

# Tritonn - MySQL with Senna



Sumisho Computer Systems Corporation  
Brazil, Inc.



Sumisho Computer Systems Corporation



# About Tritonn

- Tritonn is a patch for MySQL
  - Replace MySQL original built-in fulltext index with Senna
    - Create indexes and search records with Senna
    - Query syntax is the same with MySQL.
    - Only MyISAM supported
  - Support MySQL 4.1 / 5.0
  - Extend syntax of create index/alter table
    - Handle Senna flags
  - 2ind feature (Second-Index)
    - Use a fulltext index (Senna) and a B-tree index together
- MySQL meets Senna with Tritonn!



# About Senna

- Three tokenizer
  - MeCab (for Japanese)
  - Bi-gram (for Japanese/non-tokenized languages)
  - Delimited (for tokenized languages)
- Fulltext search library
  - License : LGPL 2.1 and later
  - Dependency : MeCab (morphologic analysis engine/work with dictionary)
  - Plathome : Linux / FreeBSD / Windows
- Support multi encodings
  - UTF-8 / latin1 / koi8r / EUC-JP / Shift-JIS



# Problems with MySQL built-in fulltext index

- Language specific
  - Cannot handle text not tokenized with space like Japanese/Chinese/etc.
- Performance/Capability
  - Phrase search is slow (ex. “united states”).
  - Updating index is slow compared to newly-creating.
  - Using fulltext and B-Tree indexes together to search/sort records is not available.



Tritonn solves !



# Comparative table

Data : Wikipedia English 458,713 records 1,088 MB

	Built-in	MySQL with Senna
Index size	109 MB	1028 MB
Phrase search with "united states"	44.91 sec	0.40 sec
Create index <b>AFTER</b> insert records	1,474 sec	1,808 sec
Create index <b>BEFORE</b> insert records	28,182 sec	1,839 sec
Ordering records containing 'united' by primary key	20.33 sec	0.89 sec
Search records containing 'united' and primary key > 1000	6.55 sec	0.32 sec



## Feature comparison

- All feature implemented original fulltext search is implemented by Tritonn without query expansion, min\_word\_len and stop words.
  - All of operators supported in MySQL original built-in fulltext boolean search are supported.
  - You can get score to call MATCH() AGAINST().
  - No extra operation is required without CREATE INDEX/ALTER TABLE.
- Tritonn/Senna has extended features.
  - Despite boolean mode, records are sorted by score.
  - Search related document.
  - Search words which appears in specified number of tokens.



# Why Senna is fast ?

- Data Structure
  - Full Inverted Index
    - Holds not only record-id but also positions of a word within a document
    - Phrase search is fast
- Implementation
  - Buffering
    - Cache some segments on memory
  - Read lock-less
    - High parallel performance
  - Auto vacuum
    - Relocate discarded segments on index.



## Index of Senna has many information!

- Handle multi-sectional document
  - Index holds record id and section id
    - In database, view a section as a column/field.
  - Tritonn will use this feature.
- Keep customizable score
  - Index holds score multiplier to a part of section
    - If a query word is appeared in the specified part of section, score is multiplied.
- Index compression with fast original algorithm
  - Less memory/storage

# Features on the horizon

- Support partial column search with weight
  - If Columns (title, summary, body) is indexed with senna, these queries can be executed correctly.
    - MATCH (body) AGAINST ('query')
    - MATCH (title, summary) AGAINST ('query')
    - MATCH (title, summary, body) WEIGHT (10, 5, 1) AGAINST ('query' in boolean mode)



# Conclusions

- Tritonn
  - Fast updating
  - Fast phrase search
  - Fulltext and B-Tree index are able to be used together
- Tritonn Site (only Japanese)
  - <http://qwik.jp/tritonn/>
- Senna Site
  - <http://qwik.jp/senna/>
- Why don't you embed senna to your storage engine :-) ?

# Importance of phrase search in Japanese



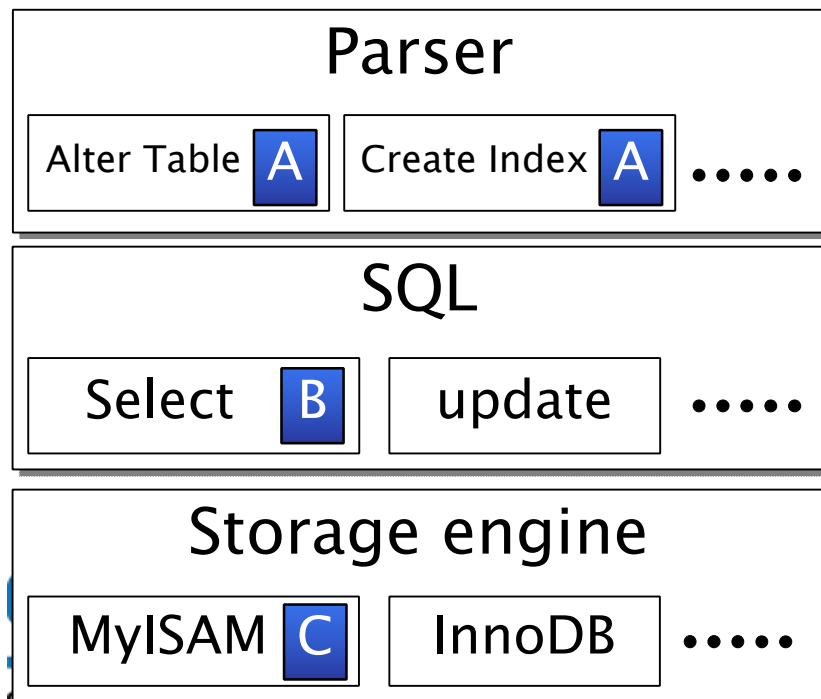
## Why performance of phrase search is required ?

- Japanese words are tokenized to very small pieces
  - Ex. 日本語 (Japanese Language) -> ‘日本’ (Japanese) ‘語’ (language)
- Query string which is recognized ‘a word’ by user is often not a word but tokenized to some tokens.
  - Phrase search is required because user recognize ‘日本語’ as one word. ‘日本人が話す英語’ (English spoken by Japanese) which is not contains a sequence ‘日本語’ but contains ‘日本’ ‘語’ should not be included in search results.



# Where is MySQL patched by Tritonn ?

- Insert/Update/Delete/Select with fulltext index
- Fix an index used a query
- File scan
- Enhancement syntax to specify Senna flags



- A. Handle Senna flags specified by user
- B. Patch logic to fix an index and read\_record()
- C. Patch MyISAM Handler (Insert/Update/Delete Index, Search with Index)



# Index structure/files

- Create SEN files on the directory which MYD file exists.

