1 on user 29 took 432161 ticks
Using a summary table for updates to BucketCount, storing BucketExponent in the entity
Seeding DB with 1000 users
Saving changes
Adding 30 users individually
Iteration 0 took 208731 ticks
Iteration 1 took 278937 ticks
Iteration 2 took 213173 ticks
Iteration 3 took 215883 ticks
Iteration 4 took 223765 ticks
Iteration 5 took 209869 ticks
Iteration 6 took 231237 ticks
Iteration 7 took 245687 ticks
Iteration 8 took 255155 ticks
Iteration 9 took 278936 ticks
Iteration 10 took 266315 ticks
Iteration 11 took 253414 ticks
Iteration 12 took 268338 ticks
Iteration 13 took 300411 ticks
Iteration 14 took 290266 ticks
Iteration 15 took 299737 ticks
Iteration 16 took 330462 ticks
Iteration 17 took 314441 ticks
Iteration 18 took 330258 ticks
Iteration 19 took 344817 ticks
Iteration 20 took 311784 ticks
Iteration 21 took 345324 ticks
Iteration 22 took 420405 ticks
Iteration 23 took 406446 ticks
Iteration 24 took 384234 ticks
Iteration 25 took 409437 ticks
Iteration 26 took 398746 ticks
Iteration 27 took 387730 ticks
Iteration 28 took 404422 ticks
Iteration 29 took 432046 ticks
Changing 30 users email addresses, twice each
Iteration 0 on user 0 took 412496 ticks
Iteration 0 on user 1 took 451851 ticks
Iteration 0 on user 2 took 429601 ticks
Iteration 0 on user 3 took 483354 ticks
Iteration 0 on user 4 took 473471 ticks
Iteration 0 on user 5 took 452776 ticks
Iteration 0 on user 6 took 448631 ticks
Iteration 0 on user 7 took 436991 ticks
Iteration 0 on user 8 took 434827 ticks
Iteration 0 on user 9 took 436248 ticks
Iteration 0 on user 10 took 417280 ticks
Iteration 0 on user 11 took 412873 ticks
Iteration 0 on user 12 took 423932 ticks
Iteration 0 on user 13 took 446367 ticks
Iteration 0 on user 14 took 434720 ticks
Iteration 0 on user 15 took 446583 ticks
Iteration 0 on user 16 took 416538 ticks
Iteration 0 on user 17 took 415469 ticks
Iteration 0 on user 18 took 405063 ticks
Iteration 0 on user 19 took 424881 ticks
Iteration 0 on user 20 took 435268 ticks
Iteration 0 on user 21 took 428710 ticks
Iteration 0 on user 22 took 444704 ticks
Iteration 0 on user 23 took 417366 ticks
Iteration 0 on user 24 took 405767 ticks
Iteration 0 on user 25 took 408228 ticks
Iteration 0 on user 26 took 419054 ticks
Iteration 0 on user 27 took 431049 ticks
Iteration 0 on user 28 took 428661 ticks
Iteration 0 on user 29 took 422293 ticks
Iteration 1 on user 0 took 418881 ticks
Iteration 1 on user 1 took 409959 ticks
Iteration 1 on user 2 took 423896 ticks
Iteration 1 on user 3 took 427648 ticks
Iteration 1 on user 4 took 426802 ticks
Iteration 1 on user 5 took 463483 ticks
Iteration 1 on user 6 took 430061 ticks
Iteration 1 on user 7 took 418140 ticks
Iteration 1 on user 8 took 402102 ticks
Iteration 1 on user 9 took 466995 ticks
Iteration 1 on user 10 took 444137 ticks
Iteration 1 on user 11 took 432280 ticks
Iteration 1 on user 12 took 445688 ticks
Iteration 1 on user 13 took 442183 ticks
Iteration 1 on user 14 took 409889 ticks
Iteration 1 on user 15 took 400856 ticks
Iteration 1 on user 16 took 416527 ticks
Iteration 1 on user 17 took 430446 ticks
Iteration 1 on user 18 took 435789 ticks
Iteration 1 on user 19 took 448868 ticks
Iteration 1 on user 20 took 430446 ticks
Iteration 1 on user 21 took 412113 ticks
Iteration 1 on user 22 took 422797 ticks
Iteration 1 on user 23 took 437677 ticks
Iteration 1 on user 24 took 429075 ticks
Iteration 1 on user 25 took 420583 ticks
Iteration 1 on user 26 took 422936 ticks
Iteration 1 on user 27 took 384714 ticks
Iteration 1 on user 28 took 410179 ticks
Iteration 1 on user 29 took 419806 ticks
Using autoincremented row numbers
Seeding DB with 1000 users
Saving changes
Adding 30 users individually
Iteration 0 took 196914 ticks
Iteration 1 took 189802 ticks
Iteration 2 took 180146 ticks
Iteration 3 took 189136 ticks
Iteration 4 took 190684 ticks
Iteration 5 took 194919 ticks
Iteration 6 took 205358 ticks
Iteration 7 took 209400 ticks
Iteration 8 took 225133 ticks
Iteration 9 took 226544 ticks
Iteration 10 took 241753 ticks
Iteration 11 took 308494 ticks
Iteration 12 took 283496 ticks
Iteration 13 took 279717 ticks
Iteration 14 took 283852 ticks
Iteration 15 took 297119 ticks
Iteration 16 took 275543 ticks
Iteration 17 took 294473 ticks
Iteration 18 took 316377 ticks
Iteration 19 took 333386 ticks
Iteration 20 took 371606 ticks
Iteration 21 took 348953 ticks
Iteration 22 took 363823 ticks
Iteration 23 took 366546 ticks
Iteration 24 took 362604 ticks
Iteration 25 took 358754 ticks
Iteration 26 took 360026 ticks
Iteration 27 took 391579 ticks
Iteration 28 took 401483 ticks
Iteration 29 took 411406 ticks
Changing 30 users email addresses, twice each
Iteration 0 on user 0 took 438285 ticks
Iteration 0 on user 1 took 432216 ticks
Iteration 0 on user 2 took 410204 ticks
Iteration 0 on user 3 took 445447 ticks
Iteration 0 on user 4 took 415956 ticks
Iteration 0 on user 5 took 429096 ticks
Iteration 0 on user 6 took 426264 ticks
Iteration 0 on user 7 took 440496 ticks
Iteration 0 on user 8 took 503376 ticks
Iteration 0 on user 9 took 441266 ticks
Iteration 0 on user 10 took 441514 ticks
Iteration 0 on user 11 took 445771 ticks
Iteration 0 on user 12 took 464467 ticks
Iteration 0 on user 13 took 439159 ticks
Iteration 0 on user 14 took 417186 ticks
Iteration 0 on user 15 took 422979 ticks
Iteration 0 on user 16 took 456864 ticks
Iteration 0 on user 17 took 440929 ticks
Iteration 0 on user 18 took 432054 ticks
Iteration 0 on user 19 took 437693 ticks
Iteration 0 on user 20 took 439961 ticks
Iteration 0 on user 21 took 414000 ticks
Iteration 0 on user 22 took 428041 ticks
Iteration 0 on user 23 took 444117 ticks
Iteration 0 on user 24 took 429560 ticks
Iteration 0 on user 25 took 432381 ticks
Iteration 0 on user 26 took 432776 ticks
Iteration 0 on user 27 took 416631 ticks
Iteration 0 on user 28 took 416774 ticks
Iteration 0 on user 29 took 431683 ticks
Iteration 1 on user 0 took 438290 ticks
Iteration 1 on user 1 took 424193 ticks
Iteration 1 on user 2 took 444503 ticks
Iteration 1 on user 3 took 425846 ticks
Iteration 1 on user 4 took 402343 ticks
Iteration 1 on user 5 took 402077 ticks
Iteration 1 on user 6 took 434138 ticks
Iteration 1 on user 7 took 436989 ticks
Iteration 1 on user 8 took 422164 ticks
Iteration 1 on user 9 took 433291 ticks
Iteration 1 on user 10 took 416099 ticks
Iteration 1 on user 11 took 417641 ticks
Iteration 1 on user 12 took 520280 ticks
Iteration 1 on user 13 took 501496 ticks
Iteration 1 on user 14 took 436882 ticks
Iteration 1 on user 15 took 427186 ticks
Iteration 1 on user 16 took 451686 ticks
Iteration 1 on user 17 took 402341 ticks
Iteration 1 on user 18 took 412447 ticks
Iteration 1 on user 19 took 425428 ticks
Iteration 1 on user 20 took 415398 ticks
Iteration 1 on user 21 took 426800 ticks
Iteration 1 on user 22 took 434361 ticks
Iteration 1 on user 23 took 429154 ticks
Iteration 1 on user 24 took 419930 ticks
Iteration 1 on user 25 took 465952 ticks
Iteration 1 on user 26 took 430641 ticks
Iteration 1 on user 27 took 427852 ticks
Iteration 1 on user 28 took 427795 ticks
Iteration 1 on user 29 took 414332 ticks
Using ‘SELECT COUNT(*)‘ for updates to BucketCount, updating all bucket numbers after threshold is passe
Seeding DB with 1000 users
Saving changes
Adding 30 users individually
Iteration 0 took 41996 ticks
Iteration 1 took 49259 ticks
Iteration 2 took 40521 ticks
Iteration 3 took 42252 ticks
Iteration 4 took 40532 ticks
Iteration 5 took 37285 ticks
Iteration 6 took 36595 ticks
Iteration 7 took 33337 ticks
Iteration 8 took 30922 ticks
Iteration 9 took 31050 ticks
Iteration 10 took 29722 ticks
Iteration 11 took 31232 ticks
Iteration 12 took 31756 ticks
Iteration 13 took 31898 ticks
Iteration 14 took 33032 ticks
Iteration 15 took 29488 ticks
Iteration 16 took 30056 ticks
Iteration 17 took 29246 ticks
Iteration 18 took 29924 ticks
Iteration 19 took 35277 ticks
Iteration 20 took 29238 ticks
Iteration 21 took 30479 ticks
Iteration 22 took 29482 ticks
Iteration 23 took 33395 ticks
Iteration 24 took 30510 ticks
Iteration 25 took 30965 ticks
Iteration 26 took 30900 ticks
Iteration 27 took 30711 ticks
Iteration 28 took 57885 ticks
Iteration 29 took 35801 ticks
Changing 30 users email addresses, twice each
Iteration 0 on user 0 took 128909 ticks
Iteration 0 on user 1 took 113870 ticks
Iteration 0 on user 2 took 117703 ticks
Iteration 0 on user 3 took 110283 ticks
Iteration 0 on user 4 took 134024 ticks
Iteration 0 on user 5 took 165646 ticks
Iteration 0 on user 6 took 135106 ticks
Iteration 0 on user 7 took 132648 ticks
Iteration 0 on user 8 took 109748 ticks
Iteration 0 on user 9 took 112865 ticks
Iteration 0 on user 10 took 131618 ticks
Iteration 0 on user 11 took 135405 ticks
Iteration 0 on user 12 took 128105 ticks
Iteration 0 on user 13 took 154275 ticks
Iteration 0 on user 14 took 253430 ticks
Iteration 0 on user 15 took 139115 ticks
Iteration 0 on user 16 took 144909 ticks
Iteration 0 on user 17 took 119814 ticks
Iteration 0 on user 18 took 136716 ticks
Iteration 0 on user 19 took 137217 ticks
Iteration 0 on user 20 took 108546 ticks
Iteration 0 on user 21 took 103837 ticks
Iteration 0 on user 22 took 109135 ticks
Iteration 0 on user 23 took 102708 ticks
Iteration 0 on user 24 took 101884 ticks
Iteration 0 on user 25 took 101580 ticks
Iteration 0 on user 26 took 97333 ticks
Iteration 0 on user 27 took 99237 ticks
Iteration 0 on user 28 took 103955 ticks
Iteration 0 on user 29 took 102092 ticks
Iteration 1 on user 0 took 91819 ticks
Iteration 1 on user 1 took 110756 ticks
Iteration 1 on user 2 took 132123 ticks
Iteration 1 on user 3 took 146206 ticks
Iteration 1 on user 4 took 133788 ticks
Iteration 1 on user 5 took 134593 ticks
Iteration 1 on user 6 took 113434 ticks
Iteration 1 on user 7 took 109649 ticks
Iteration 1 on user 8 took 119984 ticks
Iteration 1 on user 9 took 129949 ticks
Iteration 1 on user 10 took 133677 ticks
Iteration 1 on user 11 took 114955 ticks
Iteration 1 on user 12 took 115459 ticks
Iteration 1 on user 13 took 141039 ticks
Iteration 1 on user 14 took 140121 ticks
Iteration 1 on user 15 took 121168 ticks
Iteration 1 on user 16 took 112701 ticks
Iteration 1 on user 17 took 141400 ticks
Iteration 1 on user 18 took 142986 ticks
Iteration 1 on user 19 took 137486 ticks
Iteration 1 on user 20 took 136757 ticks
Iteration 1 on user 21 took 117705 ticks
Iteration 1 on user 22 took 110671 ticks
Iteration 1 on user 23 took 112655 ticks
Iteration 1 on user 24 took 135052 ticks
Iteration 1 on user 25 took 132111 ticks
Iteration 1 on user 26 took 123168 ticks
Iteration 1 on user 27 took 120480 ticks
Iteration 1 on user 28 took 122478 ticks
Iteration 1 on user 29 took 132145 ticks
Control: Using no encryption
Seeding DB with 10000 users
```

```
Saving changes
Adding 10 users individually
Iteration 0 took 278577 ticks
Iteration 1 took 206448 ticks
Iteration 2 took 174385 ticks
Iteration 3 took 165496 ticks
Iteration 4 took 194372 ticks
Iteration 5 took 207237 ticks
Iteration 6 took 199415 ticks
Iteration 7 took 152148 ticks
Iteration 8 took 165553 ticks
Iteration 9 took 160702 ticks
Changing 10 users email addresses, twice each
Iteration 0 on user 0 took 148854 ticks
Iteration 0 on user 1 took 165662 ticks
Iteration 0 on user 2 took 164327 ticks
Iteration 0 on user 3 took 158193 ticks
Iteration 0 on user 4 took 157813 ticks
Iteration 0 on user 5 took 167331 ticks
Iteration 0 on user 6 took 147906 ticks
Iteration 0 on user 7 took 154871 ticks
Iteration 0 on user 8 took 164997 ticks
Iteration 0 on user 9 took 160583 ticks
Iteration 1 on user 0 took 159680 ticks
Iteration 1 on user 1 took 159779 ticks
Iteration 1 on user 2 took 161324 ticks
Iteration 1 on user 3 took 149201 ticks
Iteration 1 on user 4 took 164890 ticks
Iteration 1 on user 5 took 152962 ticks
Iteration 1 on user 6 took 168492 ticks
Iteration 1 on user 7 took 162118 ticks
Iteration 1 on user 8 took 144675 ticks
Iteration 1 on user 9 took 162175 ticks
Control: using a fixed number of buckets
Seeding DB with 10000 users
Saving changes
Adding 10 users individually
Iteration 0 took 214900 ticks
Iteration 1 took 215640 ticks
Iteration 2 took 221624 ticks
Iteration 3 took 236195 ticks
Iteration 4 took 210730 ticks
Iteration 5 took 193414 ticks
Iteration 6 took 188687 ticks
Iteration 7 took 190822 ticks
Iteration 8 took 182144 ticks
Iteration 9 took 203957 ticks
Changing 10 users email addresses, twice each
Iteration 0 on user 0 took 280998 ticks
Iteration 0 on user 1 took 279542 ticks
Iteration 0 on user 2 took 301291 ticks
Iteration 0 on user 3 took 275507 ticks
Iteration 0 on user 4 took 276607 ticks
Iteration 0 on user 5 took 274028 ticks
Iteration 0 on user 6 took 279442 ticks
Iteration 0 on user 7 took 278471 ticks
Iteration 0 on user 8 took 289874 ticks
Iteration 0 on user 9 took 278356 ticks
Iteration 1 on user 0 took 275814 ticks
