

Современные тенденции в разработке средств визуализации программного обеспечения параллельных вычислений

В.Л. Авербух, М.О. Бахтерев, П.А. Васёв, А.П. Уросов

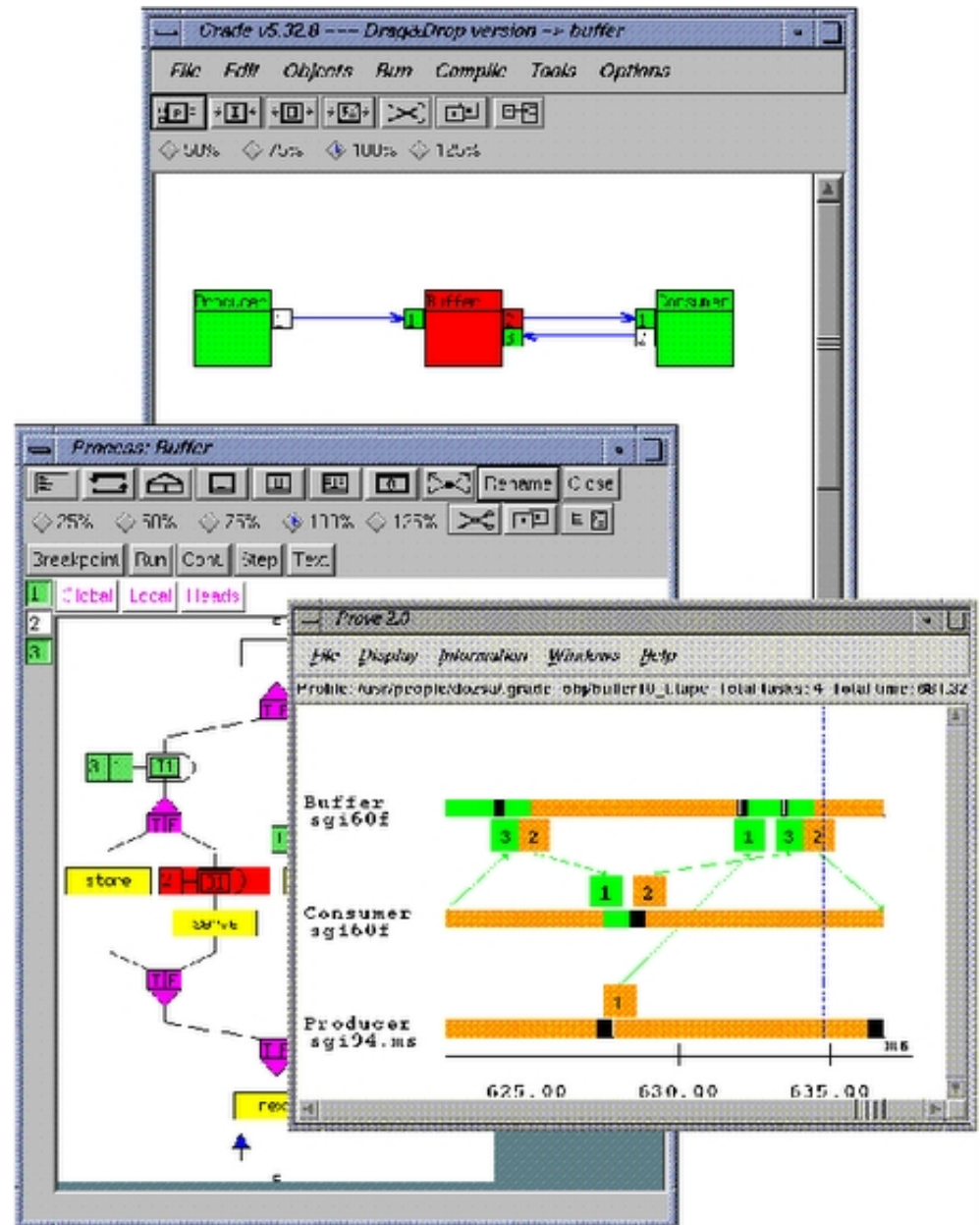
ИММ УрО РАН, УрФУ, г. Екатеринбург

90-е годы XX века

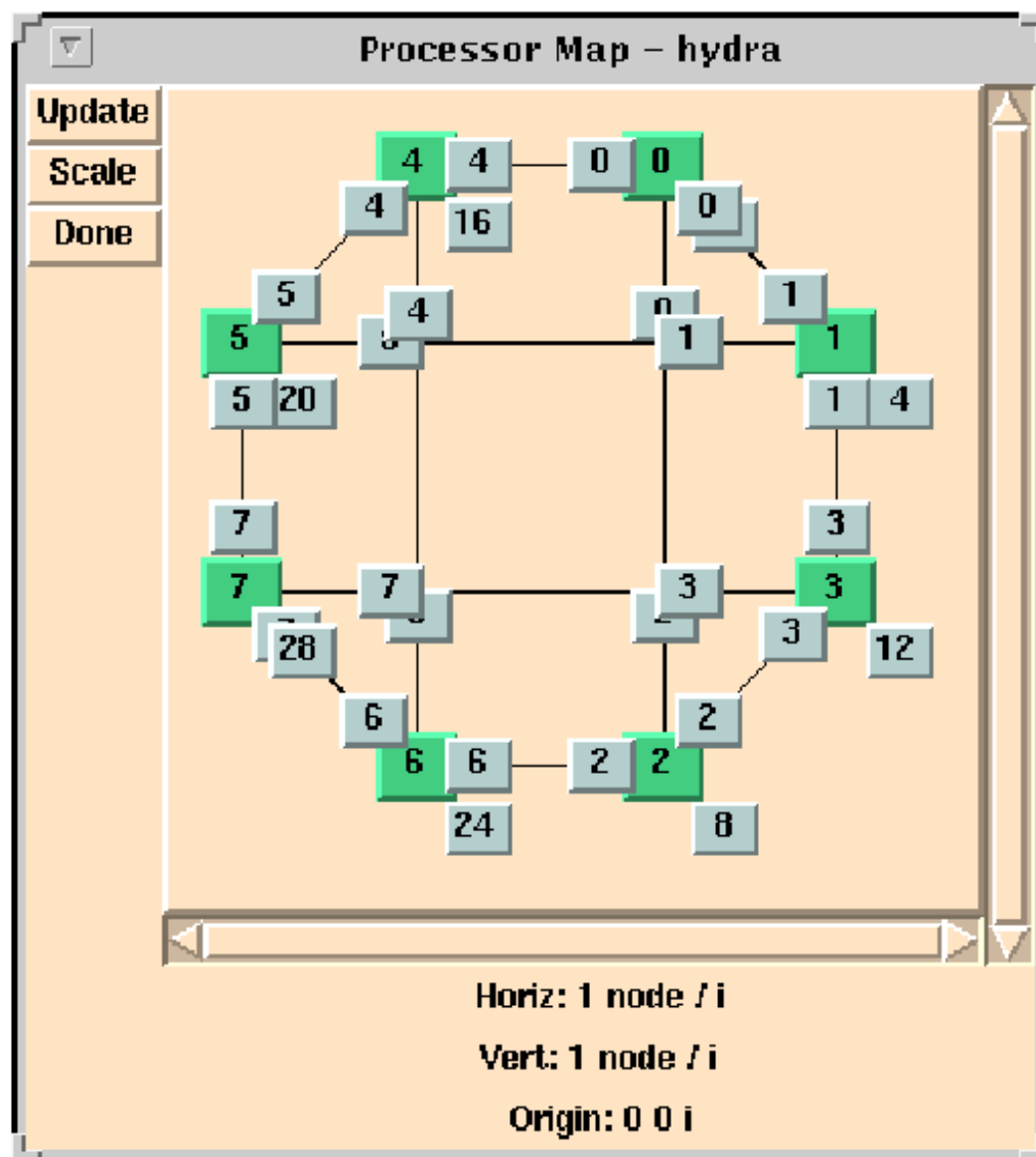
Системы визуализации программного обеспечения параллельных вычислений:

- ◆ визуальные языки параллельного программирования;
- ◆ визуальные отладчики правильности параллельных программ;
- ◆ визуальные средства отладки и настройки эффективности параллельных программ
- ◆ средства визуализации процесса разработки программного обеспечения

