

[www.nbitsearch.com](http://www.nbitsearch.com)

Novosib-BIT LLC

# Indexing and Fast *Search*

engine

***NBITSearch***

parameters

***version 1.03.3***



# NBITSearch System

## *NBITSearch*

is a search engine with an open API.

-----

*NBITSearch* is a programme kernel for

- Database Management Systems,
- Warehouses of Large Data,
- Search Systems applied to any Objects.

# The System is Designed for

***Compact*** indexing of ***huge***  
arrays of data on a hard disk

for

***high-speed*** exact and fuzzy search for  
objects with ***minimum*** use of RAM.

# Exact and Fuzzy Search

**Interval queries** provide

**fuzzy** (inexact) search.

**Precise** (exact) search  
is a particular case of **fuzzy** search.

# Indexable Objects

Objects **S**  
of any  
types **T**

# Multifunctionality

The system indexes  
objects **S**  
of any types **T**  
simultaneously  
by a set  
any functions **F (S)**.

# Sizes of Indexable Arrays

The most tangible effect in the speed of search is shown for such arrays of objects, which support  **$\approx 50 \div 100$  million and more objects** for one index.

A size of arrays of indexable objects can be  **$10 \div 100$  terabyte and larger.**

# Indexing Limitations

One index supports  
**≈ 2 billion of**  
its objects.

**Limitations**  
of number of indexes are artificial.



# What is a Billion?

**1 billion seconds is  
≈ 32 years.**

**1 billion pages  
for a laser printer is  
a pile with a height of ≈ 100 km.**

# Indexing Speed

Estimator:

$$T \sim (N) * LOG (N)$$

**T** – time of forming one index,  
**N** – number of indexable objects.

# Compactness of Indexes

A size of one index  
can vary  
within the range of

**0.1 % ÷ 5.0 %**

of the size  
of indexable objects.

# Search Speed

Time estimation  
of defining the address of the first  
potential block of data:

$$T \sim \text{LOG}(N)$$

**T** – time of “*logic probing*”,  
**N** – number of indexed objects.

# Search Speed

A speed of fetching  
the result of interval queries  
from a hard disk can be

**10 ÷ 100 times higher than**  
(for the large data array),

the speed of similar operation  
in a standard relational DBMS.

# Search Speed

A speed of fetching  
the result of interval queries  
from a hard disk can be

**1000 times (and more) higher than**  
(for the large data array),

the speed of similar operation  
when solving the problems  
with the use of brute force method.

# Search Speed

**A time of fetching  
the result of interval queries  
from a hard disk**

**depends linearly**

**on objects number in  
result set.**

# Search Memory

Due to  
compactness of indexes,  
the system loads each of them  
**in RAM**  
**entirely**  
before queries are made.



# Search Memory

**A size  
of memory buffers**

**to fetch the data depends  
on user's needs.**

**This size is often infinitesimal  
(~10 megabyte).**

# Reading of Result Set

**Reading**

**the result set  
from a hard disk  
to RAM**

**is optimum:  
magnetic head does not oscillate.**

# *THANK YOU!*

[www.nbitsearch.com](http://www.nbitsearch.com)

*Technology developed with support from [FASIE](#)  
formed by the Government of Russian Federation*

*Novosib-BIT LLC © 2004 - 2011*

*Patented*