Iteration 1 on user 1 took 283821 ticks
Iteration 1 on user 2 took 279743 ticks
Iteration 1 on user 3 took 255191 ticks
Iteration 1 on user 4 took 274058 ticks
Iteration 1 on user 5 took 287424 ticks
Iteration 1 on user 6 took 280779 ticks
Iteration 1 on user 7 took 279666 ticks
Iteration 1 on user 8 took 285733 ticks
Iteration 1 on user 9 took 287346 ticks
Using ‘SELECT COUNT(*)‘ for updates to BucketCount, storing BucketExponent in the entity
Seeding DB with 10000 users
Saving changes
Adding 10 users individually
Iteration 0 took 357488 ticks
Iteration 1 took 363565 ticks
Iteration 2 took 356082 ticks
Iteration 3 took 485339 ticks
Iteration 4 took 358854 ticks
Iteration 5 took 350351 ticks
Iteration 6 took 345207 ticks
Iteration 7 took 384420 ticks
Iteration 8 took 408529 ticks
Iteration 9 took 428033 ticks
Changing 10 users email addresses, twice each
Iteration 0 on user 0 took 407962 ticks
Iteration 0 on user 1 took 419061 ticks
Iteration 0 on user 2 took 424923 ticks
Iteration 0 on user 3 took 406724 ticks
Iteration 0 on user 4 took 393219 ticks
Iteration 0 on user 5 took 406485 ticks
Iteration 0 on user 6 took 418453 ticks
Iteration 0 on user 7 took 432844 ticks
Iteration 0 on user 8 took 426887 ticks
Iteration 0 on user 9 took 400771 ticks
Iteration 1 on user 0 took 403852 ticks
Iteration 1 on user 1 took 415875 ticks
Iteration 1 on user 2 took 403588 ticks
Iteration 1 on user 3 took 414215 ticks
Iteration 1 on user 4 took 374003 ticks
Iteration 1 on user 5 took 421161 ticks
Iteration 1 on user 6 took 438043 ticks
Iteration 1 on user 7 took 421858 ticks
Iteration 1 on user 8 took 400825 ticks
Iteration 1 on user 9 took 415589 ticks
Using a summary table for updates to BucketCount, storing BucketExponent in the entity
Seeding DB with 10000 users
Saving changes
Adding 10 users individually
Iteration 0 took 369932 ticks
Iteration 1 took 337968 ticks
Iteration 2 took 362122 ticks
Iteration 3 took 366295 ticks
Iteration 4 took 378418 ticks
Iteration 5 took 386342 ticks
Iteration 6 took 397820 ticks
Iteration 7 took 398794 ticks
Iteration 8 took 386804 ticks
Iteration 9 took 377514 ticks
Changing 10 users email addresses, twice each
Iteration 0 on user 0 took 494071 ticks
Iteration 0 on user 1 took 424189 ticks
Iteration 0 on user 2 took 426278 ticks
Iteration 0 on user 3 took 425920 ticks
Iteration 0 on user 4 took 417424 ticks
Iteration 0 on user 5 took 400655 ticks
Iteration 0 on user 6 took 386055 ticks
Iteration 0 on user 7 took 418373 ticks
Iteration 0 on user 8 took 417980 ticks
Iteration 0 on user 9 took 421500 ticks
Iteration 1 on user 0 took 470320 ticks
Iteration 1 on user 1 took 456712 ticks
Iteration 1 on user 2 took 463996 ticks
Iteration 1 on user 3 took 463328 ticks
Iteration 1 on user 4 took 380717 ticks
Iteration 1 on user 5 took 385330 ticks
Iteration 1 on user 6 took 396687 ticks
Iteration 1 on user 7 took 487987 ticks
Iteration 1 on user 8 took 462061 ticks
Iteration 1 on user 9 took 445446 ticks
Using autoincremented row numbers
Seeding DB with 10000 users
Saving changes
Adding 10 users individually
Iteration 0 took 408083 ticks
Iteration 1 took 373667 ticks
Iteration 2 took 390096 ticks
Iteration 3 took 414586 ticks
Iteration 4 took 394803 ticks
Iteration 5 took 427722 ticks
Iteration 6 took 404801 ticks
Iteration 7 took 413769 ticks
Iteration 8 took 412439 ticks
Iteration 9 took 456981 ticks
Changing 10 users email addresses, twice each
Iteration 0 on user 0 took 458825 ticks
Iteration 0 on user 1 took 477154 ticks
Iteration 0 on user 2 took 453564 ticks
Iteration 0 on user 3 took 472245 ticks
Iteration 0 on user 4 took 427412 ticks
Iteration 0 on user 5 took 452856 ticks
Iteration 0 on user 6 took 456119 ticks
Iteration 0 on user 7 took 459961 ticks
Iteration 0 on user 8 took 467456 ticks
Iteration 0 on user 9 took 431530 ticks
Iteration 1 on user 0 took 473047 ticks
Iteration 1 on user 1 took 457991 ticks
Iteration 1 on user 2 took 453517 ticks
Iteration 1 on user 3 took 470939 ticks
Iteration 1 on user 4 took 468542 ticks
Iteration 1 on user 5 took 459105 ticks
Iteration 1 on user 6 took 462647 ticks
Iteration 1 on user 7 took 495940 ticks
Iteration 1 on user 8 took 451720 ticks
Iteration 1 on user 9 took 448106 ticks
Using ‘SELECT COUNT(*)‘ for updates to BucketCount, updating all bucket numbers after threshold is passe
Seeding DB with 10000 users
Saving changes
Adding 10 users individually
Iteration 0 took 145310 ticks
Iteration 1 took 145582 ticks
Iteration 2 took 142123 ticks
Iteration 3 took 141652 ticks
Iteration 4 took 141104 ticks
Iteration 5 took 140467 ticks
Iteration 6 took 143684 ticks
Iteration 7 took 154171 ticks
Iteration 8 took 140420 ticks
Iteration 9 took 144408 ticks
Changing 10 users email addresses, twice each
Iteration 0 on user 0 took 441233 ticks
Iteration 0 on user 1 took 437012 ticks
Iteration 0 on user 2 took 444482 ticks
Iteration 0 on user 3 took 471804 ticks
Iteration 0 on user 4 took 408957 ticks
Iteration 0 on user 5 took 404810 ticks
Iteration 0 on user 6 took 523844 ticks
Iteration 0 on user 7 took 482229 ticks
Iteration 0 on user 8 took 452097 ticks
Iteration 0 on user 9 took 403522 ticks
Iteration 1 on user 0 took 491827 ticks
Iteration 1 on user 1 took 408233 ticks
Iteration 1 on user 2 took 425133 ticks
Iteration 1 on user 3 took 408147 ticks
Iteration 1 on user 4 took 412856 ticks
Iteration 1 on user 5 took 434277 ticks
Iteration 1 on user 6 took 437725 ticks
Iteration 1 on user 7 took 402754 ticks
Iteration 1 on user 8 took 434313 ticks
Iteration 1 on user 9 took 433198 ticks
Control: Using no encryption
Seeding DB with 10000 users
Saving changes
Adding 20 users individually
Iteration 0 took 237033 ticks
Iteration 1 took 263619 ticks
Iteration 2 took 258860 ticks
Iteration 3 took 260814 ticks
Iteration 4 took 231505 ticks
Iteration 5 took 226066 ticks
Iteration 6 took 222814 ticks
Iteration 7 took 232209 ticks
Iteration 8 took 207031 ticks
Iteration 9 took 200325 ticks
Iteration 10 took 240534 ticks
Iteration 11 took 226503 ticks
Iteration 12 took 233712 ticks
Iteration 13 took 192072 ticks
Iteration 14 took 253704 ticks
Iteration 15 took 328646 ticks
Iteration 16 took 231475 ticks
Iteration 17 took 197795 ticks
Iteration 18 took 249403 ticks
Iteration 19 took 221309 ticks
Changing 20 users email addresses, twice each
Iteration 0 on user 0 took 258470 ticks
Iteration 0 on user 1 took 223335 ticks
Iteration 0 on user 2 took 194891 ticks
Iteration 0 on user 3 took 232882 ticks
Iteration 0 on user 4 took 222916 ticks
Iteration 0 on user 5 took 223893 ticks
Iteration 0 on user 6 took 226310 ticks
Iteration 0 on user 7 took 194111 ticks
Iteration 0 on user 8 took 249162 ticks
Iteration 0 on user 9 took 285829 ticks
Iteration 0 on user 10 took 241413 ticks
Iteration 0 on user 11 took 351213 ticks
Iteration 0 on user 12 took 236230 ticks
Iteration 0 on user 13 took 297999 ticks
Iteration 0 on user 14 took 296731 ticks
Iteration 0 on user 15 took 235519 ticks
Iteration 0 on user 16 took 235453 ticks
Iteration 0 on user 17 took 197062 ticks
Iteration 0 on user 18 took 221980 ticks
Iteration 0 on user 19 took 251509 ticks
Iteration 1 on user 0 took 222240 ticks
Iteration 1 on user 1 took 194764 ticks
Iteration 1 on user 2 took 250886 ticks
Iteration 1 on user 3 took 229879 ticks
Iteration 1 on user 4 took 229136 ticks
Iteration 1 on user 5 took 196127 ticks
Iteration 1 on user 6 took 254000 ticks
Iteration 1 on user 7 took 230447 ticks
Iteration 1 on user 8 took 245941 ticks
Iteration 1 on user 9 took 225757 ticks
Iteration 1 on user 10 took 232315 ticks
Iteration 1 on user 11 took 195137 ticks
Iteration 1 on user 12 took 226017 ticks
Iteration 1 on user 13 took 207375 ticks
Iteration 1 on user 14 took 265540 ticks
Iteration 1 on user 15 took 228145 ticks
Iteration 1 on user 16 took 198478 ticks
Iteration 1 on user 17 took 217704 ticks
Iteration 1 on user 18 took 259739 ticks
Iteration 1 on user 19 took 224837 ticks
Control: using a fixed number of buckets
Seeding DB with 10000 users
Saving changes
Adding 20 users individually
Iteration 0 took 295254 ticks
Iteration 1 took 275542 ticks
Iteration 2 took 262960 ticks
Iteration 3 took 258465 ticks
Iteration 4 took 346062 ticks
Iteration 5 took 355486 ticks
Iteration 6 took 313795 ticks
Iteration 7 took 363007 ticks
Iteration 8 took 289071 ticks
Iteration 9 took 279413 ticks
Iteration 10 took 306680 ticks
Iteration 11 took 264795 ticks
Iteration 12 took 281618 ticks
Iteration 13 took 255789 ticks
Iteration 14 took 289837 ticks
Iteration 15 took 262624 ticks
Iteration 16 took 264776 ticks
Iteration 17 took 266853 ticks
Iteration 18 took 264885 ticks
Iteration 19 took 264907 ticks
Changing 20 users email addresses, twice each
Iteration 0 on user 0 took 318679 ticks
Iteration 0 on user 1 took 353492 ticks
Iteration 0 on user 2 took 350640 ticks
Iteration 0 on user 3 took 315574 ticks
Iteration 0 on user 4 took 350320 ticks
Iteration 0 on user 5 took 350640 ticks
Iteration 0 on user 6 took 346893 ticks
Iteration 0 on user 7 took 344128 ticks
Iteration 0 on user 8 took 374252 ticks
Iteration 0 on user 9 took 347353 ticks
Iteration 0 on user 10 took 378369 ticks
Iteration 0 on user 11 took 358137 ticks
Iteration 0 on user 12 took 349344 ticks
Iteration 0 on user 13 took 354270 ticks
Iteration 0 on user 14 took 353123 ticks
Iteration 0 on user 15 took 408399 ticks
Iteration 0 on user 16 took 507859 ticks
Iteration 0 on user 17 took 408815 ticks
Iteration 0 on user 18 took 422105 ticks
Iteration 0 on user 19 took 449614 ticks
Iteration 1 on user 0 took 348890 ticks
Iteration 1 on user 1 took 371931 ticks
Iteration 1 on user 2 took 355724 ticks
Iteration 1 on user 3 took 351966 ticks
Iteration 1 on user 4 took 351573 ticks
Iteration 1 on user 5 took 346191 ticks
Iteration 1 on user 6 took 348492 ticks
Iteration 1 on user 7 took 404169 ticks
Iteration 1 on user 8 took 350440 ticks
Iteration 1 on user 9 took 359532 ticks
Iteration 1 on user 10 took 374109 ticks
Iteration 1 on user 11 took 347239 ticks
Iteration 1 on user 12 took 371676 ticks
Iteration 1 on user 13 took 355956 ticks
Iteration 1 on user 14 took 343382 ticks
Iteration 1 on user 15 took 356282 ticks
Iteration 1 on user 16 took 352365 ticks
Iteration 1 on user 17 took 354666 ticks
Iteration 1 on user 18 took 354607 ticks
Iteration 1 on user 19 took 385141 ticks
Using ‘SELECT COUNT(*)‘ for updates to BucketCount, storing BucketExponent in the entity
Seeding DB with 10000 users
Saving changes
Adding 20 users individually
Iteration 0 took 542845 ticks
Iteration 1 took 527918 ticks
Iteration 2 took 547021 ticks
Iteration 3 took 568702 ticks
Iteration 4 took 461475 ticks
Iteration 5 took 494166 ticks
Iteration 6 took 453530 ticks
Iteration 7 took 464435 ticks
Iteration 8 took 469846 ticks
Iteration 9 took 477145 ticks
Iteration 10 took 503809 ticks
Iteration 11 took 492303 ticks
Iteration 12 took 498441 ticks
Iteration 13 took 488512 ticks
Iteration 14 took 522156 ticks
Iteration 15 took 524545 ticks
Iteration 16 took 527373 ticks
Iteration 17 took 586988 ticks
Iteration 18 took 721345 ticks
Iteration 19 took 582529 ticks
Changing 20 users email addresses, twice each
Iteration 0 on user 0 took 641017 ticks
Iteration 0 on user 1 took 577548 ticks
Iteration 0 on user 2 took 578086 ticks
Iteration 0 on user 3 took 587349 ticks
Iteration 0 on user 4 took 573597 ticks
Iteration 0 on user 5 took 589196 ticks
Iteration 0 on user 6 took 555309 ticks
Iteration 0 on user 7 took 553621 ticks
Iteration 0 on user 8 took 573120 ticks
Iteration 0 on user 9 took 577837 ticks
Iteration 0 on user 10 took 582806 ticks
Iteration 0 on user 11 took 602749 ticks
Iteration 0 on user 12 took 569953 ticks
Iteration 0 on user 13 took 574957 ticks
Iteration 0 on user 14 took 571613 ticks
Iteration 0 on user 15 took 689333 ticks
Iteration 0 on user 16 took 575583 ticks
Iteration 0 on user 17 took 572916 ticks
Iteration 0 on user 18 took 571689 ticks
Iteration 0 on user 19 took 586870 ticks
Iteration 1 on user 0 took 586157 ticks
Iteration 1 on user 1 took 704850 ticks
Iteration 1 on user 2 took 588264 ticks
Iteration 1 on user 3 took 597516 ticks
Iteration 1 on user 4 took 556524 ticks
Iteration 1 on user 5 took 549306 ticks
Iteration 1 on user 6 took 577862 ticks
Iteration 1 on user 7 took 607852 ticks
Iteration 1 on user 8 took 573033 ticks
Iteration 1 on user 9 took 570894 ticks
Iteration 1 on user 10 took 591351 ticks
Iteration 1 on user 11 took 542364 ticks
Iteration 1 on user 12 took 696476 ticks
Iteration 1 on user 13 took 563922 ticks
Iteration 1 on user 14 took 591406 ticks
Iteration 1 on user 15 took 612330 ticks
Iteration 1 on user 16 took 588935 ticks
Iteration 1 on user 17 took 579628 ticks
Iteration 1 on user 18 took 547209 ticks
Iteration 1 on user 19 took 572841 ticks
Using a summary table for updates to BucketCount, storing BucketExponent in the entity
Seeding DB with 10000 users
Saving changes
Adding 20 users individually
Iteration 0 took 496694 ticks
Iteration 1 took 481554 ticks
Iteration 2 took 458432 ticks
Iteration 3 took 465778 ticks
Iteration 4 took 432873 ticks
Iteration 5 took 450355 ticks
Iteration 6 took 521401 ticks
Iteration 7 took 639188 ticks
Iteration 8 took 549853 ticks
Iteration 9 took 571725 ticks
Iteration 10 took 517472 ticks
Iteration 11 took 507097 ticks
Iteration 12 took 496846 ticks
Iteration 13 took 509583 ticks
Iteration 14 took 546368 ticks
Iteration 15 took 571741 ticks
Iteration 16 took 550490 ticks
Iteration 17 took 551582 ticks
Iteration 18 took 572257 ticks
Iteration 19 took 528604 ticks
Changing 20 users email addresses, twice each
Iteration 0 on user 0 took 567469 ticks
Iteration 0 on user 1 took 562291 ticks
Iteration 0 on user 2 took 576356 ticks
Iteration 0 on user 3 took 574467 ticks
Iteration 0 on user 4 took 656329 ticks
Iteration 0 on user 5 took 578768 ticks
Iteration 0 on user 6 took 546520 ticks
Iteration 0 on user 7 took 662892 ticks
Iteration 0 on user 8 took 595558 ticks
Iteration 0 on user 9 took 581967 ticks
Iteration 0 on user 10 took 562299 ticks
Iteration 0 on user 11 took 577591 ticks
Iteration 0 on user 12 took 584959 ticks
Iteration 0 on user 13 took 561787 ticks
Iteration 0 on user 14 took 565813 ticks
Iteration 0 on user 15 took 595744 ticks
Iteration 0 on user 16 took 558272 ticks
Iteration 0 on user 17 took 567922 ticks
Iteration 0 on user 18 took 571355 ticks
Iteration 0 on user 19 took 551411 ticks
Iteration 1 on user 0 took 559748 ticks
Iteration 1 on user 1 took 686465 ticks
Iteration 1 on user 2 took 672997 ticks
Iteration 1 on user 3 took 568497 ticks
Iteration 1 on user 4 took 600469 ticks
Iteration 1 on user 5 took 553169 ticks
Iteration 1 on user 6 took 592033 ticks
Iteration 1 on user 7 took 568296 ticks
Iteration 1 on user 8 took 578796 ticks
Iteration 1 on user 9 took 627027 ticks
Iteration 1 on user 10 took 572776 ticks
Iteration 1 on user 11 took 545285 ticks
Iteration 1 on user 12 took 561993 ticks
Iteration 1 on user 13 took 601432 ticks
Iteration 1 on user 14 took 563394 ticks
Iteration 1 on user 15 took 580036 ticks
Iteration 1 on user 16 took 581285 ticks
Iteration 1 on user 17 took 592573 ticks
Iteration 1 on user 18 took 663350 ticks
Iteration 1 on user 19 took 667508 ticks
Using autoincremented row numbers
Seeding DB with 10000 users
Saving changes
Adding 20 users individually
Iteration 0 took 527806 ticks
Iteration 1 took 495947 ticks
Iteration 2 took 495161 ticks
Iteration 3 took 452155 ticks
Iteration 4 took 489259 ticks
Iteration 5 took 463202 ticks
Iteration 6 took 478968 ticks
Iteration 7 took 484957 ticks
Iteration 8 took 506948 ticks
Iteration 9 took 537280 ticks
Iteration 10 took 546135 ticks
Iteration 11 took 517191 ticks
Iteration 12 took 512140 ticks
Iteration 13 took 535832 ticks
Iteration 14 took 567705 ticks
Iteration 15 took 552257 ticks
Iteration 16 took 568127 ticks
Iteration 17 took 581180 ticks
Iteration 18 took 565811 ticks
Iteration 19 took 607537 ticks
Changing 20 users email addresses, twice each
Iteration 0 on user 0 took 696708 ticks
Iteration 0 on user 1 took 597562 ticks
Iteration 0 on user 2 took 607319 ticks
Iteration 0 on user 3 took 619309 ticks
Iteration 0 on user 4 took 588902 ticks
Iteration 0 on user 5 took 629805 ticks
Iteration 0 on user 6 took 605183 ticks
Iteration 0 on user 7 took 636651 ticks
Iteration 0 on user 8 took 623951 ticks
Iteration 0 on user 9 took 617522 ticks
Iteration 0 on user 10 took 601382 ticks
Iteration 0 on user 11 took 593421 ticks
Iteration 0 on user 12 took 617203 ticks
Iteration 0 on user 13 took 636417 ticks
Iteration 0 on user 14 took 633931 ticks
Iteration 0 on user 15 took 703901 ticks
Iteration 0 on user 16 took 694188 ticks
Iteration 0 on user 17 took 625739 ticks
Iteration 0 on user 18 took 601390 ticks
Iteration 0 on user 19 took 602140 ticks
Iteration 1 on user 0 took 597033 ticks
Iteration 1 on user 1 took 628559 ticks
Iteration 1 on user 2 took 600428 ticks
Iteration 1 on user 3 took 631475 ticks
Iteration 1 on user 4 took 629882 ticks
Iteration 1 on user 5 took 606151 ticks
Iteration 1 on user 6 took 633460 ticks
Iteration 1 on user 7 took 596644 ticks
Iteration 1 on user 8 took 589913 ticks
Iteration 1 on user 9 took 602694 ticks
Iteration 1 on user 10 took 593982 ticks
Iteration 1 on user 11 took 689053 ticks
Iteration 1 on user 12 took 725575 ticks
Iteration 1 on user 13 took 624966 ticks
Iteration 1 on user 14 took 610032 ticks
Iteration 1 on user 15 took 597431 ticks
Iteration 1 on user 16 took 589901 ticks
Iteration 1 on user 17 took 603427 ticks
Iteration 1 on user 18 took 609961 ticks
Iteration 1 on user 19 took 595591 ticks
Using ‘SELECT COUNT(*)‘ for updates to BucketCount, updating all bucket numbers after threshold is passe
Seeding DB with 10000 users
Saving changes
Adding 20 users individually
Iteration 0 took 148497 ticks
Iteration 1 took 147492 ticks
Iteration 2 took 146788 ticks
Iteration 3 took 144356 ticks
Iteration 4 took 141298 ticks
Iteration 5 took 141571 ticks
Iteration 6 took 144537 ticks
Iteration 7 took 142157 ticks
Iteration 8 took 142044 ticks
Iteration 9 took 145689 ticks
Iteration 10 took 141747 ticks
Iteration 11 took 143082 ticks
Iteration 12 took 144479 ticks
Iteration 13 took 140799 ticks
Iteration 14 took 144471 ticks
Iteration 15 took 143009 ticks
Iteration 16 took 144832 ticks
Iteration 17 took 145232 ticks
Iteration 18 took 143068 ticks
Iteration 19 took 139191 ticks
Changing 20 users email addresses, twice each
Iteration 0 on user 0 took 398503 ticks
Iteration 0 on user 1 took 428717 ticks
Iteration 0 on user 2 took 426820 ticks
Iteration 0 on user 3 took 469030 ticks
Iteration 0 on user 4 took 397868 ticks
Iteration 0 on user 5 took 401153 ticks
Iteration 0 on user 6 took 411593 ticks
Iteration 0 on user 7 took 405212 ticks
Iteration 0 on user 8 took 432713 ticks
Iteration 0 on user 9 took 406179 ticks
Iteration 0 on user 10 took 428969 ticks
Iteration 0 on user 11 took 400971 ticks
Iteration 0 on user 12 took 432941 ticks
Iteration 0 on user 13 took 425581 ticks
Iteration 0 on user 14 took 455612 ticks
Iteration 0 on user 15 took 488452 ticks
Iteration 0 on user 16 took 399306 ticks
Iteration 0 on user 17 took 407058 ticks
Iteration 0 on user 18 took 426049 ticks
Iteration 0 on user 19 took 400497 ticks
Iteration 1 on user 0 took 434770 ticks
Iteration 1 on user 1 took 401420 ticks
Iteration 1 on user 2 took 433653 ticks
Iteration 1 on user 3 took 412297 ticks
Iteration 1 on user 4 took 428861 ticks
Iteration 1 on user 5 took 371386 ticks
Iteration 1 on user 6 took 401724 ticks
Iteration 1 on user 7 took 430672 ticks
Iteration 1 on user 8 took 402821 ticks
Iteration 1 on user 9 took 382900 ticks
Iteration 1 on user 10 took 401758 ticks
Iteration 1 on user 11 took 408212 ticks
Iteration 1 on user 12 took 428541 ticks
Iteration 1 on user 13 took 396445 ticks
Iteration 1 on user 14 took 429265 ticks
Iteration 1 on user 15 took 432520 ticks
Iteration 1 on user 16 took 405929 ticks
Iteration 1 on user 17 took 434607 ticks
Iteration 1 on user 18 took 405513 ticks
Iteration 1 on user 19 took 527175 ticks
Control: Using no encryption
Seeding DB with 10000 users
Saving changes
Adding 30 users individually
Iteration 0 took 264552 ticks
Iteration 1 took 272534 ticks
Iteration 2 took 196142 ticks
Iteration 3 took 230269 ticks
Iteration 4 took 191957 ticks
Iteration 5 took 230848 ticks
Iteration 6 took 222835 ticks
Iteration 7 took 261669 ticks
Iteration 8 took 222356 ticks
Iteration 9 took 252709 ticks
Iteration 10 took 235855 ticks
Iteration 11 took 255211 ticks
Iteration 12 took 318400 ticks
Iteration 13 took 255062 ticks
Iteration 14 took 367318 ticks
Iteration 15 took 266125 ticks
Iteration 16 took 356341 ticks
Iteration 17 took 239011 ticks
Iteration 18 took 240898 ticks
Iteration 19 took 261988 ticks
Iteration 20 took 253363 ticks
Iteration 21 took 223090 ticks
Iteration 22 took 234337 ticks
Iteration 23 took 191441 ticks
Iteration 24 took 225732 ticks
Iteration 25 took 228432 ticks
Iteration 26 took 229246 ticks
Iteration 27 took 247409 ticks
Iteration 28 took 227029 ticks
Iteration 29 took 253975 ticks
Changing 30 users email addresses, twice each
Iteration 0 on user 0 took 230779 ticks
Iteration 0 on user 1 took 257534 ticks
Iteration 0 on user 2 took 193222 ticks
Iteration 0 on user 3 took 240149 ticks
Iteration 0 on user 4 took 223685 ticks
Iteration 0 on user 5 took 244755 ticks
Iteration 0 on user 6 took 220979 ticks
Iteration 0 on user 7 took 246990 ticks
Iteration 0 on user 8 took 228882 ticks
Iteration 0 on user 9 took 192484 ticks
Iteration 0 on user 10 took 331828 ticks
Iteration 0 on user 11 took 229977 ticks
Iteration 0 on user 12 took 247544 ticks
Iteration 0 on user 13 took 190123 ticks
Iteration 0 on user 14 took 224900 ticks
Iteration 0 on user 15 took 230486 ticks
Iteration 0 on user 16 took 237134 ticks
Iteration 0 on user 17 took 196656 ticks
Iteration 0 on user 18 took 227695 ticks
Iteration 0 on user 19 took 256805 ticks
Iteration 0 on user 20 took 226704 ticks
Iteration 0 on user 21 took 195623 ticks
Iteration 0 on user 22 took 227560 ticks
Iteration 0 on user 23 took 236117 ticks
Iteration 0 on user 24 took 254907 ticks
Iteration 0 on user 25 took 236918 ticks
Iteration 0 on user 26 took 274988 ticks
Iteration 0 on user 27 took 319395 ticks
Iteration 0 on user 28 took 274730 ticks
Iteration 0 on user 29 took 222801 ticks
Iteration 1 on user 0 took 259043 ticks
Iteration 1 on user 1 took 223784 ticks
Iteration 1 on user 2 took 227266 ticks
Iteration 1 on user 3 took 227983 ticks
Iteration 1 on user 4 took 213684 ticks
Iteration 1 on user 5 took 240467 ticks
Iteration 1 on user 6 took 259441 ticks
Iteration 1 on user 7 took 258558 ticks
Iteration 1 on user 8 took 267294 ticks
Iteration 1 on user 9 took 245426 ticks
Iteration 1 on user 10 took 228350 ticks
Iteration 1 on user 11 took 196706 ticks
Iteration 1 on user 12 took 257463 ticks
Iteration 1 on user 13 took 259312 ticks
Iteration 1 on user 14 took 255000 ticks
Iteration 1 on user 15 took 232278 ticks
Iteration 1 on user 16 took 197798 ticks
Iteration 1 on user 17 took 230201 ticks
Iteration 1 on user 18 took 232693 ticks
Iteration 1 on user 19 took 227206 ticks
Iteration 1 on user 20 took 230835 ticks
Iteration 1 on user 21 took 248531 ticks
Iteration 1 on user 22 took 231612 ticks
Iteration 1 on user 23 took 226172 ticks
Iteration 1 on user 24 took 228871 ticks
Iteration 1 on user 25 took 245764 ticks
Iteration 1 on user 26 took 250991 ticks
Iteration 1 on user 27 took 250609 ticks
Iteration 1 on user 28 took 226662 ticks
Iteration 1 on user 29 took 233330 ticks
Control: using a fixed number of buckets
Seeding DB with 10000 users
Saving changes
Adding 30 users individually
Iteration 0 took 344969 ticks
Iteration 1 took 369587 ticks
Iteration 2 took 280487 ticks
Iteration 3 took 279136 ticks
Iteration 4 took 254199 ticks
Iteration 5 took 327849 ticks
Iteration 6 took 259161 ticks
Iteration 7 took 246521 ticks
Iteration 8 took 254173 ticks
Iteration 9 took 258897 ticks
Iteration 10 took 260942 ticks
Iteration 11 took 259565 ticks
Iteration 12 took 254959 ticks
Iteration 13 took 256604 ticks
Iteration 14 took 255984 ticks
Iteration 15 took 278966 ticks
Iteration 16 took 281153 ticks
Iteration 17 took 281698 ticks
Iteration 18 took 309252 ticks
Iteration 19 took 279221 ticks
Iteration 20 took 284029 ticks
Iteration 21 took 262243 ticks
Iteration 22 took 279053 ticks
Iteration 23 took 257144 ticks
Iteration 24 took 281735 ticks
Iteration 25 took 286790 ticks
Iteration 26 took 256824 ticks
Iteration 27 took 260766 ticks
Iteration 28 took 286691 ticks
Iteration 29 took 280416 ticks
Changing 30 users email addresses, twice each
Iteration 0 on user 0 took 333301 ticks
Iteration 0 on user 1 took 420901 ticks
Iteration 0 on user 2 took 335661 ticks
Iteration 0 on user 3 took 425062 ticks
Iteration 0 on user 4 took 430089 ticks
Iteration 0 on user 5 took 426252 ticks
Iteration 0 on user 6 took 373689 ticks
Iteration 0 on user 7 took 343007 ticks
Iteration 0 on user 8 took 419829 ticks
Iteration 0 on user 9 took 333849 ticks
Iteration 0 on user 10 took 333658 ticks
Iteration 0 on user 11 took 339943 ticks
Iteration 0 on user 12 took 365036 ticks
Iteration 0 on user 13 took 362573 ticks
Iteration 0 on user 14 took 374191 ticks
Iteration 0 on user 15 took 338678 ticks
Iteration 0 on user 16 took 334825 ticks
Iteration 0 on user 17 took 355100 ticks
Iteration 0 on user 18 took 334292 ticks
Iteration 0 on user 19 took 371261 ticks
Iteration 0 on user 20 took 337689 ticks
Iteration 0 on user 21 took 367165 ticks
Iteration 0 on user 22 took 340859 ticks
Iteration 0 on user 23 took 365587 ticks