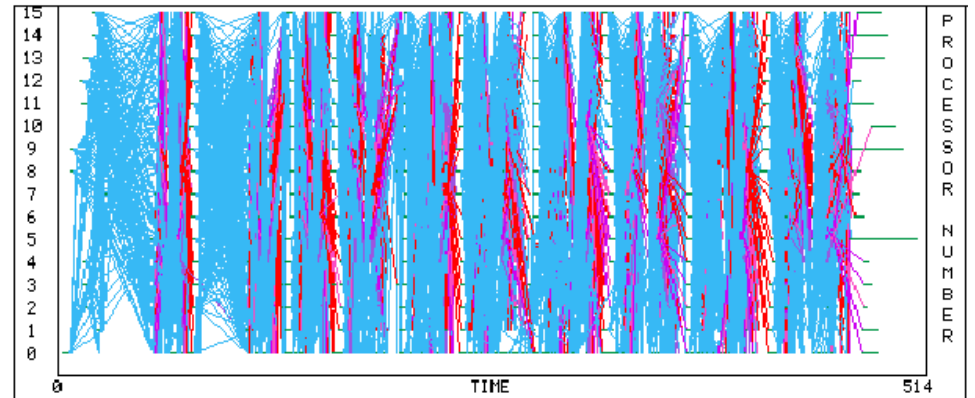
90-е годы XX века



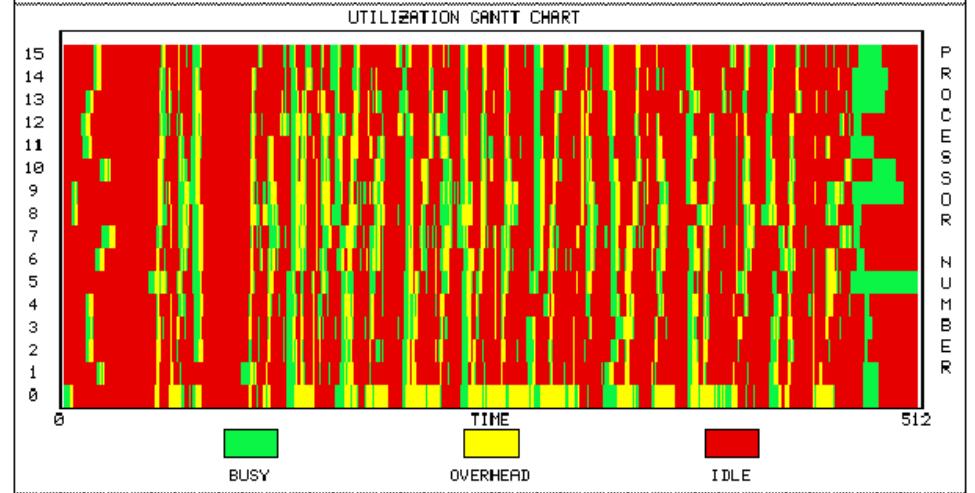
90-е годы XX века



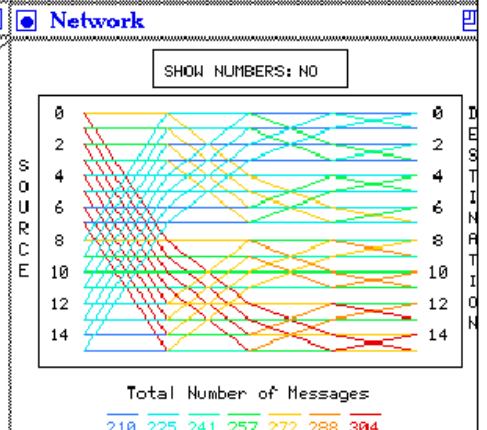
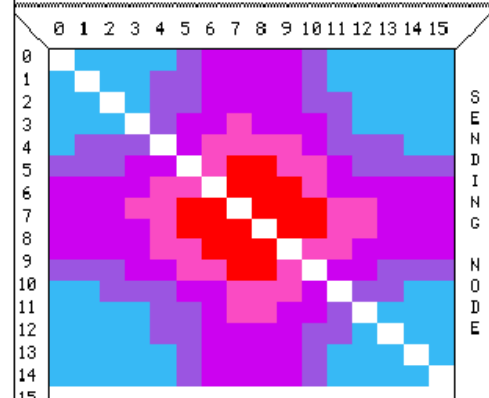
90-е годы XX века



