**PERSONAL INFORMATION**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | Yuankun Wu | **Gender** | Male |
| **Birth Date** | 01/07/1986 | **Work Experience** | 4 years |
| **Mobile Phone** | 86-15810951365 | **Email** | 573737693@qq.com |

**EDUCATION & WORK EXPERIENCE**

|  |  |  |
| --- | --- | --- |
| 03/2014-07/2016 | Map search department of AMAP | Senior R & D Engineer |
| 06/2012-03/2014 | Advertising R&D department of Panguso | Algorithm R & D Engineer |
| 09/2009-05/2012 | University of Science and Technology of China | Master |
| 09/2005-07/2009 | East China Jiaotong University | Bachelor |

**TECHNICAL ABILITY**

|  |  |
| --- | --- |
| Technical Foundation | Mastering C/C++, familiar with python and shell;  Mastering basic data structures & algorithms and design patterns;  Mastering multithreading concurrency, synchronization programming, and TCP/IP protocol and socket programming;  Proficient in Linux environment. |
| Technical Expertise | Index structure design and performance optimization of online search engine;  Architecture design of online search service. |

**PROJECT EXPERIENCE**

|  |
| --- |
| AMAP’s Dialogue System(03/2016-06/2016)  Role: Architecture design member & development team leader  Dialogue system is divided into 4 layers: services framework layer, session state controller layer, business policy layer and data adapter layer.  (1) Services framework layer controls the high concurrency flow, and manages internal infrastructure components.  (2) Session state controller layer fully maintains the user’s session state. It calls the semantic module to parse the current query-text, and then, according to the current semantic and context state, determines the next scene-intention-state. After that, the controller schedules corresponding policy module to process the task.  (3) Business policy layer. In this layer, every individual policy component generates specific tasks based on its own business intention, and schedules corresponding data-adapter to request data. And then, the policy component renders the data to generate slot information and patter. The business policy layer use the plug-in interface design, this can insert many other vertical products easily and upgraded with low cost.  (4) Data adapter layer. This layer isolates business data services, provides request-interface to different business systems, and translates the source data to internal data’s structure for the upper policy module.  So far, the dialogue system has successfully supported instruction operation, map poi searching, navigation planning, prattle, traffic information querying. Other entertainment business will be connected in the future. |
| HSDT engine (A model-dictionary retrieval engine) (08/2015-11/2015)  Role: Architecture design leader & development team leader  HSDT engine consists of model-dictionary manage platform and retrieval engine.  (1) Model-dictionary manage platform monitors all model dictionaries’ update-state in its own process transparently, and controls the update operation of each dictionary in a serial event flow, in order to prevent memory spikes.  (2) Retrieval engine can automatically compute the optimal hash table and automatically select one best hash function from those functions in the engine. So, the engine can reduce the terms’ collision rate.  In the point of index storage space, we migrate the conflict from the hash table to the data segment, then the hash table reduced to one integer array; In the data segment, all the key-values terms are continuously stored in the memory, and all the terms in one bucket are clustered, thereby, we use the term’s length to express and link the conflict items, it is called offset-bytes. The offset-bytes can be compressed automatically. All these have been effectively improved the utilization of memory resource. At the same time, it also reduces the jumps of memory access, improves the hit rate of CPU cache access.  Performance of the engine. The average conflict ratio of the billion scale dictionary’s index is 1.08, and the engine access performance is at 1us scale. The rate of the binary dictionary’s memory expansion is about 5% on average (compared to text dictionary’s size).  Meanwhile, the engine provides supporting offline tools: dictionary index generator and index checking tool.  The HSDT’s interface is very simple. Now, HSDT has been fully applied in the search department. The strategy analysis module, semantic analysis module, word segmentation module, error correction module and suggestion module all have been involved in HSDT, for the replacement of the original TDBM engine developed by yitao department of Alibaba. |
| Map search engine optimization(11/2014-05/2015)  Role: Architecture development member  The optimization is divided into three parts: index slimming, query process optimization and detail service separation.  (1)The design of index slimming is divided into two aspects. Inverted-index chains automatic match the original information, or blocked PForDelta, or bitmap methods to compress and store; in the filter-fields, we delete the field that can be inferred. Besides, we use field value encoding and BitRecord methods to pack the fixed length field. The result is that the Index loading space is reduced by 30%.  (2)In the query process optimization, as the PForDelta inverted-index chains have been divided into blocks of same size, then the binary chop can be used, and we just needed to decode the target block to get the document; Besides, we pre-process all documents in order by poi-weight, and then the progressive method can be used in the query process. After those above steps optimization, the poi retrieval process can finish in a controllable depth and in a small intermediate result scale (in particular the case of high query latency). This optimization has improved the average performance of the whole retrieval process by 5 times, and reduced the average time of request query about 5 ms.  (3) The separation and independence of detail services. It has effectively reduced the search servers’ memory footprint, and greatly shortened the time of real-time displaying data up to the on-line. |
| Search service disaster tolerance system(07/2014-09/2014)  Role: Architecture development member  The disaster tolerance system is divided into two parts: the disaster tolerance service and online service overload protective framework.  (1) The disaster tolerance service consists of generalization cache and insulation delayed environment. Every morning, generalization cache is filled with the training result, which is calculated according the previous day’s request log. The delayed environment is in a completely isolated state, it is the supplement to cache. When one request is not hit in the cache, it can re-request to the delayed environment, in order to reduce the no-result rate.  (2) Online service overload protective framework, according to the busy status of the workers in the process, can adjust its query window to protect itself maintaining in a stable state for responding certain amount request. It works in three stages: overload warning, overload protection and overload removed. |
| Advertising engine (2012, Panguso)  Role: Development member  The advertising engine is divided into 4 layers: framework layer, business layer, control layer, and data engine layer.  Framework layer responses external service and balances internal concurrency load scheduling.  Business layer parses all different business related request, and renders result.  Controller layer provides recalling and filtering interface to the underlying basic data layer, and handlers the modify request to update the index.  Data engine layer is in charge of loading and updating index data, and associating multi-level table in memory with the offset relation. Besides, in the memory, the data of one document is distinguished and continuously stored into two parts, the fixed-length-field segment and the variable-length-field segment. Then we can get all data of one document in twice memory access.  Comparing to web search system, in the advertising search system, advertisers have lots of modify-operations on the filter-fields, such as, keywords, bid, budget, or the advertising area and times. About 40% advertising plans have this modify operations, based on the stat of management platform’s log, this cause the index dynamic update frequently.  We changed those filter-fields of multi-values into <field, value>’s 0 or 1 bitmap-mapping. This changes income us two respects: 1) The index dynamic update can be regenerated in situ. Advertisers’ modify-operations are translated into 0 to 1 or 1 to 0 bit-set operating. 2) The bitmap is efficiently to handle the filter-operation. Besides, the bitmap is stored in contiguous memory space, this feature can improve the CPU-cache hit rate, compared to the random memory accessory of the filter-fields. |