Iteration 0 on user 24 took 323291 ticks
Iteration 0 on user 25 took 363486 ticks
Iteration 0 on user 26 took 341169 ticks
Iteration 0 on user 27 took 357752 ticks
Iteration 0 on user 28 took 342658 ticks
Iteration 0 on user 29 took 391103 ticks
Iteration 1 on user 0 took 494957 ticks
Iteration 1 on user 1 took 427864 ticks
Iteration 1 on user 2 took 340976 ticks
Iteration 1 on user 3 took 452829 ticks
Iteration 1 on user 4 took 404169 ticks
Iteration 1 on user 5 took 363498 ticks
Iteration 1 on user 6 took 363257 ticks
Iteration 1 on user 7 took 359768 ticks
Iteration 1 on user 8 took 362517 ticks
Iteration 1 on user 9 took 365168 ticks
Iteration 1 on user 10 took 336336 ticks
Iteration 1 on user 11 took 333634 ticks
Iteration 1 on user 12 took 338319 ticks
Iteration 1 on user 13 took 361474 ticks
Iteration 1 on user 14 took 364048 ticks
Iteration 1 on user 15 took 336131 ticks
Iteration 1 on user 16 took 359817 ticks
Iteration 1 on user 17 took 338403 ticks
Iteration 1 on user 18 took 376470 ticks
Iteration 1 on user 19 took 335447 ticks
Iteration 1 on user 20 took 339368 ticks
Iteration 1 on user 21 took 351060 ticks
Iteration 1 on user 22 took 380106 ticks
Iteration 1 on user 23 took 342076 ticks
Iteration 1 on user 24 took 373168 ticks
Iteration 1 on user 25 took 350884 ticks
Iteration 1 on user 26 took 451760 ticks
Iteration 1 on user 27 took 437960 ticks
Iteration 1 on user 28 took 451747 ticks
Iteration 1 on user 29 took 427155 ticks
Using ‘SELECT COUNT(*)‘ for updates to BucketCount, storing BucketExponent in the entity
Seeding DB with 10000 users
Saving changes
Adding 30 users individually
Iteration 0 took 433765 ticks
Iteration 1 took 465008 ticks
Iteration 2 took 430723 ticks
Iteration 3 took 428786 ticks
Iteration 4 took 455515 ticks
Iteration 5 took 439768 ticks
Iteration 6 took 451847 ticks
Iteration 7 took 456261 ticks
Iteration 8 took 470261 ticks
Iteration 9 took 490109 ticks
Iteration 10 took 472004 ticks
Iteration 11 took 461131 ticks
Iteration 12 took 495565 ticks
Iteration 13 took 526021 ticks
Iteration 14 took 564359 ticks
Iteration 15 took 563134 ticks
Iteration 16 took 662983 ticks
Iteration 17 took 631393 ticks
Iteration 18 took 521224 ticks
Iteration 19 took 561909 ticks
Iteration 20 took 599338 ticks
Iteration 21 took 583317 ticks
Iteration 22 took 585453 ticks
Iteration 23 took 595202 ticks
Iteration 24 took 631929 ticks
Iteration 25 took 584217 ticks
Iteration 26 took 604464 ticks
Iteration 27 took 628795 ticks
Iteration 28 took 657058 ticks
Iteration 29 took 640531 ticks
Changing 30 users email addresses, twice each
Iteration 0 on user 0 took 700521 ticks
Iteration 0 on user 1 took 776460 ticks
Iteration 0 on user 2 took 754279 ticks
Iteration 0 on user 3 took 700200 ticks
Iteration 0 on user 4 took 685331 ticks
Iteration 0 on user 5 took 699493 ticks
Iteration 0 on user 6 took 661320 ticks
Iteration 0 on user 7 took 647949 ticks
Iteration 0 on user 8 took 699092 ticks
Iteration 0 on user 9 took 644498 ticks
Iteration 0 on user 10 took 686893 ticks
Iteration 0 on user 11 took 670781 ticks
Iteration 0 on user 12 took 660828 ticks
Iteration 0 on user 13 took 642390 ticks
Iteration 0 on user 14 took 665316 ticks
Iteration 0 on user 15 took 782730 ticks
Iteration 0 on user 16 took 687988 ticks
Iteration 0 on user 17 took 650472 ticks
Iteration 0 on user 18 took 665095 ticks
Iteration 0 on user 19 took 690130 ticks
Iteration 0 on user 20 took 740846 ticks
Iteration 0 on user 21 took 661343 ticks
Iteration 0 on user 22 took 676806 ticks
Iteration 0 on user 23 took 649228 ticks
Iteration 0 on user 24 took 672671 ticks
Iteration 0 on user 25 took 691919 ticks
Iteration 0 on user 26 took 678891 ticks
Iteration 0 on user 27 took 688776 ticks
Iteration 0 on user 28 took 645767 ticks
Iteration 0 on user 29 took 658729 ticks
Iteration 1 on user 0 took 716961 ticks
Iteration 1 on user 1 took 676218 ticks
Iteration 1 on user 2 took 650350 ticks
Iteration 1 on user 3 took 650080 ticks
Iteration 1 on user 4 took 618899 ticks
Iteration 1 on user 5 took 670102 ticks
Iteration 1 on user 6 took 648291 ticks
Iteration 1 on user 7 took 653182 ticks
Iteration 1 on user 8 took 663665 ticks
Iteration 1 on user 9 took 649316 ticks
Iteration 1 on user 10 took 670172 ticks
Iteration 1 on user 11 took 706561 ticks
Iteration 1 on user 12 took 671783 ticks
Iteration 1 on user 13 took 652298 ticks
Iteration 1 on user 14 took 680640 ticks
Iteration 1 on user 15 took 675692 ticks
Iteration 1 on user 16 took 637566 ticks
Iteration 1 on user 17 took 682552 ticks
Iteration 1 on user 18 took 643766 ticks
Iteration 1 on user 19 took 672817 ticks
Iteration 1 on user 20 took 660208 ticks
Iteration 1 on user 21 took 663200 ticks
Iteration 1 on user 22 took 647659 ticks
Iteration 1 on user 23 took 678155 ticks
Iteration 1 on user 24 took 647936 ticks
Iteration 1 on user 25 took 661820 ticks
Iteration 1 on user 26 took 643096 ticks
Iteration 1 on user 27 took 678524 ticks
Iteration 1 on user 28 took 671955 ticks
Iteration 1 on user 29 took 681097 ticks
Using a summary table for updates to BucketCount, storing BucketExponent in the entity
Seeding DB with 10000 users
Saving changes
Adding 30 users individually
Iteration 0 took 562666 ticks
Iteration 1 took 413370 ticks
Iteration 2 took 470296 ticks
Iteration 3 took 459721 ticks
Iteration 4 took 477406 ticks
Iteration 5 took 497324 ticks
Iteration 6 took 476293 ticks
Iteration 7 took 511791 ticks
Iteration 8 took 472484 ticks
Iteration 9 took 463331 ticks
Iteration 10 took 495557 ticks
Iteration 11 took 529144 ticks
Iteration 12 took 541629 ticks
Iteration 13 took 544276 ticks
Iteration 14 took 529253 ticks
Iteration 15 took 583104 ticks
Iteration 16 took 592635 ticks
Iteration 17 took 668216 ticks
Iteration 18 took 598979 ticks
Iteration 19 took 589808 ticks
Iteration 20 took 608916 ticks
Iteration 21 took 605436 ticks
Iteration 22 took 581062 ticks
Iteration 23 took 591036 ticks
Iteration 24 took 617280 ticks
Iteration 25 took 631422 ticks
Iteration 26 took 641051 ticks
Iteration 27 took 664304 ticks
Iteration 28 took 643450 ticks
Iteration 29 took 663400 ticks
Changing 30 users email addresses, twice each
```

```
Iteration 0 on user 0 took 649952 ticks
Iteration 0 on user 1 took 665703 ticks
Iteration 0 on user 2 took 716613 ticks
Iteration 0 on user 3 took 653674 ticks
Iteration 0 on user 4 took 658532 ticks
Iteration 0 on user 5 took 652042 ticks
Iteration 0 on user 6 took 644569 ticks
Iteration 0 on user 7 took 673851 ticks
Iteration 0 on user 8 took 636803 ticks
Iteration 0 on user 9 took 659396 ticks
Iteration 0 on user 10 took 655111 ticks
Iteration 0 on user 11 took 671153 ticks
Iteration 0 on user 12 took 676861 ticks
Iteration 0 on user 13 took 679650 ticks
Iteration 0 on user 14 took 631288 ticks
Iteration 0 on user 15 took 678900 ticks
Iteration 0 on user 16 took 637159 ticks
Iteration 0 on user 17 took 717085 ticks
Iteration 0 on user 18 took 664470 ticks
Iteration 0 on user 19 took 658903 ticks
Iteration 0 on user 20 took 640186 ticks
Iteration 0 on user 21 took 635339 ticks
Iteration 0 on user 22 took 684289 ticks
Iteration 0 on user 23 took 654490 ticks
Iteration 0 on user 24 took 636735 ticks
Iteration 0 on user 25 took 643000 ticks
Iteration 0 on user 26 took 679900 ticks
Iteration 0 on user 27 took 681894 ticks
Iteration 0 on user 28 took 690491 ticks
Iteration 0 on user 29 took 646523 ticks
Iteration 1 on user 0 took 636066 ticks
Iteration 1 on user 1 took 666149 ticks
Iteration 1 on user 2 took 762182 ticks
Iteration 1 on user 3 took 649559 ticks
Iteration 1 on user 4 took 656544 ticks
Iteration 1 on user 5 took 631687 ticks
Iteration 1 on user 6 took 656728 ticks
Iteration 1 on user 7 took 676491 ticks
Iteration 1 on user 8 took 672371 ticks
Iteration 1 on user 9 took 676688 ticks
Iteration 1 on user 10 took 685571 ticks
Iteration 1 on user 11 took 659580 ticks
Iteration 1 on user 12 took 619270 ticks
Iteration 1 on user 13 took 636384 ticks
Iteration 1 on user 14 took 661257 ticks
Iteration 1 on user 15 took 701869 ticks
Iteration 1 on user 16 took 698294 ticks
Iteration 1 on user 17 took 733867 ticks
Iteration 1 on user 18 took 623102 ticks
Iteration 1 on user 19 took 644372 ticks
Iteration 1 on user 20 took 665552 ticks
Iteration 1 on user 21 took 687373 ticks
Iteration 1 on user 22 took 634587 ticks
Iteration 1 on user 23 took 666640 ticks
Iteration 1 on user 24 took 645550 ticks
Iteration 1 on user 25 took 677272 ticks
Iteration 1 on user 26 took 643060 ticks
Iteration 1 on user 27 took 647790 ticks
Iteration 1 on user 28 took 631676 ticks
Iteration 1 on user 29 took 658711 ticks
Using autoincremented row numbers
Seeding DB with 10000 users
Saving changes
Adding 30 users individually
Iteration 0 took 473411 ticks
Iteration 1 took 457822 ticks
Iteration 2 took 482259 ticks
Iteration 3 took 485726 ticks
Iteration 4 took 491042 ticks
Iteration 5 took 505936 ticks
Iteration 6 took 533562 ticks
Iteration 7 took 569445 ticks
Iteration 8 took 510327 ticks
Iteration 9 took 495208 ticks
Iteration 10 took 511085 ticks
Iteration 11 took 506862 ticks
Iteration 12 took 532242 ticks
Iteration 13 took 541280 ticks
Iteration 14 took 585775 ticks
Iteration 15 took 553055 ticks
Iteration 16 took 618564 ticks
Iteration 17 took 565550 ticks
Iteration 18 took 591742 ticks
Iteration 19 took 607729 ticks
Iteration 20 took 637514 ticks
Iteration 21 took 634746 ticks
Iteration 22 took 618078 ticks
Iteration 23 took 618641 ticks
Iteration 24 took 615011 ticks
Iteration 25 took 676908 ticks
Iteration 26 took 643936 ticks
Iteration 27 took 690219 ticks
Iteration 28 took 662537 ticks
Iteration 29 took 697401 ticks
Changing 30 users email addresses, twice each
Iteration 0 on user 0 took 673603 ticks
Iteration 0 on user 1 took 687856 ticks
Iteration 0 on user 2 took 709009 ticks
Iteration 0 on user 3 took 732749 ticks
Iteration 0 on user 4 took 706024 ticks
Iteration 0 on user 5 took 697248 ticks
Iteration 0 on user 6 took 695141 ticks
Iteration 0 on user 7 took 684072 ticks
Iteration 0 on user 8 took 707543 ticks
Iteration 0 on user 9 took 843192 ticks
Iteration 0 on user 10 took 722495 ticks
Iteration 0 on user 11 took 718582 ticks
Iteration 0 on user 12 took 667671 ticks
Iteration 0 on user 13 took 703140 ticks
Iteration 0 on user 14 took 661839 ticks
Iteration 0 on user 15 took 690294 ticks
Iteration 0 on user 16 took 671264 ticks
Iteration 0 on user 17 took 676319 ticks
Iteration 0 on user 18 took 694324 ticks
Iteration 0 on user 19 took 688706 ticks
Iteration 0 on user 20 took 701476 ticks
Iteration 0 on user 21 took 706174 ticks
Iteration 0 on user 22 took 671842 ticks
Iteration 0 on user 23 took 735822 ticks
Iteration 0 on user 24 took 785910 ticks
Iteration 0 on user 25 took 692581 ticks
Iteration 0 on user 26 took 692457 ticks
Iteration 0 on user 27 took 671852 ticks
Iteration 0 on user 28 took 693687 ticks
Iteration 0 on user 29 took 673773 ticks
Iteration 1 on user 0 took 683596 ticks
Iteration 1 on user 1 took 700406 ticks
Iteration 1 on user 2 took 667365 ticks
Iteration 1 on user 3 took 695571 ticks
Iteration 1 on user 4 took 705743 ticks
Iteration 1 on user 5 took 698814 ticks
Iteration 1 on user 6 took 681867 ticks
Iteration 1 on user 7 took 674155 ticks
Iteration 1 on user 8 took 830182 ticks
Iteration 1 on user 9 took 710571 ticks
Iteration 1 on user 10 took 693141 ticks
Iteration 1 on user 11 took 725471 ticks
Iteration 1 on user 12 took 671401 ticks
Iteration 1 on user 13 took 674041 ticks
Iteration 1 on user 14 took 670134 ticks
Iteration 1 on user 15 took 698585 ticks
Iteration 1 on user 16 took 672016 ticks
Iteration 1 on user 17 took 711959 ticks
Iteration 1 on user 18 took 675174 ticks
Iteration 1 on user 19 took 687172 ticks
Iteration 1 on user 20 took 716115 ticks
Iteration 1 on user 21 took 705533 ticks
Iteration 1 on user 22 took 704939 ticks
Iteration 1 on user 23 took 712502 ticks
Iteration 1 on user 24 took 687119 ticks
Iteration 1 on user 25 took 705133 ticks
Iteration 1 on user 26 took 711468 ticks
Iteration 1 on user 27 took 700098 ticks
Iteration 1 on user 28 took 702249 ticks
Iteration 1 on user 29 took 719745 ticks
Using ‘SELECT COUNT(*)‘ for updates to BucketCount, updating all bucket numbers after threshold is passe
Seeding DB with 10000 users
Saving changes
Adding 30 users individually
Iteration 0 took 146581 ticks
Iteration 1 took 145142 ticks
Iteration 2 took 145545 ticks
Iteration 3 took 145584 ticks
Iteration 4 took 142154 ticks
Iteration 5 took 146218 ticks
Iteration 6 took 145007 ticks
Iteration 7 took 139171 ticks
Iteration 8 took 145237 ticks
Iteration 9 took 145853 ticks
Iteration 10 took 143038 ticks
Iteration 11 took 143317 ticks
Iteration 12 took 135793 ticks
Iteration 13 took 143262 ticks
Iteration 14 took 142973 ticks
Iteration 15 took 139292 ticks
