Prefetch Policies for Large Objects in a Web-Enabled GIS Application [†]

Dong-Joo Park   Hyoung-Joo Kim

School of Computer Science and Engineering, Seoul National University
San 56-1, Shillim-dong, Gwanak-gu, Seoul 151-742, KOREA

{*djpark, hjk*} *@oopsla.snu.ac.kr*

Abstract

In a Web-enabled GIS application, it would be possible for users to navigate existing spatial objects(e.g., points, lines) or spatial query results containing large objects(e.g., raster images, Web documents) on the Web browser. For efficiency, relatively "light" spatial objects exist on the Web browser, while "heavy" real information like large objects resides in the remote server. Only when the users callback real information for a certain spatial object, the server transmits it to the browser. In this paper, we propose several prefetch policies, especially the Hilbert curve-based ones which can alleviate user response time under the assumption that user callback access pattern has spatial locality. We conducted diverse experiments to show that our prefetch strategies based on the Hilbert curve achieve higher efficiency than the other naive or no prefetch ones.

1   Introduction

For the last decade, there has been, in the area of spatial databases and geographic information systems(GISs), a large amount of research in spatial data storage, the support for spatial data types, spatial query optimization, user interfaces, and so on[17, 21, 22, 23]. Recently, attention has been also paid to efficient integration of the Web and GIS application[4, 7, 10]. This is because the Web itself has many advantages, including its popularity, transparent accesses to heterogeneous distributed database systems, proficient display functionalities of the Java-supported Web browser, etc.

Web-enabled GIS systems may allow users to submit SQL-like spatial queries to a remote spatial database server and to look at their results through the Web browser. However, such systems should tackle their inherent problems: the transmission delay of the query results including large objects(e.g., raster image, Web documents) as well as spatial objects(e.g., points, lines, polygons), the presentation of spatial objects on the Web browser, efficient integration of the Web and the spatial database server, and so forth. Of them, the delay of user response time due to the transmission of "heavy" query results from the server to the client is fairly critical.

In the 2-dimensional map on the Web browser, users can submit SQL-like spatial queries such as: