(54) **SYSTEM AND METHOD FOR REDUCING THE FOOTPRINT OF PRELOADED CLASSES**

(75) Inventors: **Hideya Kawahara**, Mountain View, CA (US); **Nedim Fresko**, San Francisco, CA (US)

(73) Assignee: **Sun Microsystems, Inc.**, Santa Clara, CA (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

(21) Appl. No.: **09/045,508**

(22) Filed: **Mar. 20, 1998**

(51) Int. Cl.^{7} **G06F 9/00**

(52) **U.S. Cl.** **709/332**; 717/166; 717/159

(58) **Field of Search** 709/332, 331; 717/151–167

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,734,822 A	*	3/1998	Houha et al.	709/216
5,815,718 A	*	9/1998	Tock	709/331
5,966,542 A	*	10/1999	Tock	713/1
5,966,702 A	*	10/1999	Fresko et al.	707/1
6,052,778 A	*	4/2000	Hagy et al.	709/331
6,061,520 A	*	5/2000	Yellin et al.	703/22
6,223,346 B1	*	4/2001	Tock	713/1
6,324,637 B1	*	11/2001	Hamilton	711/216
6,339,841 B1	*	1/2002	Merrick et al.	707/103 R
6,349,344 B1	*	2/2002	Sauntry et al.	709/1
6,363,436 B1	*	3/2002	Hagy et al.	709/331
6,366,898 B2	*	4/2002	Taivalsaari et al.	707/1
6,526,565 B1	*	2/2003	Nally	717/108
6,530,080 B2	*	3/2003	Fresko et al.	717/166

FOREIGN PATENT DOCUMENTS

EP		943989 A2	* 9/1999	G06F/9/44

OTHER PUBLICATIONS

Sun Microsystems. "PersonalJava 1.1 Memory Usage Technical Note." 1998.*

"The Java™ Virtual Machine Specification", Tim Lindholm, Frank Yellin, 475 pages, ©1997, ISBN 0201–63452–X.

* cited by examiner

Primary Examiner—John Follansbee
Assistant Examiner—Lewis A. Bullock, Jr.
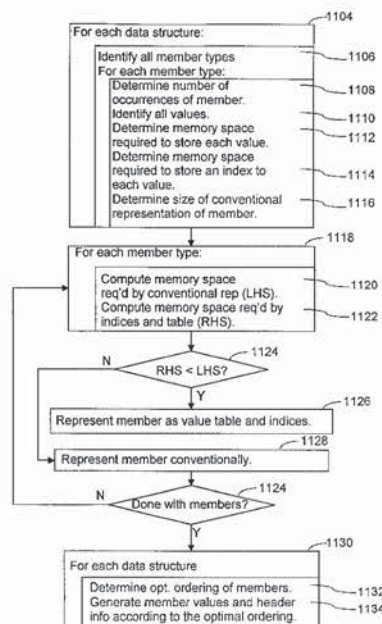(74) *Attorney, Agent, or Firm*—Pennie & Edmonds LLP

(57) **ABSTRACT**

A method and system that reduces the space allocated for internal data structures by a runtime engine. The internal data structures store member information for preloaded classes used by applications executed by the runtime engine. The system determines the different types of internal data structures represented in the classes and identifies thee possible values of each type's members. The system next determines the amount of space required to store the values for each type in a respective value table and the number of bits needed to index each entry of that table. The system determines based on the stored information whether occurrences of a member are optimally represented as a set of value table indices and a value table or, in the conventional manner, as a general variable that stores the member's value for each occurrence. The system then emits appropriate information for the member and its parent data structure.

44 Claims, 12 Drawing Sheets

Class File
400

Header Info	— 402
Constant Pool	— 404
Methods Table	— 406
Fields Table	— 408

• • •

FIG. 1
(Prior Art)

1200

One per Class

Class Block
1202

1230 — Class Name

1232 — Superclass | ptr

1234 — Method Blocks | ptr

1236 — Field Blocks | ptr

1238 — Constant Pool | ptr

. . .

Method Blocks
1204

Method Blk 1
1204.1

Method Blk 2
1204.2

. . .