Utilization Gantt Chart

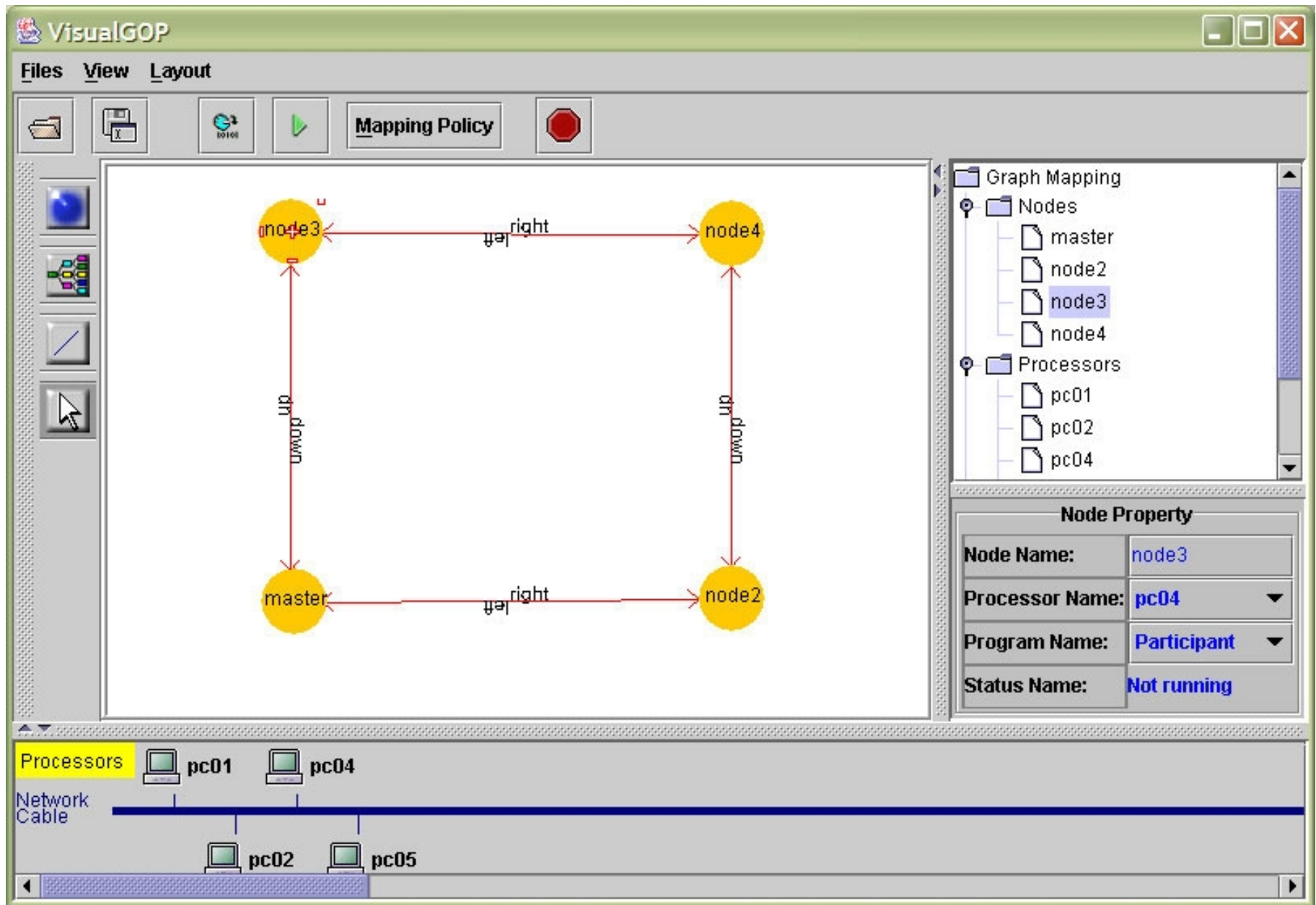


