A Client-Server Based Architecture for Parallel Design Query and Analysis

IEEE/DATC EDP 2000 Workshop

Bruce Winter
David Hathaway
IBM

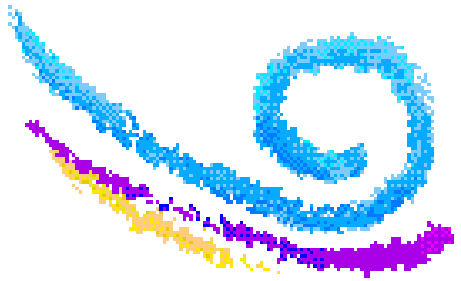
ıvironment: ASIC Physical Design

Placement, Clock and Timing Optimization, Wiring, and Post-PD checking

Team of 6, doing about 30 chips per year

Small, slow chips in older technologies (stable rules) may take only a few days

Large, fast chips in new technologies (unstable rules) may take months



ıe Problems

Sharing data

- Large databases. For example:
  - 1 million objects
  - 500 Meg netlists
  - 700 Meg timing extraction
  - 1000 Meg wire files
  - Takes 30 minutes to load the model
  - Requires large box: 64 bit, > 2 Gig of memory
- Many people (PD guys and designers) working on the design at the same time

ıe Problems *(continued)*

Need quick solutions to dynamic problems

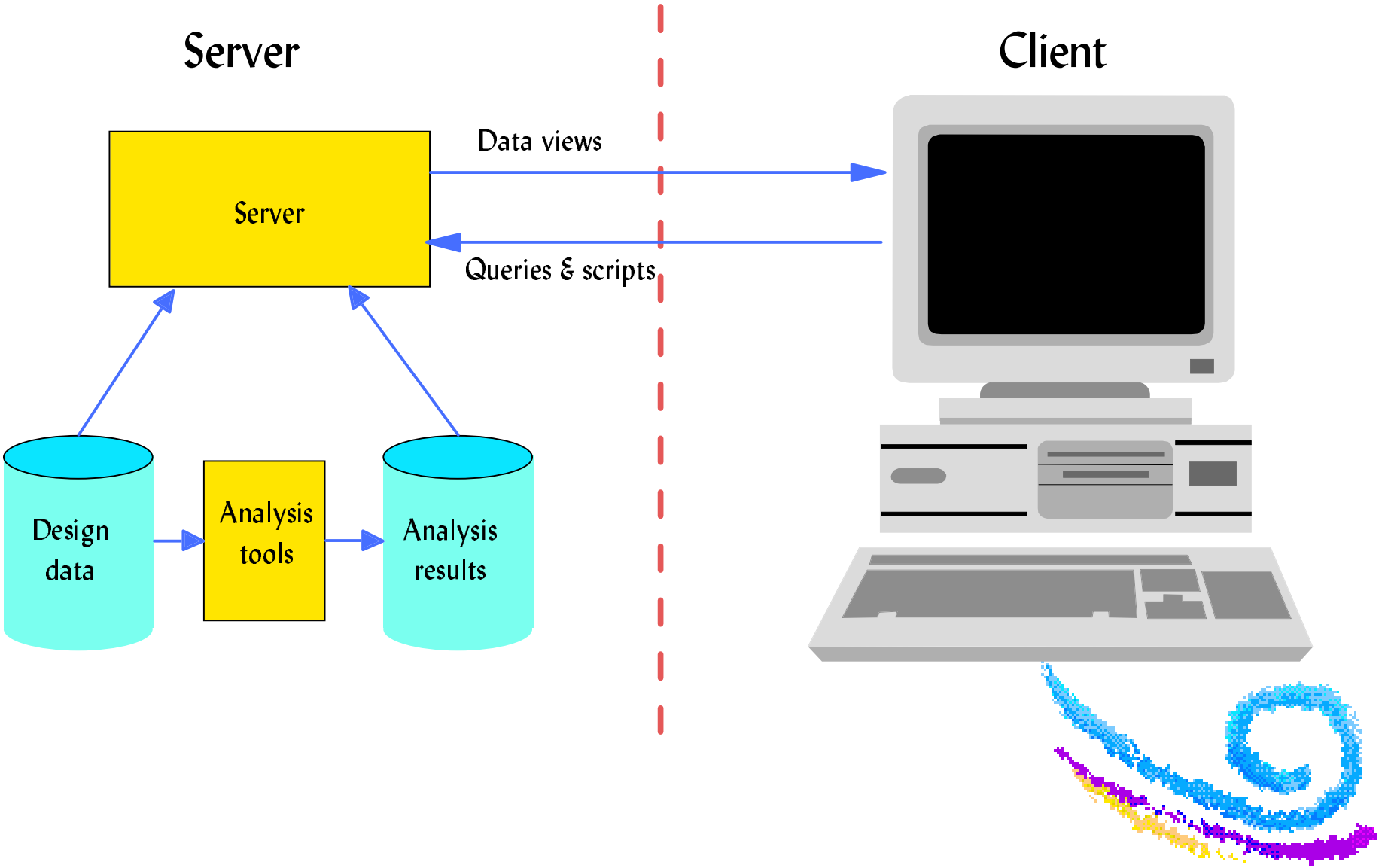
- Requirements are constantly changing.
  - Technology rules
  - Design Specs
- Examples:
  - A clock buffer is determined to have reliability problems, so we need trace through various clock trees and substitute clock inverters.
  - Temperature or VDD specs change, so we need to adjust padding books to fix fast chip hold violations.