Method Blk N
1204.N

Constant Pool
1224

Field Blocks
1214

Field Blk 1
1214.1

Field Blk 2
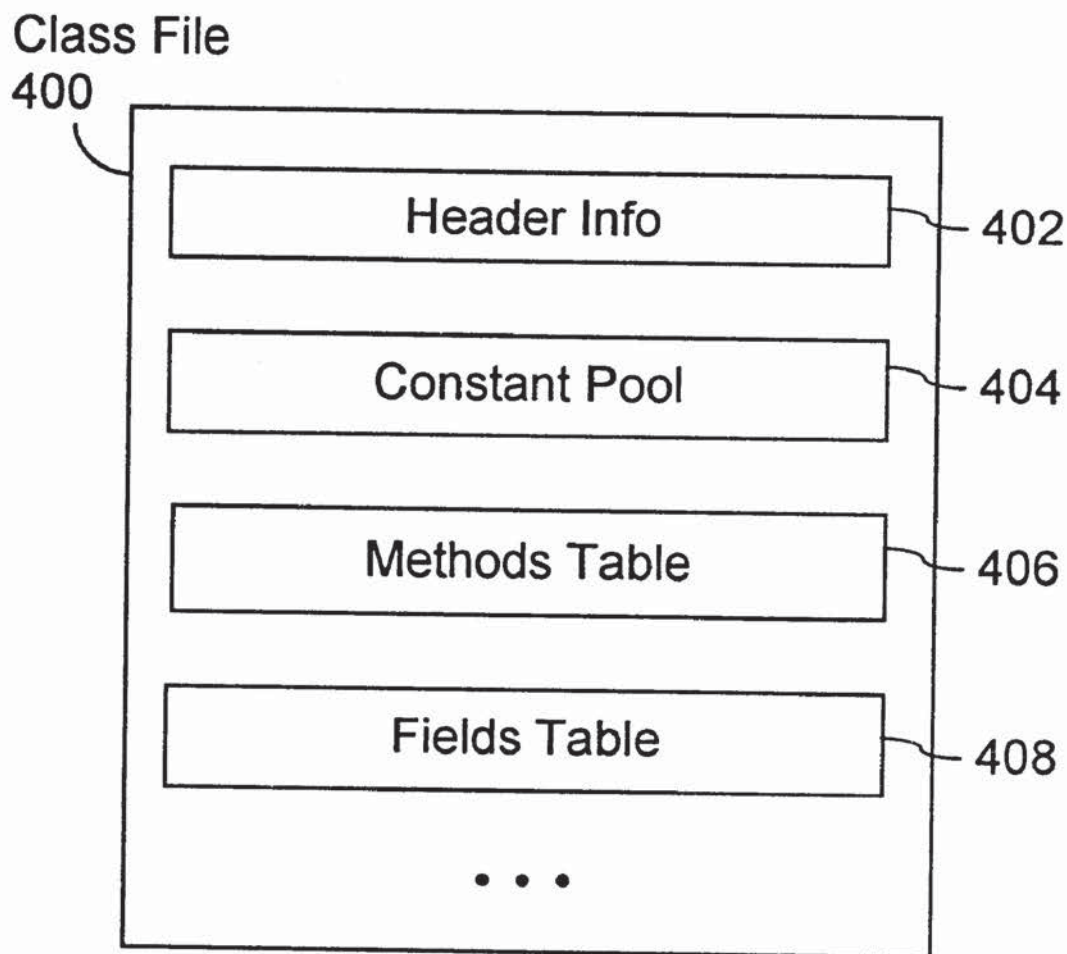1214.2

. . .