Iteration 16 took 140295 ticks
Iteration 17 took 137361 ticks
Iteration 18 took 143751 ticks
Iteration 19 took 139927 ticks
Iteration 20 took 139092 ticks
Iteration 21 took 144012 ticks
Iteration 22 took 140557 ticks
Iteration 23 took 140612 ticks
Iteration 24 took 140781 ticks
Iteration 25 took 142184 ticks
Iteration 26 took 145430 ticks
Iteration 27 took 139471 ticks
Iteration 28 took 138581 ticks
Iteration 29 took 135978 ticks
Changing 30 users email addresses, twice each
Iteration 0 on user 0 took 397351 ticks
Iteration 0 on user 1 took 428300 ticks
Iteration 0 on user 2 took 439593 ticks
Iteration 0 on user 3 took 431984 ticks
Iteration 0 on user 4 took 437047 ticks
Iteration 0 on user 5 took 409883 ticks
Iteration 0 on user 6 took 438770 ticks
Iteration 0 on user 7 took 429702 ticks
Iteration 0 on user 8 took 429805 ticks
Iteration 0 on user 9 took 427245 ticks
Iteration 0 on user 10 took 439875 ticks
Iteration 0 on user 11 took 399630 ticks
Iteration 0 on user 12 took 435915 ticks
Iteration 0 on user 13 took 403622 ticks
Iteration 0 on user 14 took 465086 ticks
Iteration 0 on user 15 took 436040 ticks
Iteration 0 on user 16 took 531927 ticks
Iteration 0 on user 17 took 540251 ticks
Iteration 0 on user 18 took 439490 ticks
Iteration 0 on user 19 took 407586 ticks
Iteration 0 on user 20 took 434153 ticks
Iteration 0 on user 21 took 412302 ticks
Iteration 0 on user 22 took 410633 ticks
Iteration 0 on user 23 took 411027 ticks
Iteration 0 on user 24 took 426285 ticks
Iteration 0 on user 25 took 439262 ticks
Iteration 0 on user 26 took 405692 ticks
Iteration 0 on user 27 took 401057 ticks
Iteration 0 on user 28 took 403917 ticks
Iteration 0 on user 29 took 426699 ticks
Iteration 1 on user 0 took 420570 ticks
Iteration 1 on user 1 took 419886 ticks
Iteration 1 on user 2 took 401619 ticks
Iteration 1 on user 3 took 397800 ticks
Iteration 1 on user 4 took 391290 ticks
Iteration 1 on user 5 took 429099 ticks
Iteration 1 on user 6 took 468808 ticks
Iteration 1 on user 7 took 473819 ticks
Iteration 1 on user 8 took 385939 ticks
Iteration 1 on user 9 took 401977 ticks
Iteration 1 on user 10 took 442845 ticks
Iteration 1 on user 11 took 429954 ticks
Iteration 1 on user 12 took 433737 ticks
Iteration 1 on user 13 took 432042 ticks
Iteration 1 on user 14 took 403487 ticks
Iteration 1 on user 15 took 403366 ticks
Iteration 1 on user 16 took 433181 ticks
Iteration 1 on user 17 took 421322 ticks
Iteration 1 on user 18 took 394241 ticks
Iteration 1 on user 19 took 424206 ticks
Iteration 1 on user 20 took 395359 ticks
Iteration 1 on user 21 took 418268 ticks
Iteration 1 on user 22 took 404533 ticks
Iteration 1 on user 23 took 428491 ticks
Iteration 1 on user 24 took 420981 ticks
Iteration 1 on user 25 took 426999 ticks
Iteration 1 on user 26 took 400878 ticks
Iteration 1 on user 27 took 400636 ticks
Iteration 1 on user 28 took 432114 ticks
Iteration 1 on user 29 took 391531 ticks
Control: Using no encryption
Seeding DB with 100000 users
Saving changes
Adding 10 users individually
Iteration 0 took 919421 ticks
Iteration 1 took 896792 ticks
Iteration 2 took 844699 ticks
Iteration 3 took 921934 ticks
Iteration 4 took 850323 ticks
Iteration 5 took 899731 ticks
Iteration 6 took 831369 ticks
Iteration 7 took 797110 ticks
Iteration 8 took 735053 ticks
Iteration 9 took 744649 ticks
Changing 10 users email addresses, twice each
Iteration 0 on user 0 took 773428 ticks
Iteration 0 on user 1 took 745068 ticks
Iteration 0 on user 2 took 767191 ticks
Iteration 0 on user 3 took 747316 ticks
Iteration 0 on user 4 took 785324 ticks
Iteration 0 on user 5 took 761662 ticks
Iteration 0 on user 6 took 742172 ticks
Iteration 0 on user 7 took 736877 ticks
Iteration 0 on user 8 took 748642 ticks
Iteration 0 on user 9 took 718319 ticks
Iteration 1 on user 0 took 735774 ticks
Iteration 1 on user 1 took 742567 ticks
Iteration 1 on user 2 took 751567 ticks
Iteration 1 on user 3 took 738423 ticks
Iteration 1 on user 4 took 738935 ticks
Iteration 1 on user 5 took 730574 ticks
Iteration 1 on user 6 took 775157 ticks
Iteration 1 on user 7 took 826330 ticks
Iteration 1 on user 8 took 746783 ticks
Iteration 1 on user 9 took 742629 ticks
Control: using a fixed number of buckets
Seeding DB with 100000 users
Saving changes
Adding 10 users individually
Iteration 0 took 945817 ticks
Iteration 1 took 916931 ticks
Iteration 2 took 934679 ticks
Iteration 3 took 933130 ticks
Iteration 4 took 901959 ticks
Iteration 5 took 902486 ticks
Iteration 6 took 928632 ticks
Iteration 7 took 912577 ticks
Iteration 8 took 882783 ticks
Iteration 9 took 875813 ticks
Changing 10 users email addresses, twice each
Iteration 0 on user 0 took 1719240 ticks
Iteration 0 on user 1 took 1734223 ticks
Iteration 0 on user 2 took 1691780 ticks
Iteration 0 on user 3 took 1675201 ticks
Iteration 0 on user 4 took 1685144 ticks
Iteration 0 on user 5 took 1671399 ticks
Iteration 0 on user 6 took 1772041 ticks
Iteration 0 on user 7 took 1659708 ticks
Iteration 0 on user 8 took 1701578 ticks
Iteration 0 on user 9 took 1644581 ticks
Iteration 1 on user 0 took 1630748 ticks
Iteration 1 on user 1 took 1676241 ticks
Iteration 1 on user 2 took 1702035 ticks
Iteration 1 on user 3 took 1656577 ticks
Iteration 1 on user 4 took 1678496 ticks
Iteration 1 on user 5 took 1642777 ticks
Iteration 1 on user 6 took 1677346 ticks
Iteration 1 on user 7 took 1657296 ticks
Iteration 1 on user 8 took 1698857 ticks
Iteration 1 on user 9 took 1665213 ticks
Using ‘SELECT COUNT(*)‘ for updates to BucketCount, storing BucketExponent in the entity
Seeding DB with 100000 users
Saving changes
Adding 10 users individually
```

```
Iteration 0 took 1360660 ticks
Iteration 1 took 1290918 ticks
Iteration 2 took 1258620 ticks
Iteration 3 took 1455900 ticks
Iteration 4 took 1249564 ticks
Iteration 5 took 1327156 ticks
Iteration 6 took 1305167 ticks
Iteration 7 took 1244757 ticks
Iteration 8 took 1270278 ticks
Iteration 9 took 1331129 ticks
Changing 10 users email addresses, twice each
Iteration 0 on user 0 took 1241636 ticks
Iteration 0 on user 1 took 1313937 ticks
Iteration 0 on user 2 took 1331372 ticks
Iteration 0 on user 3 took 1300749 ticks
Iteration 0 on user 4 took 1267440 ticks
Iteration 0 on user 5 took 1300162 ticks
Iteration 0 on user 6 took 1300327 ticks
Iteration 0 on user 7 took 1333592 ticks
Iteration 0 on user 8 took 1330649 ticks
Iteration 0 on user 9 took 1291240 ticks
Iteration 1 on user 0 took 1317637 ticks
Iteration 1 on user 1 took 1290073 ticks
Iteration 1 on user 2 took 1344030 ticks
Iteration 1 on user 3 took 1376478 ticks
Iteration 1 on user 4 took 1301346 ticks
Iteration 1 on user 5 took 1268141 ticks
Iteration 1 on user 6 took 1311524 ticks
Iteration 1 on user 7 took 1289612 ticks
Iteration 1 on user 8 took 1342534 ticks
Iteration 1 on user 9 took 1340750 ticks
Using a summary table for updates to BucketCount, storing BucketExponent in the entity
Seeding DB with 100000 users
Saving changes
Adding 10 users individually
Iteration 0 took 1390028 ticks
Iteration 1 took 1275296 ticks
Iteration 2 took 1266608 ticks
Iteration 3 took 1223523 ticks
Iteration 4 took 1265026 ticks
Iteration 5 took 1303297 ticks
Iteration 6 took 1219643 ticks
Iteration 7 took 1263274 ticks
Iteration 8 took 1315090 ticks
Iteration 9 took 1242213 ticks
Changing 10 users email addresses, twice each
Iteration 0 on user 0 took 1250582 ticks
Iteration 0 on user 1 took 1248673 ticks
Iteration 0 on user 2 took 1258583 ticks
Iteration 0 on user 3 took 1283447 ticks
Iteration 0 on user 4 took 1262144 ticks
Iteration 0 on user 5 took 1274505 ticks
Iteration 0 on user 6 took 1298462 ticks
Iteration 0 on user 7 took 1246225 ticks
Iteration 0 on user 8 took 1231688 ticks
Iteration 0 on user 9 took 1230748 ticks
Iteration 1 on user 0 took 1260685 ticks
Iteration 1 on user 1 took 1246118 ticks
Iteration 1 on user 2 took 1307353 ticks
Iteration 1 on user 3 took 1258168 ticks
Iteration 1 on user 4 took 1312739 ticks
Iteration 1 on user 5 took 1252563 ticks
Iteration 1 on user 6 took 1240445 ticks
Iteration 1 on user 7 took 1339072 ticks
Iteration 1 on user 8 took 1275715 ticks
Iteration 1 on user 9 took 1281396 ticks
Using autoincremented row numbers
Seeding DB with 100000 users
Saving changes
Adding 10 users individually
Iteration 0 took 1903525 ticks
Iteration 1 took 1779925 ticks
Iteration 2 took 1518949 ticks
Iteration 3 took 1785779 ticks
Iteration 4 took 1602841 ticks
Iteration 5 took 1644235 ticks
Iteration 6 took 1531513 ticks
Iteration 7 took 1694202 ticks
Iteration 8 took 1700648 ticks
Iteration 9 took 1664700 ticks
Changing 10 users email addresses, twice each
Iteration 0 on user 0 took 1853271 ticks
Iteration 0 on user 1 took 1922356 ticks
Iteration 0 on user 2 took 1839070 ticks
Iteration 0 on user 3 took 1972204 ticks
Iteration 0 on user 4 took 1726922 ticks
Iteration 0 on user 5 took 1658494 ticks
Iteration 0 on user 6 took 1924619 ticks
Iteration 0 on user 7 took 1922047 ticks
Iteration 0 on user 8 took 1614434 ticks
Iteration 0 on user 9 took 1850811 ticks
Iteration 1 on user 0 took 1875805 ticks
Iteration 1 on user 1 took 1635974 ticks
Iteration 1 on user 2 took 1776286 ticks
Iteration 1 on user 3 took 1835272 ticks
Iteration 1 on user 4 took 1856904 ticks
Iteration 1 on user 5 took 1620373 ticks
Iteration 1 on user 6 took 1922867 ticks
Iteration 1 on user 7 took 1860391 ticks
Iteration 1 on user 8 took 1686158 ticks
Iteration 1 on user 9 took 1609941 ticks
Using ‘SELECT COUNT(*)‘ for updates to BucketCount, updating all bucket numbers after threshold is passe
Seeding DB with 100000 users
Saving changes
Adding 10 users individually
Iteration 0 took 1029653 ticks
Iteration 1 took 1063913 ticks
Iteration 2 took 1051285 ticks
Iteration 3 took 1051043 ticks
Iteration 4 took 1056600 ticks
Iteration 5 took 1043047 ticks
Iteration 6 took 1075884 ticks
Iteration 7 took 1053861 ticks
Iteration 8 took 1079466 ticks
Iteration 9 took 1093273 ticks
Changing 10 users email addresses, twice each
Iteration 0 on user 0 took 2210119 ticks
Iteration 0 on user 1 took 2195455 ticks
Iteration 0 on user 2 took 2245411 ticks
Iteration 0 on user 3 took 2192961 ticks
Iteration 0 on user 4 took 2192619 ticks
Iteration 0 on user 5 took 2240931 ticks
Iteration 0 on user 6 took 2261365 ticks
Iteration 0 on user 7 took 2208519 ticks
Iteration 0 on user 8 took 2199382 ticks
Iteration 0 on user 9 took 2184369 ticks
Iteration 1 on user 0 took 2237993 ticks
Iteration 1 on user 1 took 2228500 ticks
Iteration 1 on user 2 took 2212558 ticks
Iteration 1 on user 3 took 2221784 ticks
Iteration 1 on user 4 took 2207982 ticks
Iteration 1 on user 5 took 2349524 ticks
Iteration 1 on user 6 took 2212264 ticks
Iteration 1 on user 7 took 2160318 ticks
Iteration 1 on user 8 took 2170677 ticks
Iteration 1 on user 9 took 2211956 ticks
Control: Using no encryption
Seeding DB with 100000 users
Saving changes
Adding 20 users individually
Iteration 0 took 763871 ticks
Iteration 1 took 727112 ticks
Iteration 2 took 742188 ticks
Iteration 3 took 736266 ticks
Iteration 4 took 740252 ticks
Iteration 5 took 712522 ticks
Iteration 6 took 739263 ticks
Iteration 7 took 732277 ticks
Iteration 8 took 734059 ticks
Iteration 9 took 735172 ticks
Iteration 10 took 736575 ticks
Iteration 11 took 760382 ticks
Iteration 12 took 769613 ticks
Iteration 13 took 758979 ticks
Iteration 14 took 737370 ticks
Iteration 15 took 744871 ticks
Iteration 16 took 740703 ticks
Iteration 17 took 738910 ticks
Iteration 18 took 734348 ticks
Iteration 19 took 744935 ticks
Changing 20 users email addresses, twice each
Iteration 0 on user 0 took 734583 ticks
Iteration 0 on user 1 took 730111 ticks
Iteration 0 on user 2 took 736487 ticks
Iteration 0 on user 3 took 728266 ticks
Iteration 0 on user 4 took 767434 ticks
Iteration 0 on user 5 took 755193 ticks
Iteration 0 on user 6 took 721719 ticks
Iteration 0 on user 7 took 799769 ticks
Iteration 0 on user 8 took 762712 ticks
Iteration 0 on user 9 took 741507 ticks
Iteration 0 on user 10 took 730597 ticks
Iteration 0 on user 11 took 724681 ticks
Iteration 0 on user 12 took 748062 ticks
Iteration 0 on user 13 took 750615 ticks
Iteration 0 on user 14 took 726254 ticks
Iteration 0 on user 15 took 738523 ticks
Iteration 0 on user 16 took 706944 ticks
Iteration 0 on user 17 took 730906 ticks
Iteration 0 on user 18 took 741144 ticks
Iteration 0 on user 19 took 729823 ticks
Iteration 1 on user 0 took 721703 ticks
Iteration 1 on user 1 took 735953 ticks
Iteration 1 on user 2 took 730597 ticks
Iteration 1 on user 3 took 727796 ticks
Iteration 1 on user 4 took 722283 ticks
Iteration 1 on user 5 took 733606 ticks
Iteration 1 on user 6 took 737921 ticks
Iteration 1 on user 7 took 717224 ticks
Iteration 1 on user 8 took 726252 ticks
Iteration 1 on user 9 took 724982 ticks
Iteration 1 on user 10 took 739028 ticks
Iteration 1 on user 11 took 719921 ticks