ıe Solution

An idea from the 80s: Client/Server

- Load the data once, on a big, fast server, and write clients to read and analyze the data.
- Solves data sharing problem
  - Multiple clients can quickly and simultaneously read the data
  - Multiple designers creating ECs in parallel.
- Solution to the dynamic problem
  - Quickly code, test, and use small chunks of throw away client code.

ocess flow


Server
Client
Server
Data views
Queries & scripts
Design data
Analysis tools
Analysis results

:ample: netlist_server

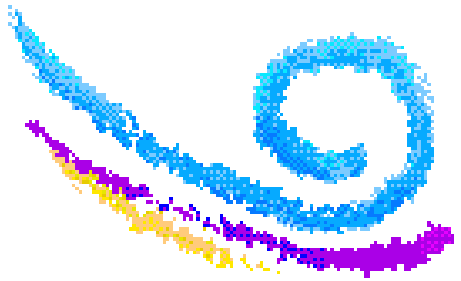
netlist_server: reads the netlist, placement, and wiring data

- Written as a stand alone Perl script, but could be integrated into the IBM IDM-based design system
- Forking allows for simultaneous access with shared memory
- For 1 million objects, it runs in 30 minutes and uses 2 Gig of memory
- Registers the hostname and tcp/ip port used, based on chip name and version.

