



THE APPLICATION PLATFORM FOR SUPERFAST AND LARGE REAL-TIME SOLUTIONS

Today's application users are sophisticated and demanding with high expectations on features and performance. With the continuous evolution of the Internet and the development of innovative web applications we see large numbers of simultaneous users to online applications resulting in huge volume of data.

Starcounter is a superfast combined in-memory database engine and application server for ultra fast development of real-time business applications. Starcounter will radically change what solutions you can develop. We provide a fully featured and proven platform that enables you to:

- Run mission critical applications at least a 100 times faster. Optimal for real-time high transactional data.
- Radically reduce time and cost for development of applications.
- Makes development of critical applications possible/easier.
- Enable next generation services.
- Always have true ACID compliant transactions.
- Radcially reduce the number of needed servers.

System requirements

- **Multi-core CPU.** Intel and AMD CPUs with at least two logical cores are supported. For Intel processors, the Nehalem micro-architecture or later is supported. For AMD processors, the Barcelona micro-architecture or later is supported. This typically means that Intel computers prior to 2009 and AMD computers prior to 2008 will not be able to run Starcounter.
- **Microsoft Windows 64-bit operating system** (e.g. Windows 8 x64, Windows 7 x64, Windows Server 2008 x64, Windows Server 2012 x64)
- .NET Framework 4.5, 4.0, 3.5
- Visual Studio 2015, 2013, 2012 (for development)

SUPER FAST & SUPER EASY

Super fast

Starcounter include an in-memory database than is more than 100 times faster than traditional databases. It means performing millions of transactions per second on a standard server.

In the market today, with expectations on many applications for an always-on experience and with accurate data instantly available, requirements on database performance is huge. Retail chains with thousands of decentralized stores, demanding information on the same goods in these different locations, are in need for faster platforms to be able to provide all in one system. Internet applications supporting huge numbers of simultaneous users, wants both to be fast in response times and provide accurate information.

How do we handle performance issues today? The number one choice is to cluster to improve performance. Using two servers instead of one you gain some performance, using 100 servers instead of one; you gain even bigger in performance. To get performance over all nodes in the cluster you need to give up consistency of your data. In a business critical application this is not an option. We see more and more examples of large server plants, great in performance, but astronomic in costs of hardware, software licenses, cooling, labor costs to maintain all of this and probably most of all the power costs. With an extremely fast database like Starcounter this could change.

Being 100 times faster means you theoretically could reduce you server park

with a factor 100. Starcounter's extreme speed is the result of new innovations in combination of well chosen foundation pieces.

- Being in-memory increase the performance by a factor of 10.
- Starcounter has a patent invention called VMDBMS that also gives approximately a factor 10 in speed gain
- Starcounter is optimized for modern hardware with scalable parallel processing, and to optimize the use of CPU caches.

With this extreme performance at your hands, you will be able to ease the performance pains of today, while reducing costs. Looking into future applications, this is the chance to use your imagination and start developing applications as you would like to, without considering performance issues.

Super easy

Developing an application in Starcounter can be done reducing the lines of code with at least a third.

The number of lines of codes can be used to determine development cost for an application and also maintenance costs of an application. With the majority of modern applications being object-oriented and the majority of enterprise databases structured on relational models, the technical difficulty to combine them is a well-known problem. The developer cannot just focus on developing a cutting edge application but also needs to implement code for sending data back and forth between the application

and the data storage. This creates an overlay of code; that besides taking time from the developer, also is a potential source of programming errors. With Starcounter this step will be eliminated.

Starcounter has an object oriented interface, which makes it super easy for the developer.

- The database model in Starcounter is the database schema. All lines of code for the database schema are eliminated.
- As all data is stored as objects, the lines of codes to translate between object-oriented and relational models are eliminated.

As Starcounter has a patented innovation integrating the application and the database, it also makes it easier for the developer. The developer uses a native .NET API to access all features of the database.

Being able to reduce the lines of code with this order of magnitude will make the developers more productive. With higher productivity the time to market for a new product will be reduced. Less lines of code creates less reasons for errors and less to review, which will lower the overall maintenance costs. And for the developer himself, hopefully this can be connected to the future opportunities mentioned above, a less complex environment will give him time to focus his creativity for future applications.

CORE FUNCTIONALITY

General purpose application platform

Starcounter is a general purpose application platform that combines an in-

memory transactional database and an application server. As it is extremely fast, it will make most difference in applications handling live data for many simultaneous users. It could be used within any vertical, but likely within industries like bank/finance, retail, internet, telecom and gambling/gaming.