Iteration 1 on user 12 took 752635 ticks
Iteration 1 on user 13 took 729008 ticks
Iteration 1 on user 14 took 745791 ticks
Iteration 1 on user 15 took 723053 ticks
Iteration 1 on user 16 took 729812 ticks
Iteration 1 on user 17 took 719733 ticks
Iteration 1 on user 18 took 733321 ticks
Iteration 1 on user 19 took 722781 ticks
Control: using a fixed number of buckets
Seeding DB with 100000 users
Saving changes
Adding 20 users individually
Iteration 0 took 920104 ticks
Iteration 1 took 903647 ticks
Iteration 2 took 919805 ticks
Iteration 3 took 916558 ticks
Iteration 4 took 903049 ticks
Iteration 5 took 930772 ticks
Iteration 6 took 893846 ticks
Iteration 7 took 921708 ticks
Iteration 8 took 882560 ticks
Iteration 9 took 929607 ticks
Iteration 10 took 958017 ticks
Iteration 11 took 1021257 ticks
Iteration 12 took 949291 ticks
Iteration 13 took 917297 ticks
Iteration 14 took 954549 ticks
Iteration 15 took 902253 ticks
Iteration 16 took 909317 ticks
Iteration 17 took 901251 ticks
Iteration 18 took 952677 ticks
Iteration 19 took 930946 ticks
Changing 20 users email addresses, twice each
Iteration 0 on user 0 took 1702434 ticks
Iteration 0 on user 1 took 1643194 ticks
Iteration 0 on user 2 took 1687821 ticks
Iteration 0 on user 3 took 1644802 ticks
Iteration 0 on user 4 took 1680921 ticks
Iteration 0 on user 5 took 1704535 ticks
Iteration 0 on user 6 took 1804306 ticks
Iteration 0 on user 7 took 1668864 ticks
Iteration 0 on user 8 took 1671732 ticks
Iteration 0 on user 9 took 1666674 ticks
Iteration 0 on user 10 took 1662683 ticks
Iteration 0 on user 11 took 1648740 ticks
Iteration 0 on user 12 took 1653074 ticks
Iteration 0 on user 13 took 1619301 ticks
Iteration 0 on user 14 took 1650252 ticks
Iteration 0 on user 15 took 1654153 ticks
Iteration 0 on user 16 took 1662419 ticks
Iteration 0 on user 17 took 1670670 ticks
Iteration 0 on user 18 took 1673617 ticks
Iteration 0 on user 19 took 1678505 ticks
Iteration 1 on user 0 took 1665758 ticks
Iteration 1 on user 1 took 1674433 ticks
Iteration 1 on user 2 took 1657027 ticks
Iteration 1 on user 3 took 1645238 ticks
Iteration 1 on user 4 took 1666053 ticks
Iteration 1 on user 5 took 1654329 ticks
Iteration 1 on user 6 took 1647634 ticks
Iteration 1 on user 7 took 1631628 ticks
Iteration 1 on user 8 took 1659951 ticks
Iteration 1 on user 9 took 1649131 ticks
Iteration 1 on user 10 took 1685309 ticks
Iteration 1 on user 11 took 1667023 ticks
Iteration 1 on user 12 took 1671910 ticks
Iteration 1 on user 13 took 1660086 ticks
Iteration 1 on user 14 took 1656695 ticks
Iteration 1 on user 15 took 1648610 ticks
Iteration 1 on user 16 took 1661435 ticks
Iteration 1 on user 17 took 2486489 ticks
Iteration 1 on user 18 took 1667547 ticks
Iteration 1 on user 19 took 1636938 ticks
Using ‘SELECT COUNT(*)‘ for updates to BucketCount, storing BucketExponent in the entity
Seeding DB with 100000 users
Saving changes
Adding 20 users individually
```

```
Iteration 0 took 1387899 ticks
Iteration 1 took 1224686 ticks
Iteration 2 took 1285027 ticks
Iteration 3 took 1300110 ticks
Iteration 4 took 1276744 ticks
Iteration 5 took 1269434 ticks
Iteration 6 took 1301670 ticks
Iteration 7 took 1247848 ticks
Iteration 8 took 1312603 ticks
Iteration 9 took 1259514 ticks
Iteration 10 took 1287828 ticks
Iteration 11 took 1273909 ticks
Iteration 12 took 1343095 ticks
Iteration 13 took 1326769 ticks
Iteration 14 took 1363010 ticks
Iteration 15 took 1364791 ticks
Iteration 16 took 1374644 ticks
Iteration 17 took 1319410 ticks
Iteration 18 took 1439484 ticks
Iteration 19 took 1405681 ticks
Changing 20 users email addresses, twice each
Iteration 0 on user 0 took 1367204 ticks
Iteration 0 on user 1 took 1424226 ticks
Iteration 0 on user 2 took 1408846 ticks
Iteration 0 on user 3 took 1371674 ticks
Iteration 0 on user 4 took 1345333 ticks
Iteration 0 on user 5 took 1397714 ticks
Iteration 0 on user 6 took 1337786 ticks
Iteration 0 on user 7 took 1382865 ticks
Iteration 0 on user 8 took 1437416 ticks
Iteration 0 on user 9 took 1356868 ticks
Iteration 0 on user 10 took 1437126 ticks
Iteration 0 on user 11 took 1360802 ticks
Iteration 0 on user 12 took 1350035 ticks
Iteration 0 on user 13 took 1369262 ticks
Iteration 0 on user 14 took 1395396 ticks
Iteration 0 on user 15 took 1390383 ticks
Iteration 0 on user 16 took 1395625 ticks
Iteration 0 on user 17 took 1447019 ticks
Iteration 0 on user 18 took 1396382 ticks
Iteration 0 on user 19 took 1389753 ticks
Iteration 1 on user 0 took 1378190 ticks
Iteration 1 on user 1 took 1388008 ticks
Iteration 1 on user 2 took 1393434 ticks
Iteration 1 on user 3 took 1402631 ticks
Iteration 1 on user 4 took 1414688 ticks
Iteration 1 on user 5 took 1360746 ticks
Iteration 1 on user 6 took 1704924 ticks
Iteration 1 on user 7 took 1657785 ticks
Iteration 1 on user 8 took 1423014 ticks
Iteration 1 on user 9 took 1421434 ticks
Iteration 1 on user 10 took 1370479 ticks
Iteration 1 on user 11 took 1459267 ticks
Iteration 1 on user 12 took 1431848 ticks
Iteration 1 on user 13 took 1397570 ticks
Iteration 1 on user 14 took 1660919 ticks
Iteration 1 on user 15 took 1643170 ticks
Iteration 1 on user 16 took 1506424 ticks
Iteration 1 on user 17 took 1385293 ticks
Iteration 1 on user 18 took 1370655 ticks
Iteration 1 on user 19 took 1442740 ticks
Using a summary table for updates to BucketCount, storing BucketExponent in the entity
Seeding DB with 100000 users
Saving changes
Adding 20 users individually
Iteration 0 took 1312741 ticks
Iteration 1 took 1324025 ticks
Iteration 2 took 1273097 ticks
Iteration 3 took 1331170 ticks
Iteration 4 took 1281368 ticks
Iteration 5 took 1240545 ticks
Iteration 6 took 1261445 ticks
Iteration 7 took 1208925 ticks
Iteration 8 took 1300741 ticks
Iteration 9 took 1253005 ticks
Iteration 10 took 1279384 ticks
Iteration 11 took 1272549 ticks
Iteration 12 took 1284685 ticks
Iteration 13 took 1321260 ticks
Iteration 14 took 1322269 ticks
Iteration 15 took 1293361 ticks
Iteration 16 took 1366890 ticks
Iteration 17 took 1405442 ticks
Iteration 18 took 1297701 ticks
Iteration 19 took 1373061 ticks
Changing 20 users email addresses, twice each
Iteration 0 on user 0 took 1317808 ticks
Iteration 0 on user 1 took 1325126 ticks
Iteration 0 on user 2 took 1366211 ticks
Iteration 0 on user 3 took 1336166 ticks
Iteration 0 on user 4 took 1328968 ticks
Iteration 0 on user 5 took 1388740 ticks
Iteration 0 on user 6 took 1317886 ticks
Iteration 0 on user 7 took 1332538 ticks
Iteration 0 on user 8 took 1319160 ticks
Iteration 0 on user 9 took 1300171 ticks
Iteration 0 on user 10 took 1359668 ticks
Iteration 0 on user 11 took 1467043 ticks
Iteration 0 on user 12 took 1352319 ticks
Iteration 0 on user 13 took 1309256 ticks
Iteration 0 on user 14 took 1374979 ticks
Iteration 0 on user 15 took 1297545 ticks
Iteration 0 on user 16 took 1357450 ticks
Iteration 0 on user 17 took 1368329 ticks
Iteration 0 on user 18 took 1335260 ticks
Iteration 0 on user 19 took 1355539 ticks
Iteration 1 on user 0 took 1324122 ticks
Iteration 1 on user 1 took 1327836 ticks
Iteration 1 on user 2 took 1335253 ticks
Iteration 1 on user 3 took 1335884 ticks
Iteration 1 on user 4 took 1324265 ticks
Iteration 1 on user 5 took 1614156 ticks
Iteration 1 on user 6 took 1572804 ticks
Iteration 1 on user 7 took 1365959 ticks
Iteration 1 on user 8 took 1626175 ticks
Iteration 1 on user 9 took 1357411 ticks
Iteration 1 on user 10 took 1346356 ticks
Iteration 1 on user 11 took 1372401 ticks
Iteration 1 on user 12 took 1588885 ticks
Iteration 1 on user 13 took 1558801 ticks
Iteration 1 on user 14 took 1589534 ticks
Iteration 1 on user 15 took 1598799 ticks
Iteration 1 on user 16 took 1318123 ticks
Iteration 1 on user 17 took 1607952 ticks
Iteration 1 on user 18 took 1390300 ticks
Iteration 1 on user 19 took 1380837 ticks
Using autoincremented row numbers
Seeding DB with 100000 users
Saving changes
Adding 20 users individually
Iteration 0 took 1938040 ticks
Iteration 1 took 1495896 ticks
Iteration 2 took 1530929 ticks
Iteration 3 took 1823080 ticks
Iteration 4 took 1863918 ticks
Iteration 5 took 1516759 ticks
Iteration 6 took 1661721 ticks
Iteration 7 took 1810390 ticks
Iteration 8 took 1582049 ticks
Iteration 9 took 1587597 ticks
Iteration 10 took 1903483 ticks
Iteration 11 took 1606502 ticks
Iteration 12 took 1638753 ticks
Iteration 13 took 1884307 ticks
Iteration 14 took 1788809 ticks
Iteration 15 took 1920640 ticks
Iteration 16 took 1907707 ticks
Iteration 17 took 1704676 ticks
Iteration 18 took 1902468 ticks
Iteration 19 took 1633715 ticks
Changing 20 users email addresses, twice each
Iteration 0 on user 0 took 1928798 ticks
Iteration 0 on user 1 took 1706799 ticks
Iteration 0 on user 2 took 1743040 ticks
Iteration 0 on user 3 took 1831894 ticks
Iteration 0 on user 4 took 1930754 ticks
Iteration 0 on user 5 took 2004495 ticks
Iteration 0 on user 6 took 1790976 ticks
Iteration 0 on user 7 took 1963229 ticks
Iteration 0 on user 8 took 1741102 ticks
Iteration 0 on user 9 took 1897479 ticks
Iteration 0 on user 10 took 1983233 ticks
Iteration 0 on user 11 took 1930899 ticks
Iteration 0 on user 12 took 1990017 ticks
Iteration 0 on user 13 took 1702229 ticks
Iteration 0 on user 14 took 1899515 ticks
Iteration 0 on user 15 took 1770798 ticks
Iteration 0 on user 16 took 1735783 ticks
Iteration 0 on user 17 took 1939213 ticks
Iteration 0 on user 18 took 1979084 ticks
Iteration 0 on user 19 took 1680232 ticks
Iteration 1 on user 0 took 1937716 ticks
Iteration 1 on user 1 took 1745848 ticks
Iteration 1 on user 2 took 1735117 ticks
Iteration 1 on user 3 took 1959821 ticks
Iteration 1 on user 4 took 1924845 ticks
Iteration 1 on user 5 took 1909836 ticks
Iteration 1 on user 6 took 2004265 ticks
Iteration 1 on user 7 took 1811478 ticks
Iteration 1 on user 8 took 1809406 ticks
Iteration 1 on user 9 took 1699962 ticks
Iteration 1 on user 10 took 1931296 ticks
Iteration 1 on user 11 took 1926636 ticks
Iteration 1 on user 12 took 1943062 ticks
Iteration 1 on user 13 took 1719976 ticks
Iteration 1 on user 14 took 1861530 ticks
Iteration 1 on user 15 took 1945361 ticks
Iteration 1 on user 16 took 1923180 ticks
Iteration 1 on user 17 took 1932075 ticks
Iteration 1 on user 18 took 1916352 ticks
Iteration 1 on user 19 took 1714617 ticks
Using ‘SELECT COUNT(*)‘ for updates to BucketCount, updating all bucket numbers after threshold is passe
Seeding DB with 100000 users
Saving changes
Adding 20 users individually
Iteration 0 took 1069900 ticks
Iteration 1 took 1072321 ticks
Iteration 2 took 1097464 ticks
Iteration 3 took 1079708 ticks
Iteration 4 took 1092340 ticks
Iteration 5 took 1084890 ticks
Iteration 6 took 1083340 ticks
Iteration 7 took 1043541 ticks
Iteration 8 took 1038061 ticks
Iteration 9 took 1079528 ticks
Iteration 10 took 1054162 ticks
Iteration 11 took 1053956 ticks
Iteration 12 took 1040399 ticks
Iteration 13 took 1051169 ticks
Iteration 14 took 1049413 ticks
Iteration 15 took 1051812 ticks
Iteration 16 took 1038460 ticks
Iteration 17 took 1043997 ticks
Iteration 18 took 1047600 ticks
Iteration 19 took 1045101 ticks
Changing 20 users email addresses, twice each
```

```
Iteration 0 on user 0 took 2144929 ticks
Iteration 0 on user 1 took 2309197 ticks
Iteration 0 on user 2 took 2296804 ticks
Iteration 0 on user 3 took 2303324 ticks
Iteration 0 on user 4 took 2179032 ticks
Iteration 0 on user 5 took 2188025 ticks
Iteration 0 on user 6 took 2203000 ticks
Iteration 0 on user 7 took 2179551 ticks
Iteration 0 on user 8 took 2187701 ticks
Iteration 0 on user 9 took 2181731 ticks
Iteration 0 on user 10 took 2159705 ticks
Iteration 0 on user 11 took 2182793 ticks
Iteration 0 on user 12 took 2191284 ticks
Iteration 0 on user 13 took 2191599 ticks
Iteration 0 on user 14 took 2218704 ticks
Iteration 0 on user 15 took 2199685 ticks
Iteration 0 on user 16 took 2177090 ticks
Iteration 0 on user 17 took 2162551 ticks
Iteration 0 on user 18 took 2210758 ticks
Iteration 0 on user 19 took 2238903 ticks
Iteration 1 on user 0 took 2214070 ticks
Iteration 1 on user 1 took 2169453 ticks
Iteration 1 on user 2 took 2219987 ticks
Iteration 1 on user 3 took 2161489 ticks
Iteration 1 on user 4 took 2184964 ticks
Iteration 1 on user 5 took 2178298 ticks
Iteration 1 on user 6 took 2170673 ticks
Iteration 1 on user 7 took 2169208 ticks
Iteration 1 on user 8 took 2184253 ticks
Iteration 1 on user 9 took 2171763 ticks
Iteration 1 on user 10 took 2181439 ticks
Iteration 1 on user 11 took 2172834 ticks
Iteration 1 on user 12 took 2195703 ticks
Iteration 1 on user 13 took 2194453 ticks
Iteration 1 on user 14 took 2198717 ticks
Iteration 1 on user 15 took 2231701 ticks
Iteration 1 on user 16 took 2200611 ticks
Iteration 1 on user 17 took 2164702 ticks
Iteration 1 on user 18 took 2203711 ticks
Iteration 1 on user 19 took 2183038 ticks
Control: Using no encryption
Seeding DB with 100000 users