ample: netlist_splot

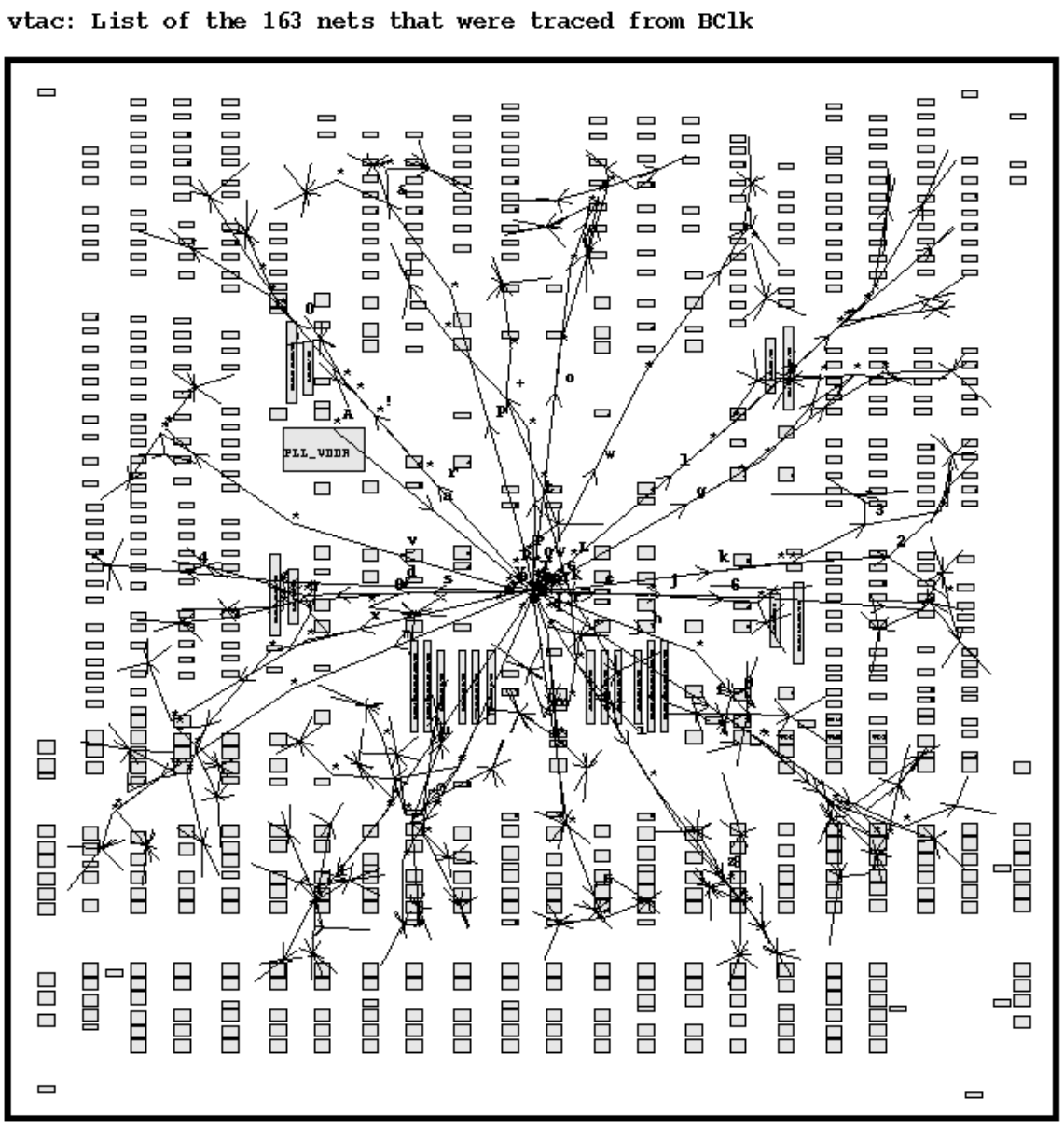
netlist_splot: plots placement and/or wiring data

- Generates and displays a postscript file in a few seconds on a entry level box.
- Often run by off-site designers, via the IBM intranet