Field Blk N
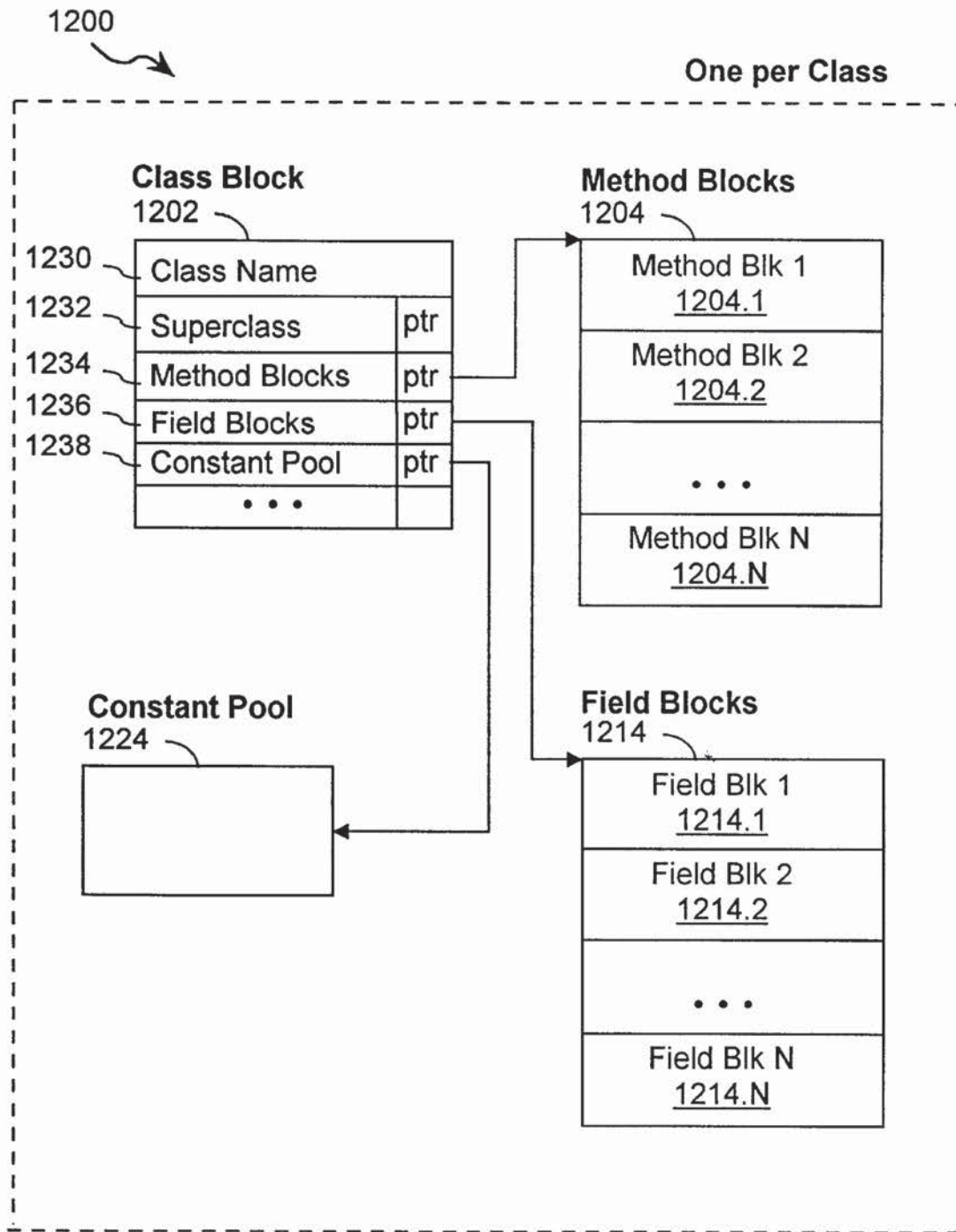1214.N

FIG. 2A
(Prior Art)

Data Structure Declarations
1230

Data Structure 1 Decl.	1230.1
Member 1.1 Decl.	1232.1.1
Member 1.2 Decl.	1232.1.2
Member 1.3 Decl.	1232.1.3
. . .	
Member 1.1 Acc'or Fun	1234.1.1
Member 1.2 Acc'or Fun	1234.1.2
Member 1.3 Acc'or Fun	1234.1.3
. . .	
Data Structure 2 Decl.	1230.2
Data Structure 3 Decl.	1230.3
. . .	
Data Structure N Decl.	1230.N