Saving changes
Adding 30 users individually
Iteration 0 took 753598 ticks
Iteration 1 took 753543 ticks
Iteration 2 took 742104 ticks
Iteration 3 took 752708 ticks
Iteration 4 took 782512 ticks
Iteration 5 took 772195 ticks
Iteration 6 took 782640 ticks
Iteration 7 took 745087 ticks
Iteration 8 took 735500 ticks
Iteration 9 took 737143 ticks
Iteration 10 took 748988 ticks
Iteration 11 took 740849 ticks
Iteration 12 took 732671 ticks
Iteration 13 took 727669 ticks
Iteration 14 took 730234 ticks
Iteration 15 took 735700 ticks
Iteration 16 took 730445 ticks
Iteration 17 took 750628 ticks
Iteration 18 took 744913 ticks
Iteration 19 took 746573 ticks
Iteration 20 took 745048 ticks
Iteration 21 took 738513 ticks
Iteration 22 took 737727 ticks
Iteration 23 took 740350 ticks
Iteration 24 took 737086 ticks
Iteration 25 took 735677 ticks
Iteration 26 took 732710 ticks
Iteration 27 took 734756 ticks
Iteration 28 took 731354 ticks
Iteration 29 took 738809 ticks
Changing 30 users email addresses, twice each
Iteration 0 on user 0 took 738332 ticks
Iteration 0 on user 1 took 725870 ticks
Iteration 0 on user 2 took 732052 ticks
Iteration 0 on user 3 took 731645 ticks
Iteration 0 on user 4 took 751716 ticks
Iteration 0 on user 5 took 746594 ticks
Iteration 0 on user 6 took 739376 ticks
Iteration 0 on user 7 took 726314 ticks
Iteration 0 on user 8 took 731069 ticks
Iteration 0 on user 9 took 751266 ticks
Iteration 0 on user 10 took 812896 ticks
Iteration 0 on user 11 took 732493 ticks
Iteration 0 on user 12 took 732957 ticks
Iteration 0 on user 13 took 773640 ticks
Iteration 0 on user 14 took 726650 ticks
Iteration 0 on user 15 took 734845 ticks
Iteration 0 on user 16 took 760128 ticks
Iteration 0 on user 17 took 741609 ticks
Iteration 0 on user 18 took 737910 ticks
Iteration 0 on user 19 took 746540 ticks
Iteration 0 on user 20 took 739454 ticks
Iteration 0 on user 21 took 731860 ticks
Iteration 0 on user 22 took 739386 ticks
Iteration 0 on user 23 took 736066 ticks
Iteration 0 on user 24 took 717195 ticks
Iteration 0 on user 25 took 734829 ticks
Iteration 0 on user 26 took 757114 ticks
Iteration 0 on user 27 took 734375 ticks
Iteration 0 on user 28 took 744126 ticks
Iteration 0 on user 29 took 759128 ticks
Iteration 1 on user 0 took 738594 ticks
Iteration 1 on user 1 took 729013 ticks
Iteration 1 on user 2 took 737042 ticks
Iteration 1 on user 3 took 743327 ticks
Iteration 1 on user 4 took 742406 ticks
Iteration 1 on user 5 took 749291 ticks
Iteration 1 on user 6 took 734503 ticks
Iteration 1 on user 7 took 733859 ticks
Iteration 1 on user 8 took 740316 ticks
Iteration 1 on user 9 took 745597 ticks
Iteration 1 on user 10 took 765531 ticks
Iteration 1 on user 11 took 724300 ticks
Iteration 1 on user 12 took 729023 ticks
Iteration 1 on user 13 took 741729 ticks
Iteration 1 on user 14 took 729054 ticks
Iteration 1 on user 15 took 737670 ticks
Iteration 1 on user 16 took 740058 ticks
Iteration 1 on user 17 took 735799 ticks
Iteration 1 on user 18 took 732457 ticks
Iteration 1 on user 19 took 743522 ticks
Iteration 1 on user 20 took 728927 ticks
Iteration 1 on user 21 took 757127 ticks
Iteration 1 on user 22 took 740264 ticks
Iteration 1 on user 23 took 720255 ticks
Iteration 1 on user 24 took 758281 ticks
Iteration 1 on user 25 took 743671 ticks
Iteration 1 on user 26 took 741371 ticks
Iteration 1 on user 27 took 757789 ticks
Iteration 1 on user 28 took 734187 ticks
Iteration 1 on user 29 took 720952 ticks
Control: using a fixed number of buckets
Seeding DB with 100000 users
Saving changes
Adding 30 users individually
Iteration 0 took 924464 ticks
Iteration 1 took 895056 ticks
Iteration 2 took 917525 ticks
Iteration 3 took 912955 ticks
Iteration 4 took 929148 ticks
Iteration 5 took 931094 ticks
Iteration 6 took 936211 ticks
Iteration 7 took 909118 ticks
Iteration 8 took 955261 ticks
Iteration 9 took 954409 ticks
Iteration 10 took 979415 ticks
Iteration 11 took 955204 ticks
Iteration 12 took 934555 ticks
Iteration 13 took 914798 ticks
Iteration 14 took 910550 ticks
Iteration 15 took 888210 ticks
Iteration 16 took 905585 ticks
Iteration 17 took 955149 ticks
Iteration 18 took 935567 ticks
Iteration 19 took 944464 ticks
Iteration 20 took 931648 ticks
Iteration 21 took 922432 ticks
Iteration 22 took 922763 ticks
Iteration 23 took 909274 ticks
Iteration 24 took 913560 ticks
Iteration 25 took 904939 ticks
Iteration 26 took 913065 ticks
Iteration 27 took 922020 ticks
Iteration 28 took 918839 ticks
Iteration 29 took 911000 ticks
Changing 30 users email addresses, twice each
Iteration 0 on user 0 took 1710054 ticks
Iteration 0 on user 1 took 1735561 ticks
Iteration 0 on user 2 took 1715647 ticks
Iteration 0 on user 3 took 1712950 ticks
Iteration 0 on user 4 took 2511633 ticks
Iteration 0 on user 5 took 1677111 ticks
Iteration 0 on user 6 took 1697136 ticks
Iteration 0 on user 7 took 1675289 ticks
Iteration 0 on user 8 took 1685916 ticks
Iteration 0 on user 9 took 1669528 ticks
Iteration 0 on user 10 took 1676084 ticks
Iteration 0 on user 11 took 1701327 ticks
Iteration 0 on user 12 took 1701339 ticks
Iteration 0 on user 13 took 1711773 ticks
Iteration 0 on user 14 took 1668283 ticks
Iteration 0 on user 15 took 1641764 ticks
Iteration 0 on user 16 took 1664441 ticks
Iteration 0 on user 17 took 1722602 ticks
Iteration 0 on user 18 took 1675973 ticks
Iteration 0 on user 19 took 1675808 ticks
Iteration 0 on user 20 took 1659350 ticks
Iteration 0 on user 21 took 1656883 ticks
Iteration 0 on user 22 took 1660153 ticks
Iteration 0 on user 23 took 1697602 ticks
Iteration 0 on user 24 took 1666857 ticks
Iteration 0 on user 25 took 1713791 ticks
Iteration 0 on user 26 took 1638086 ticks
Iteration 0 on user 27 took 1629296 ticks
Iteration 0 on user 28 took 1645768 ticks
Iteration 0 on user 29 took 1720645 ticks
Iteration 1 on user 0 took 1758793 ticks
Iteration 1 on user 1 took 1688701 ticks
Iteration 1 on user 2 took 2354238 ticks
Iteration 1 on user 3 took 1655311 ticks
Iteration 1 on user 4 took 1690476 ticks
Iteration 1 on user 5 took 1668810 ticks
Iteration 1 on user 6 took 1675522 ticks
Iteration 1 on user 7 took 1654547 ticks
Iteration 1 on user 8 took 1663325 ticks
Iteration 1 on user 9 took 1672792 ticks
Iteration 1 on user 10 took 1703563 ticks
Iteration 1 on user 11 took 1680682 ticks
Iteration 1 on user 12 took 1664823 ticks
Iteration 1 on user 13 took 1689154 ticks
Iteration 1 on user 14 took 1647602 ticks
Iteration 1 on user 15 took 1653459 ticks
Iteration 1 on user 16 took 1738939 ticks
Iteration 1 on user 17 took 1682121 ticks
Iteration 1 on user 18 took 1682847 ticks
Iteration 1 on user 19 took 1685918 ticks
Iteration 1 on user 20 took 1653328 ticks
Iteration 1 on user 21 took 1681807 ticks
Iteration 1 on user 22 took 1689056 ticks
Iteration 1 on user 23 took 1667132 ticks
Iteration 1 on user 24 took 1649850 ticks
Iteration 1 on user 25 took 1653464 ticks
Iteration 1 on user 26 took 1682669 ticks
Iteration 1 on user 27 took 1680465 ticks
Iteration 1 on user 28 took 1687412 ticks
Iteration 1 on user 29 took 1668305 ticks
Using ‘SELECT COUNT(*)‘ for updates to BucketCount, storing BucketExponent in the entity
Seeding DB with 100000 users
Saving changes
Adding 30 users individually
Iteration 0 took 1489776 ticks
Iteration 1 took 1472937 ticks
Iteration 2 took 1556628 ticks
Iteration 3 took 1474189 ticks
Iteration 4 took 1389850 ticks
Iteration 5 took 1382840 ticks
Iteration 6 took 1375135 ticks
Iteration 7 took 1334261 ticks
Iteration 8 took 1311174 ticks
Iteration 9 took 1427842 ticks
Iteration 10 took 1350221 ticks
Iteration 11 took 1425256 ticks
Iteration 12 took 1359196 ticks
Iteration 13 took 1397492 ticks
Iteration 14 took 1432199 ticks
Iteration 15 took 1471673 ticks
Iteration 16 took 2071172 ticks
Iteration 17 took 1577383 ticks
Iteration 18 took 1500116 ticks
Iteration 19 took 1592602 ticks
Iteration 20 took 1732489 ticks
Iteration 21 took 1771209 ticks
Iteration 22 took 1543665 ticks
Iteration 23 took 1610736 ticks
Iteration 24 took 1619616 ticks
Iteration 25 took 1628815 ticks
Iteration 26 took 1641886 ticks
Iteration 27 took 1713335 ticks
Iteration 28 took 1623625 ticks
Iteration 29 took 1580985 ticks
Changing 30 users email addresses, twice each
Iteration 0 on user 0 took 1594655 ticks
Iteration 0 on user 1 took 1638083 ticks
Iteration 0 on user 2 took 1612916 ticks
Iteration 0 on user 3 took 1556072 ticks
Iteration 0 on user 4 took 1871284 ticks
Iteration 0 on user 5 took 1932592 ticks
Iteration 0 on user 6 took 1858299 ticks
Iteration 0 on user 7 took 1772382 ticks
Iteration 0 on user 8 took 1865216 ticks
Iteration 0 on user 9 took 1549499 ticks
Iteration 0 on user 10 took 5237904 ticks
Iteration 0 on user 11 took 1884428 ticks
Iteration 0 on user 12 took 1816770 ticks
Iteration 0 on user 13 took 1600187 ticks
Iteration 0 on user 14 took 1804694 ticks
Iteration 0 on user 15 took 1617393 ticks
Iteration 0 on user 16 took 1864536 ticks
Iteration 0 on user 17 took 1868598 ticks
Iteration 0 on user 18 took 2067933 ticks
Iteration 0 on user 19 took 2146011 ticks
Iteration 0 on user 20 took 1722819 ticks
Iteration 0 on user 21 took 1859018 ticks
Iteration 0 on user 22 took 1707285 ticks
Iteration 0 on user 23 took 1934553 ticks
Iteration 0 on user 24 took 1685409 ticks
Iteration 0 on user 25 took 1739293 ticks
Iteration 0 on user 26 took 1852793 ticks
Iteration 0 on user 27 took 2024768 ticks
Iteration 0 on user 28 took 1788091 ticks
Iteration 0 on user 29 took 1717121 ticks
Iteration 1 on user 0 took 1531988 ticks
Iteration 1 on user 1 took 1919067 ticks
Iteration 1 on user 2 took 1631652 ticks
Iteration 1 on user 3 took 2010491 ticks
Iteration 1 on user 4 took 1679093 ticks
Iteration 1 on user 5 took 1683430 ticks
Iteration 1 on user 6 took 1613296 ticks
Iteration 1 on user 7 took 1864378 ticks
Iteration 1 on user 8 took 1947435 ticks
Iteration 1 on user 9 took 1994076 ticks
Iteration 1 on user 10 took 1615929 ticks
Iteration 1 on user 11 took 1885441 ticks
Iteration 1 on user 12 took 1729058 ticks
Iteration 1 on user 13 took 1647126 ticks
Iteration 1 on user 14 took 1847486 ticks
Iteration 1 on user 15 took 1643260 ticks
Iteration 1 on user 16 took 1574984 ticks
Iteration 1 on user 17 took 1534604 ticks
Iteration 1 on user 18 took 2399829 ticks
Iteration 1 on user 19 took 1640916 ticks
Iteration 1 on user 20 took 1836068 ticks
Iteration 1 on user 21 took 1621930 ticks
Iteration 1 on user 22 took 1630611 ticks
Iteration 1 on user 23 took 2019676 ticks
Iteration 1 on user 24 took 4653954 ticks
Iteration 1 on user 25 took 1868582 ticks
Iteration 1 on user 26 took 1612807 ticks
Iteration 1 on user 27 took 1866078 ticks
Iteration 1 on user 28 took 1681153 ticks
Iteration 1 on user 29 took 1912387 ticks
Using a summary table for updates to BucketCount, storing BucketExponent in the entity
Seeding DB with 100000 users
Saving changes
Adding 30 users individually
Iteration 0 took 1408373 ticks
Iteration 1 took 1297226 ticks
Iteration 2 took 1351080 ticks
Iteration 3 took 1335185 ticks
Iteration 4 took 1390386 ticks
Iteration 5 took 1319566 ticks
Iteration 6 took 1359053 ticks
Iteration 7 took 1347666 ticks
Iteration 8 took 1265146 ticks
Iteration 9 took 1278275 ticks
Iteration 10 took 1422274 ticks
Iteration 11 took 1373832 ticks
Iteration 12 took 1279522 ticks
Iteration 13 took 1337924 ticks
Iteration 14 took 1400896 ticks
Iteration 15 took 1355653 ticks
Iteration 16 took 1330499 ticks
Iteration 17 took 1359149 ticks
Iteration 18 took 2368564 ticks
Iteration 19 took 1420846 ticks
Iteration 20 took 1372980 ticks
Iteration 21 took 1379547 ticks
Iteration 22 took 1400825 ticks
Iteration 23 took 1404188 ticks
Iteration 24 took 1399948 ticks
Iteration 25 took 1465778 ticks
Iteration 26 took 1459342 ticks
Iteration 27 took 1510328 ticks
Iteration 28 took 1544787 ticks
Iteration 29 took 1562740 ticks
Changing 30 users email addresses, twice each
Iteration 0 on user 0 took 1463472 ticks
Iteration 0 on user 1 took 1442528 ticks
Iteration 0 on user 2 took 1447625 ticks
Iteration 0 on user 3 took 1516709 ticks
Iteration 0 on user 4 took 1493016 ticks
Iteration 0 on user 5 took 1453886 ticks
Iteration 0 on user 6 took 2541671 ticks
Iteration 0 on user 7 took 1787033 ticks
Iteration 0 on user 8 took 1778028 ticks
Iteration 0 on user 9 took 1453422 ticks
Iteration 0 on user 10 took 1679111 ticks
Iteration 0 on user 11 took 1730107 ticks
Iteration 0 on user 12 took 1402527 ticks
Iteration 0 on user 13 took 1453013 ticks
Iteration 0 on user 14 took 1472997 ticks
Iteration 0 on user 15 took 1690043 ticks
Iteration 0 on user 16 took 1566324 ticks
Iteration 0 on user 17 took 1744514 ticks
Iteration 0 on user 18 took 1443954 ticks
Iteration 0 on user 19 took 1566340 ticks
Iteration 0 on user 20 took 1742121 ticks
Iteration 0 on user 21 took 1438111 ticks
Iteration 0 on user 22 took 1694713 ticks