- Can trace repowering trees, plot group placement, show location of books, display wiring.



:xamples ...

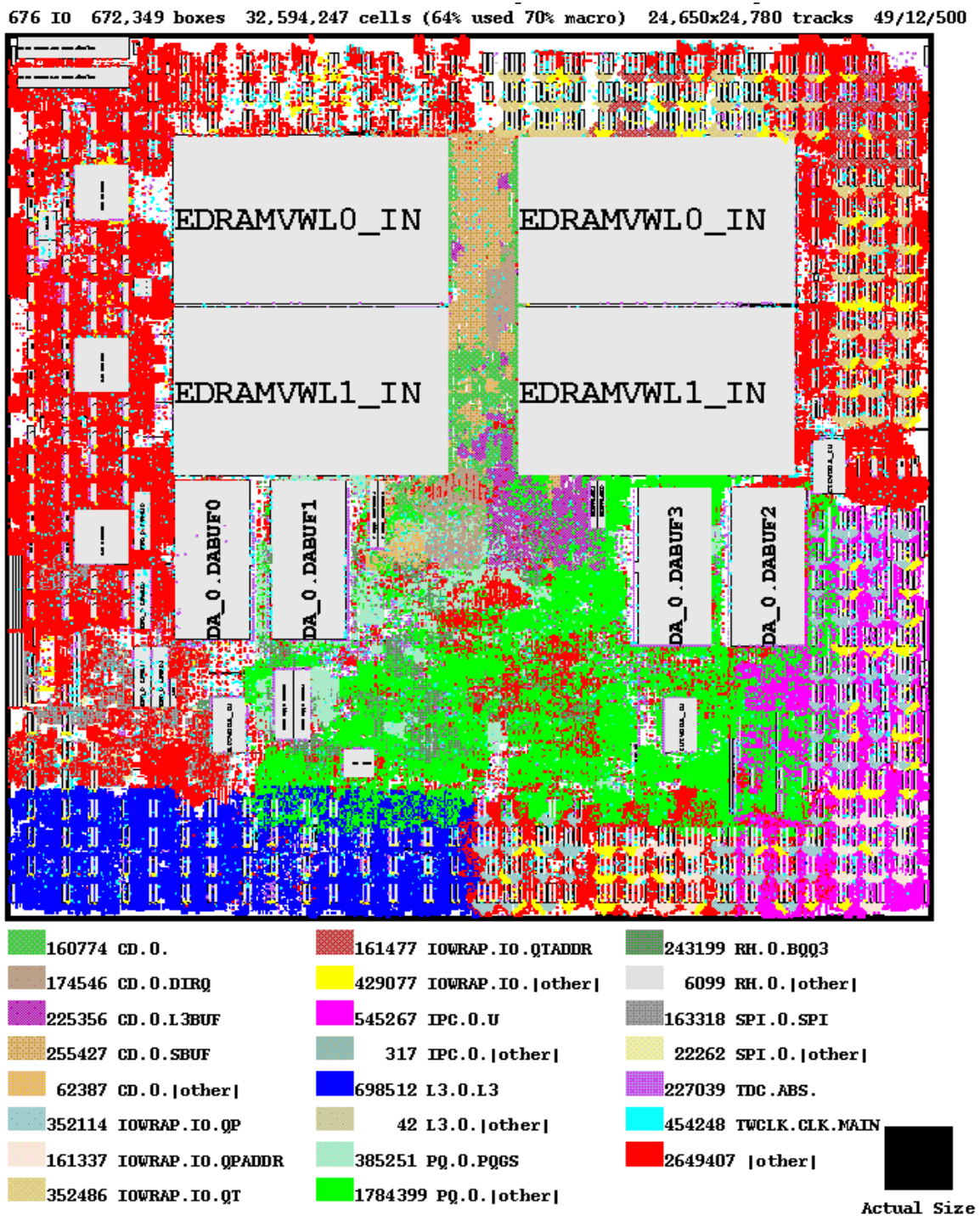
:xample of a clock
ree trace
ietlist_splot
chip vtac
nets_forward
3CLK -level_limit 4