ACID compliant

As with traditional databases, such as Oracle, Starcounter's database is fully ACID compliant. ACID (Atomicity, Consistency, Isolation, Durability) is a set of properties that guarantee database transactions are processed reliably.

For example, a transfer of funds from one bank account to another, even though that might involve multiple changes (such as debiting one account and crediting another), can be within a single transaction. If anything goes wrong during one such transaction, all will be rolled back.

Starcounter's database support transactions, checkpoints, recovery, secured disk images and logs, so your data is always safe.

In-memory - secured on disk

Starcounter include an in-memory database, meaning that it operates and stores all data in RAM. As RAM is more than 100 000 times faster than disks, Starcounter is extremely fast in accessing stored information. To be ACIC compliant however, Starcounter secures data and logs of the database on disk.

Object oriented interface

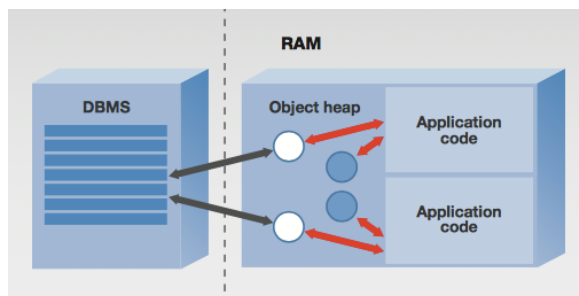
Starcounter has an object oriented interface. This makes the situation for the developer so much easier. The database

model in Starcounter is the database schema. This reduces the number of lines of codes with at least 30%. As development and maintenance costs are in direct relation to the number of lines of codes produced, this is saving money big time.

SQL support

When many alternatives to traditional databases emerge, we see proprietary solutions on how to access the data. At Starcounter we believe that if there is a standard language functional and well known, we must use it too. Hence we have SQL support on top of your code, following the ANSI SQL-92 standard. You can use SQL queries to access data in the database. The SQL can even access the properties and functions of your .NET objects.

New invention: vmDbms



Being memory centric open up opportunities in communication between the application and data. Starcounter has a patented innovation called VMDBMS. In traditional databases, the data and the application are stored in different parts of the RAM. In Starcounter we have integrated the application with the DBMS, making it possible for both the application and the DBMS to share the objects. With this patented innovation, there is no need to move data between different places in

RAM, and you do not need to transform information between different formats.

The VMDBMS is one of the core reasons to the outstanding speed of Starcounter.

```
public class Person : Entity {
    public string Name;
    public IEnumerable Quotes() {
        return SQL("SELECT q FROM Quotes q WHERE q.Who = ?",
            this);
    }
}
```

SUPER EASY .NET API

Compare Starcounter and traditional relational development

A developer using Starcounter should focus on making great object oriented implementations. Our easy to use .NET API let you develop your application and model the database at the same time. The classes you develop will automatically become the database schema, the fields of the classes will be stored and you can access properties and methods from a SQL interface.

Create persistent classes

As a developer all you have to do to create a persistent class is to derive from Starcounter.Entity class. This will tell Starcounter to use this class as a database schema and load the class into the persistent storage when the application is started. The class will correspond to what you normally relate to as a table in relational databases. All instances of this class would traditionally be rows in that table. Starcounter has an object oriented database and will store the information as objects in RAM shared with the application.

```
using Starcounter;
public class Person : Entity {
public String Name; }
```

Instantiate

Starcounter let you use the native constructors of the classes to create new objects.

```
Person p = new Person();
```

modify

You modify the values of the objects using the regular assignment operators.

```
Person p = new Person(); p.Name =
"Albert Einstein";
```

Query objects

Query the created objects using standard SQL expressions. The enumerator return by the query are the actual .NET objects. You can also query the database using path

expressions (person.City.Name) to increase performance and simplify join expressions.

```
using Starcounter;
public class Person : Entity {
    public String Name;
    public IEnumerable Quotes(){
        return SQL("SELECT q
FROM Quotes q WHERE q.Who=?, this);
    }
```

Transactions

One thing you as a developer need to consider, besides creating regular object oriented models, are the transactions. Decide what data should be kept together during modification of information. Starcounter supports transaction scopes, long running transactions, nested transactions and parallel transactions.

Compare Starcounter API to SQL

Starcounter	SQL
class definition	CREATE TABLE
new() operator	INSERT
= operator	UPDATE
SQL(String)	SELECT
Delete () method	DELETE

Starcounter AB, Nybrogatan 15, 114 39 Stockholm, sales@starcounter.com, 08-410 282 10, www.starcounter.com