1230.N

```
Struct T {
    mtype1 member1                          1232.N.1
    mtype2 member2                          1232.N.2
    mtype3 member3                          1232.N.3
    mtype4 member4                          1232.N.4
    mtype5 member5                          1232.N.5
}
member1 of (T) : T->member1                 1234.N.1
member2 of (T) : T->member2                 1234.N.2
member3 or (T) : T->member3                 1234.N.3
member2 of (T) : T->member2                 1234.N.4
member3 or (T) : T->member3                 1234.N.5
```
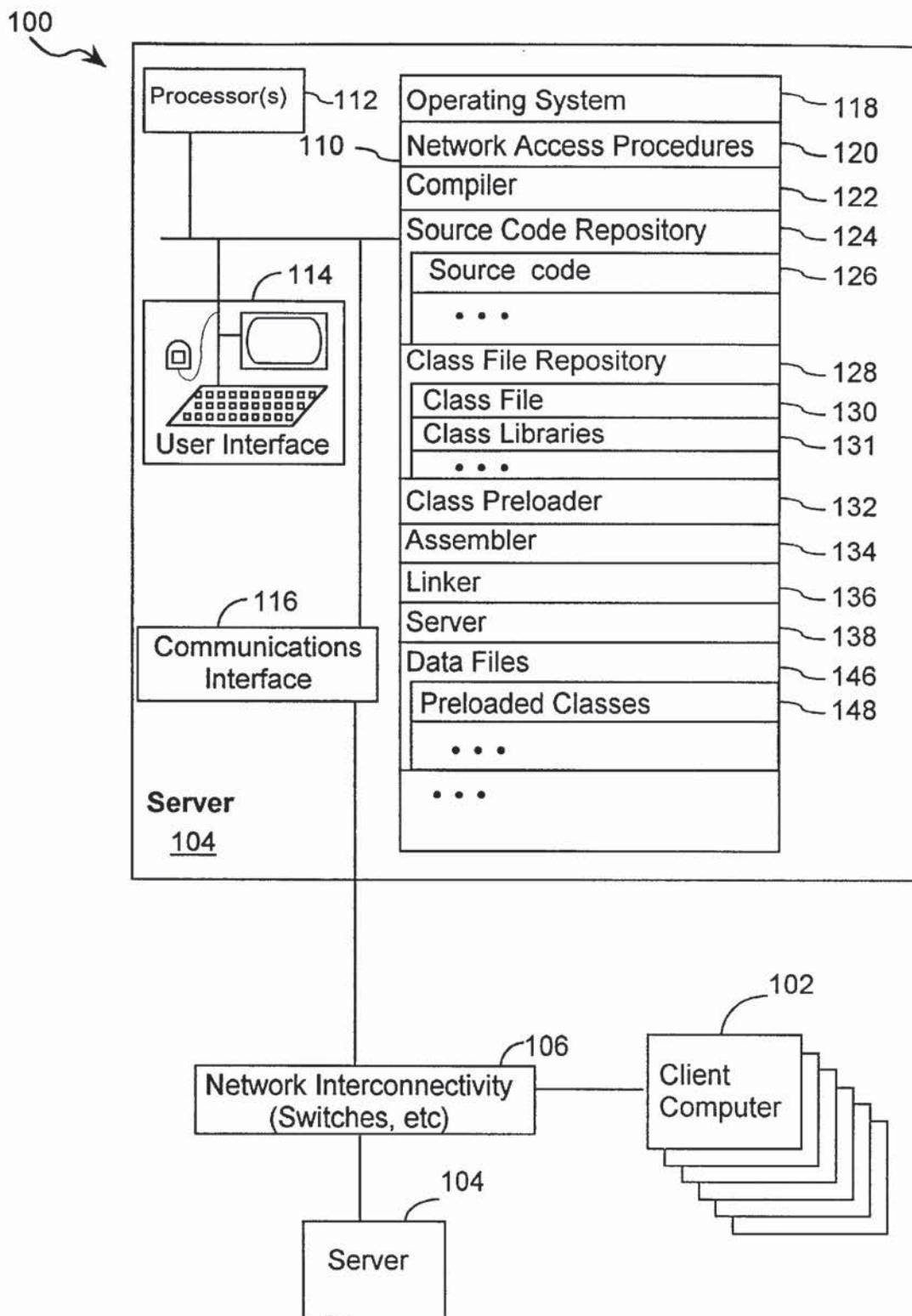
FIG. 2B
(Prior Art)

100

Processor(s) — 112

110 —

114

User Interface

116

Communications Interface

Server
<u>104</u>

Operating System	— 118
Network Access Procedures	— 120
Compiler	— 122
Source Code Repository	— 124
Source code	— 126
• • •	
Class File Repository	— 128
Class File	— 130
Class Libraries	— 131
• • •	
Class Preloader	— 132
Assembler	— 134
Linker	— 136
Server	— 138
Data Files	— 146
Preloaded Classes	— 148
• • •	
• • •	

Network Interconnectivity
(Switches, etc)

106

102

Client
Computer

104

Server

FIG. 3

DOCKET ALARM

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS
Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS
Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS
Sync your system to PACER to automate legal marketing.

fastcase
Smarter legal research.