Examples ...

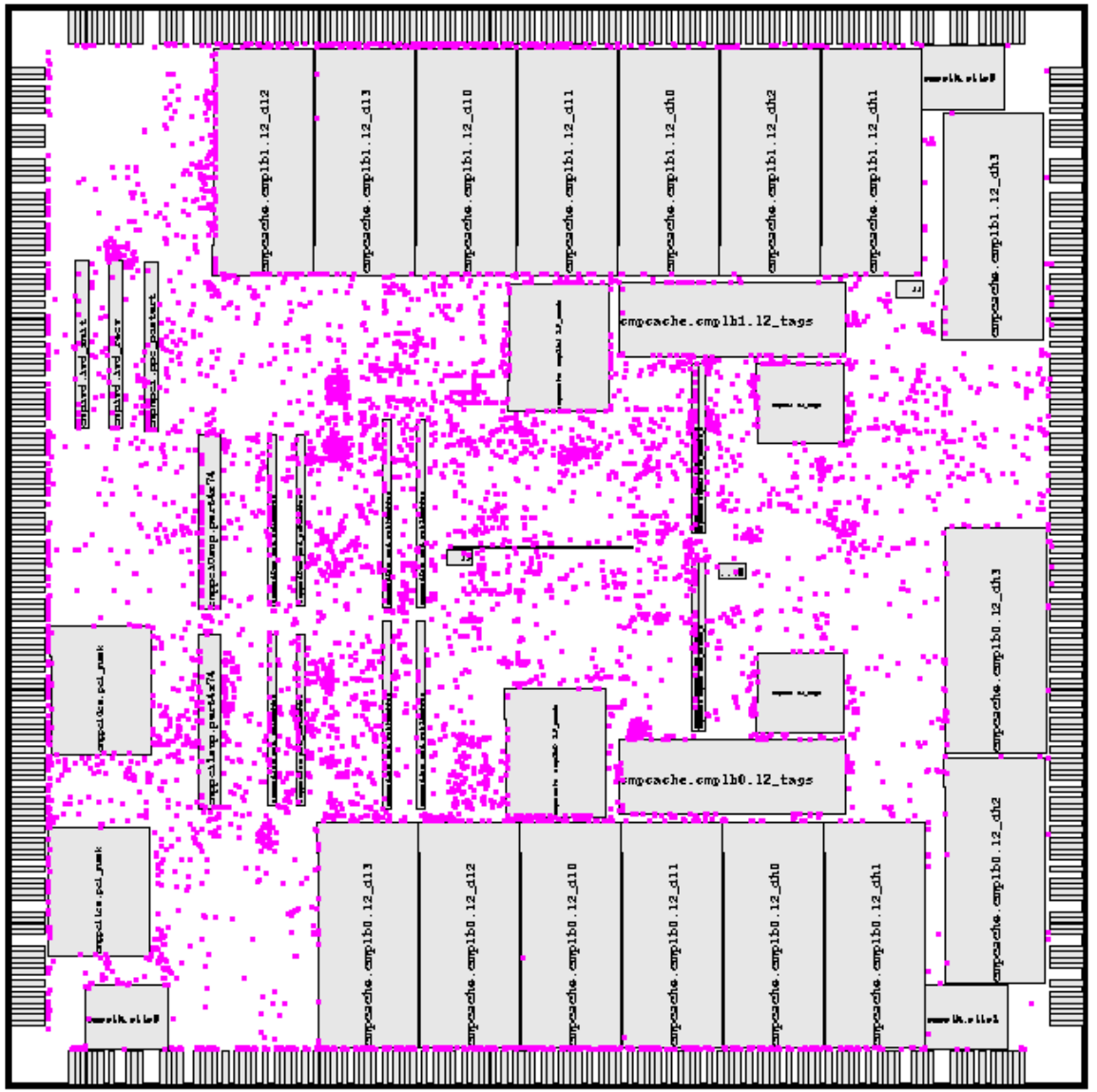
Example of a group placement plot

netlist_splot chip fred group



xamples ...

hows location of
ew repowering
ooks
etlist_splot
chip joe_postopt
boxes PDS.*



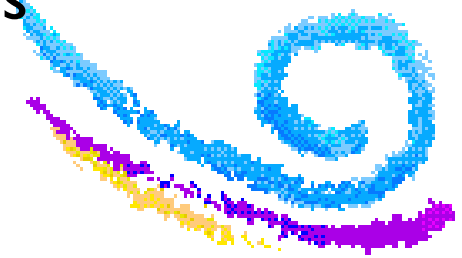
C Process

Multiple designers create ECs in parallel

- Some functional, some for timing
- Some created manually, but most are auto-generated and then hand tuned
  - e.g. netlist_repower: a client tool that generates an ECO to insert a repowering tree.

ECs processed daily

- ECs are collected and checked for overlap (eco_check)
- ECs are applied and legalized, then new timing is run
- Server is restarted, and the cycle is repeated.



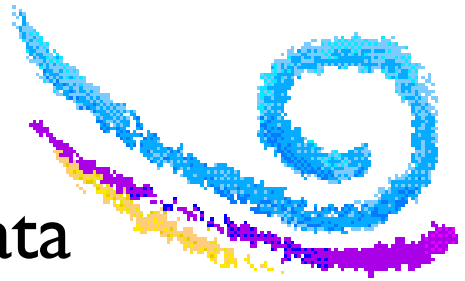
Dynamic Code

Problems are dynamic, so we can not anticipate what functions to put into the server

Some functions would be inefficient in a client (e.g. tracing a tree)

Solution:

- client passes new code to the server
- server forks, interprets and executes the new code on-the-fly, and returns results
- Perl eval function makes this possible
- Quickly write/debug new code on real data



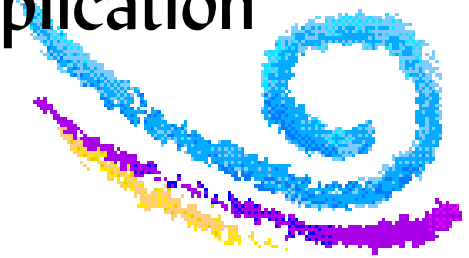
Summary

Client-Server architecture benefits

- Limits number of large machines needed
- Reduces design loading bottleneck
- Allows active queries against model instead of working from static reports

Limitations

- Current implementation does not use IDM model
  - Many analyses unavailable or must be duplicated
- Coordination issues and time lags in ECO application



ossible Extensions

Integration with IBM's IDM data model

- Direct access to all IDM-based analysis tools

Extend the available types of queries

Allow sharing of views for closer collaboration

Introduction of ECOs to active model

- Still have coordination issues

Provide guidance/control to optimization routines operating on active model