Iteration 0 on user 23 took 1463847 ticks
Iteration 0 on user 24 took 1460439 ticks
Iteration 0 on user 25 took 1555753 ticks
Iteration 0 on user 26 took 1500066 ticks
Iteration 0 on user 27 took 1722642 ticks
Iteration 0 on user 28 took 1457617 ticks
Iteration 0 on user 29 took 1478153 ticks
Iteration 1 on user 0 took 1728255 ticks
Iteration 1 on user 1 took 1764597 ticks
Iteration 1 on user 2 took 1556861 ticks
Iteration 1 on user 3 took 1762690 ticks
Iteration 1 on user 4 took 1478272 ticks
Iteration 1 on user 5 took 1692745 ticks
Iteration 1 on user 6 took 1539895 ticks
Iteration 1 on user 7 took 1469426 ticks
Iteration 1 on user 8 took 1484144 ticks
Iteration 1 on user 9 took 1533086 ticks
Iteration 1 on user 10 took 1753756 ticks
Iteration 1 on user 11 took 1500008 ticks
Iteration 1 on user 12 took 1528944 ticks
Iteration 1 on user 13 took 1811514 ticks
Iteration 1 on user 14 took 1433951 ticks
Iteration 1 on user 15 took 1667939 ticks
Iteration 1 on user 16 took 1426034 ticks
Iteration 1 on user 17 took 1743091 ticks
Iteration 1 on user 18 took 2575834 ticks
Iteration 1 on user 19 took 1710230 ticks
Iteration 1 on user 20 took 1691855 ticks
Iteration 1 on user 21 took 1520937 ticks
Iteration 1 on user 22 took 1765171 ticks
Iteration 1 on user 23 took 1502993 ticks
Iteration 1 on user 24 took 1450929 ticks
Iteration 1 on user 25 took 1759924 ticks
Iteration 1 on user 26 took 1487496 ticks
Iteration 1 on user 27 took 1461432 ticks
Iteration 1 on user 28 took 1491853 ticks
Iteration 1 on user 29 took 1521915 ticks
Using autoincremented row numbers
Seeding DB with 100000 users
Saving changes
Adding 30 users individually
Iteration 0 took 2093130 ticks
Iteration 1 took 1916367 ticks
Iteration 2 took 1954949 ticks
Iteration 3 took 1928748 ticks
Iteration 4 took 2050358 ticks
Iteration 5 took 1876941 ticks
Iteration 6 took 1947054 ticks
Iteration 7 took 1743159 ticks
Iteration 8 took 1636192 ticks
Iteration 9 took 1988000 ticks
Iteration 10 took 1850727 ticks
Iteration 11 took 1873752 ticks
Iteration 12 took 2053476 ticks
Iteration 13 took 2031619 ticks
Iteration 14 took 2055540 ticks
Iteration 15 took 2184530 ticks
Iteration 16 took 1934960 ticks
Iteration 17 took 2074536 ticks
Iteration 18 took 2151211 ticks
Iteration 19 took 1915864 ticks
Iteration 20 took 2139624 ticks
Iteration 21 took 1873169 ticks
Iteration 22 took 2105540 ticks
Iteration 23 took 2130578 ticks
Iteration 24 took 1965231 ticks
Iteration 25 took 2168810 ticks
Iteration 26 took 2023732 ticks
Iteration 27 took 2037724 ticks
Iteration 28 took 2157464 ticks
Iteration 29 took 2289360 ticks
Changing 30 users email addresses, twice each
Iteration 0 on user 0 took 1990667 ticks
Iteration 0 on user 1 took 2290808 ticks
Iteration 0 on user 2 took 1897982 ticks
Iteration 0 on user 3 took 2205677 ticks
Iteration 0 on user 4 took 2020622 ticks
Iteration 0 on user 5 took 2177581 ticks
Iteration 0 on user 6 took 2211996 ticks
Iteration 0 on user 7 took 1914300 ticks
Iteration 0 on user 8 took 1886302 ticks
Iteration 0 on user 9 took 2014267 ticks
Iteration 0 on user 10 took 2168504 ticks
Iteration 0 on user 11 took 1934716 ticks
Iteration 0 on user 12 took 1887977 ticks
Iteration 0 on user 13 took 1912724 ticks
Iteration 0 on user 14 took 2026573 ticks
Iteration 0 on user 15 took 2118973 ticks
Iteration 0 on user 16 took 2128150 ticks
Iteration 0 on user 17 took 2089086 ticks
Iteration 0 on user 18 took 2165776 ticks
Iteration 0 on user 19 took 2057729 ticks
Iteration 0 on user 20 took 1915350 ticks
Iteration 0 on user 21 took 2180901 ticks
Iteration 0 on user 22 took 1894561 ticks
Iteration 0 on user 23 took 1873168 ticks
Iteration 0 on user 24 took 2052633 ticks
Iteration 0 on user 25 took 1899084 ticks
Iteration 0 on user 26 took 2078164 ticks
Iteration 0 on user 27 took 2101185 ticks
Iteration 0 on user 28 took 1827278 ticks
Iteration 0 on user 29 took 2158606 ticks
Iteration 1 on user 0 took 2162381 ticks
Iteration 1 on user 1 took 1811418 ticks
Iteration 1 on user 2 took 1895089 ticks
Iteration 1 on user 3 took 1914206 ticks
Iteration 1 on user 4 took 2104982 ticks
Iteration 1 on user 5 took 2164760 ticks
Iteration 1 on user 6 took 1916879 ticks
Iteration 1 on user 7 took 2237540 ticks
Iteration 1 on user 8 took 2243699 ticks
Iteration 1 on user 9 took 1888181 ticks
Iteration 1 on user 10 took 2150566 ticks
Iteration 1 on user 11 took 2127797 ticks
Iteration 1 on user 12 took 2151895 ticks
Iteration 1 on user 13 took 2001274 ticks
Iteration 1 on user 14 took 1870205 ticks
Iteration 1 on user 15 took 2102912 ticks
Iteration 1 on user 16 took 1976520 ticks
Iteration 1 on user 17 took 1919304 ticks
Iteration 1 on user 18 took 1994850 ticks
Iteration 1 on user 19 took 2089044 ticks
Iteration 1 on user 20 took 2198788 ticks
Iteration 1 on user 21 took 1941740 ticks
Iteration 1 on user 22 took 2160928 ticks
Iteration 1 on user 23 took 2006290 ticks
Iteration 1 on user 24 took 2131194 ticks
Iteration 1 on user 25 took 2174508 ticks
Iteration 1 on user 26 took 1911322 ticks
Iteration 1 on user 27 took 2154827 ticks
Iteration 1 on user 28 took 2008714 ticks
Iteration 1 on user 29 took 2147777 ticks
Using ‘SELECT COUNT(*)‘ for updates to BucketCount, updating all bucket numbers after threshold is passe
Seeding DB with 100000 users
Saving changes
Adding 30 users individually
Iteration 0 took 1078759 ticks
Iteration 1 took 1052468 ticks
Iteration 2 took 1043939 ticks
Iteration 3 took 1077651 ticks
Iteration 4 took 1805152 ticks
Iteration 5 took 1072182 ticks
Iteration 6 took 1062384 ticks
Iteration 7 took 1060844 ticks
Iteration 8 took 1095828 ticks
Iteration 9 took 1095053 ticks
Iteration 10 took 1058192 ticks
Iteration 11 took 1071309 ticks
Iteration 12 took 1050967 ticks
Iteration 13 took 1069327 ticks
Iteration 14 took 1071193 ticks
Iteration 15 took 1061586 ticks
Iteration 16 took 1054072 ticks
Iteration 17 took 1057974 ticks
Iteration 18 took 1055528 ticks
Iteration 19 took 1059504 ticks
Iteration 20 took 1060441 ticks
Iteration 21 took 1057394 ticks
Iteration 22 took 1047332 ticks
Iteration 23 took 1052943 ticks
Iteration 24 took 1054839 ticks
Iteration 25 took 1056737 ticks
Iteration 26 took 1049124 ticks
Iteration 27 took 1062874 ticks
Iteration 28 took 1056950 ticks
Iteration 29 took 1055686 ticks
Changing 30 users email addresses, twice each
Iteration 0 on user 0 took 2256260 ticks
Iteration 0 on user 1 took 2228976 ticks
Iteration 0 on user 2 took 2220598 ticks
Iteration 0 on user 3 took 2204240 ticks
Iteration 0 on user 4 took 2189663 ticks
Iteration 0 on user 5 took 2195124 ticks
Iteration 0 on user 6 took 2228184 ticks
Iteration 0 on user 7 took 2205997 ticks
Iteration 0 on user 8 took 2173301 ticks
Iteration 0 on user 9 took 2155479 ticks
Iteration 0 on user 10 took 2179225 ticks
Iteration 0 on user 11 took 2186297 ticks
Iteration 0 on user 12 took 2249509 ticks
Iteration 0 on user 13 took 2197445 ticks
Iteration 0 on user 14 took 2187278 ticks
Iteration 0 on user 15 took 2168610 ticks
Iteration 0 on user 16 took 2204713 ticks
Iteration 0 on user 17 took 2234718 ticks
Iteration 0 on user 18 took 2216632 ticks
Iteration 0 on user 19 took 2177476 ticks
Iteration 0 on user 20 took 2182666 ticks
Iteration 0 on user 21 took 2217082 ticks
Iteration 0 on user 22 took 2182768 ticks
Iteration 0 on user 23 took 2197642 ticks
Iteration 0 on user 24 took 2240068 ticks
Iteration 0 on user 25 took 2201361 ticks
Iteration 0 on user 26 took 2204914 ticks
Iteration 0 on user 27 took 2192197 ticks
Iteration 0 on user 28 took 2213693 ticks
Iteration 0 on user 29 took 2196160 ticks
Iteration 1 on user 0 took 2182609 ticks
Iteration 1 on user 1 took 2175147 ticks
Iteration 1 on user 2 took 2207078 ticks
Iteration 1 on user 3 took 2243453 ticks
Iteration 1 on user 4 took 2189024 ticks
Iteration 1 on user 5 took 2279092 ticks
Iteration 1 on user 6 took 2188022 ticks
Iteration 1 on user 7 took 2174098 ticks
Iteration 1 on user 8 took 2202955 ticks
Iteration 1 on user 9 took 2190477 ticks
Iteration 1 on user 10 took 2187136 ticks
Iteration 1 on user 11 took 2204939 ticks
Iteration 1 on user 12 took 2208682 ticks
Iteration 1 on user 13 took 2207738 ticks
Iteration 1 on user 14 took 2205932 ticks
Iteration 1 on user 15 took 2192300 ticks
Iteration 1 on user 16 took 2181315 ticks
Iteration 1 on user 17 took 2234649 ticks
Iteration 1 on user 18 took 2188282 ticks
Iteration 1 on user 19 took 2206907 ticks
Iteration 1 on user 20 took 2185559 ticks
Iteration 1 on user 21 took 2211374 ticks
Iteration 1 on user 22 took 2216891 ticks
Iteration 1 on user 23 took 2168575 ticks
Iteration 1 on user 24 took 2190485 ticks
Iteration 1 on user 25 took 2163339 ticks
Iteration 1 on user 26 took 2234382 ticks
Iteration 1 on user 27 took 2202092 ticks
Iteration 1 on user 28 took 2191940 ticks
Iteration 1 on user 29 took 2189683 ticks
Control: Using no encryption
Seeding DB with 1000000 users
Saving changes
Adding 10 users individually
Iteration 0 took 6554289 ticks
Iteration 1 took 6560005 ticks
Iteration 2 took 6446588 ticks
Iteration 3 took 6545852 ticks
Iteration 4 took 6498223 ticks
Iteration 5 took 6813132 ticks
Iteration 6 took 6935391 ticks
Iteration 7 took 6805422 ticks
Iteration 8 took 7070927 ticks
Iteration 9 took 7167681 ticks
Changing 10 users email addresses, twice each
Iteration 0 on user 0 took 6749517 ticks
Iteration 0 on user 1 took 6713827 ticks
Iteration 0 on user 2 took 6675996 ticks
Iteration 0 on user 3 took 6759640 ticks
Iteration 0 on user 4 took 6712013 ticks
Iteration 0 on user 5 took 6687223 ticks
Iteration 0 on user 6 took 6733278 ticks
Iteration 0 on user 7 took 6774619 ticks
Iteration 0 on user 8 took 6731432 ticks
Iteration 0 on user 9 took 6748156 ticks
Iteration 1 on user 0 took 6733058 ticks
Iteration 1 on user 1 took 6694947 ticks
Iteration 1 on user 2 took 6810708 ticks
Iteration 1 on user 3 took 6847964 ticks
Iteration 1 on user 4 took 6681679 ticks
Iteration 1 on user 5 took 6729429 ticks
Iteration 1 on user 6 took 6728874 ticks
Iteration 1 on user 7 took 6681205 ticks
Iteration 1 on user 8 took 6746839 ticks
Iteration 1 on user 9 took 6701213 ticks
Control: using a fixed number of buckets
Seeding DB with 1000000 users
Saving changes
```

```
Adding 10 users individually
Iteration 0 took 117401279 ticks
Iteration 1 took 8521144 ticks
Iteration 2 took 8714692 ticks
Iteration 3 took 15245268 ticks
Iteration 4 took 8084167 ticks
Iteration 5 took 11658380 ticks
Iteration 6 took 8185242 ticks
Iteration 7 took 11495766 ticks
Iteration 8 took 8206163 ticks
Iteration 9 took 8156571 ticks
Changing 10 users email addresses, twice each
Iteration 0 on user 0 took 15702852 ticks
Iteration 0 on user 1 took 15477538 ticks
Iteration 0 on user 2 took 15455403 ticks
Iteration 0 on user 3 took 15534068 ticks
Iteration 0 on user 4 took 15330509 ticks
Iteration 0 on user 5 took 17242727 ticks
Iteration 0 on user 6 took 15648815 ticks
Iteration 0 on user 7 took 18636322 ticks
Iteration 0 on user 8 took 15504049 ticks
Iteration 0 on user 9 took 15278504 ticks
Iteration 1 on user 0 took 15901987 ticks
Iteration 1 on user 1 took 15405930 ticks
Iteration 1 on user 2 took 15871552 ticks
Iteration 1 on user 3 took 15512230 ticks
Iteration 1 on user 4 took 15576287 ticks
Iteration 1 on user 5 took 15943582 ticks
Iteration 1 on user 6 took 15286869 ticks
Iteration 1 on user 7 took 16225009 ticks
Iteration 1 on user 8 took 15492694 ticks
Iteration 1 on user 9 took 15976515 ticks
Using ‘SELECT COUNT(*)‘ for updates to BucketCount, storing BucketExponent in the entity
Seeding DB with 1000000 users
Saving changes
Adding 10 users individually
Iteration 0 took 76707835 ticks
Iteration 1 took 25435896 ticks
Iteration 2 took 24051210 ticks
Iteration 3 took 14730130 ticks
Iteration 4 took 13358141 ticks
Iteration 5 took 15089452 ticks
Iteration 6 took 14195089 ticks
Iteration 7 took 14664764 ticks
Iteration 8 took 13501334 ticks
Iteration 9 took 13234464 ticks
Changing 10 users email addresses, twice each
Iteration 0 on user 0 took 12431911 ticks
Iteration 0 on user 1 took 14511629 ticks
Iteration 0 on user 2 took 13474398 ticks
Iteration 0 on user 3 took 12803638 ticks
Iteration 0 on user 4 took 13352765 ticks
Iteration 0 on user 5 took 10361611 ticks
Iteration 0 on user 6 took 12737059 ticks
Iteration 0 on user 7 took 11041729 ticks
Iteration 0 on user 8 took 12916139 ticks
Iteration 0 on user 9 took 13043249 ticks
Iteration 1 on user 0 took 13146210 ticks
Iteration 1 on user 1 took 16647301 ticks
Iteration 1 on user 2 took 13333598 ticks
Iteration 1 on user 3 took 13156157 ticks
Iteration 1 on user 4 took 13328786 ticks
Iteration 1 on user 5 took 13282441 ticks
Iteration 1 on user 6 took 10397142 ticks
Iteration 1 on user 7 took 13270803 ticks
Iteration 1 on user 8 took 13256051 ticks
Iteration 1 on user 9 took 12982361 ticks
```