Communication Matrix

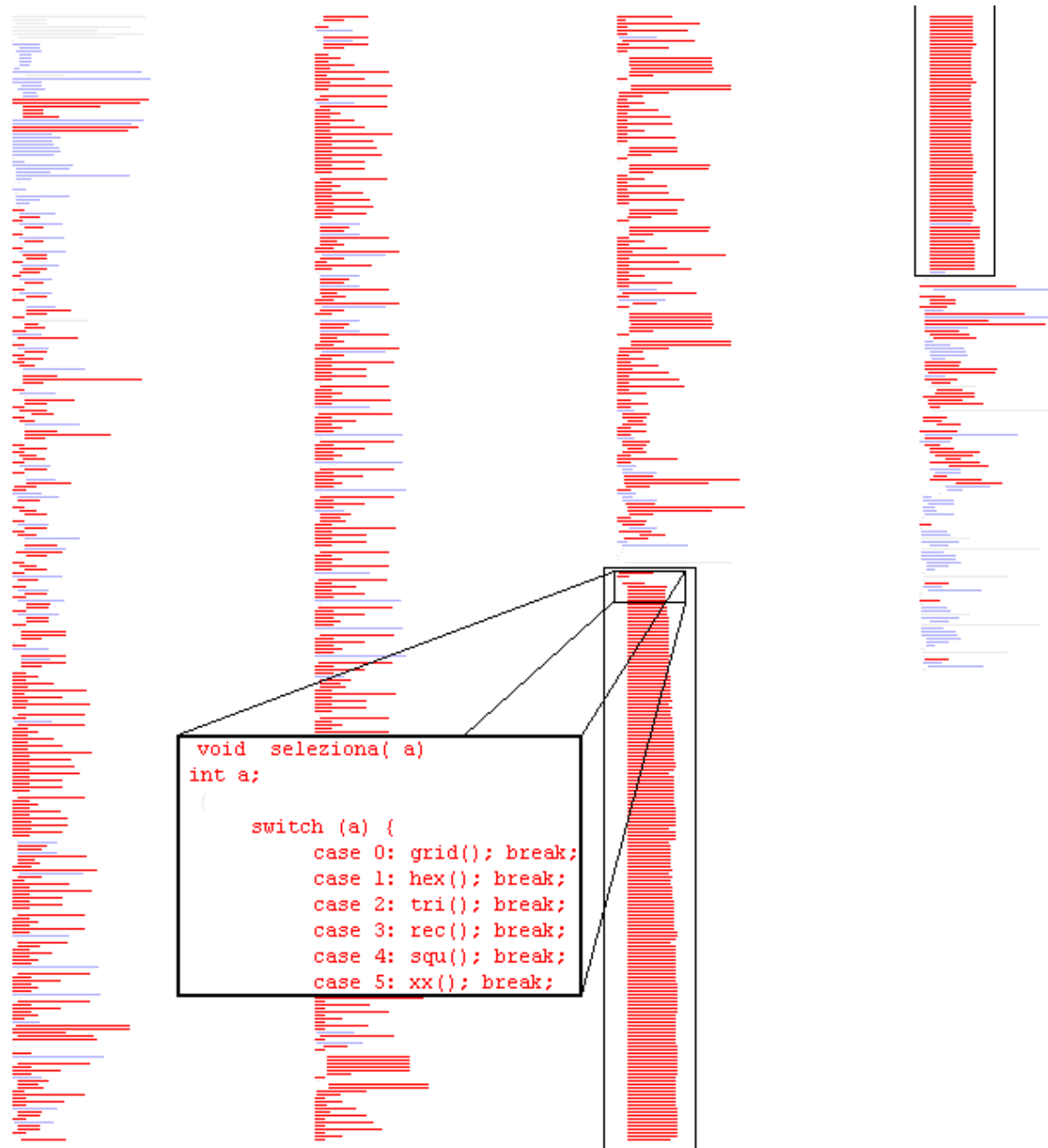


XXI век. Тенденции

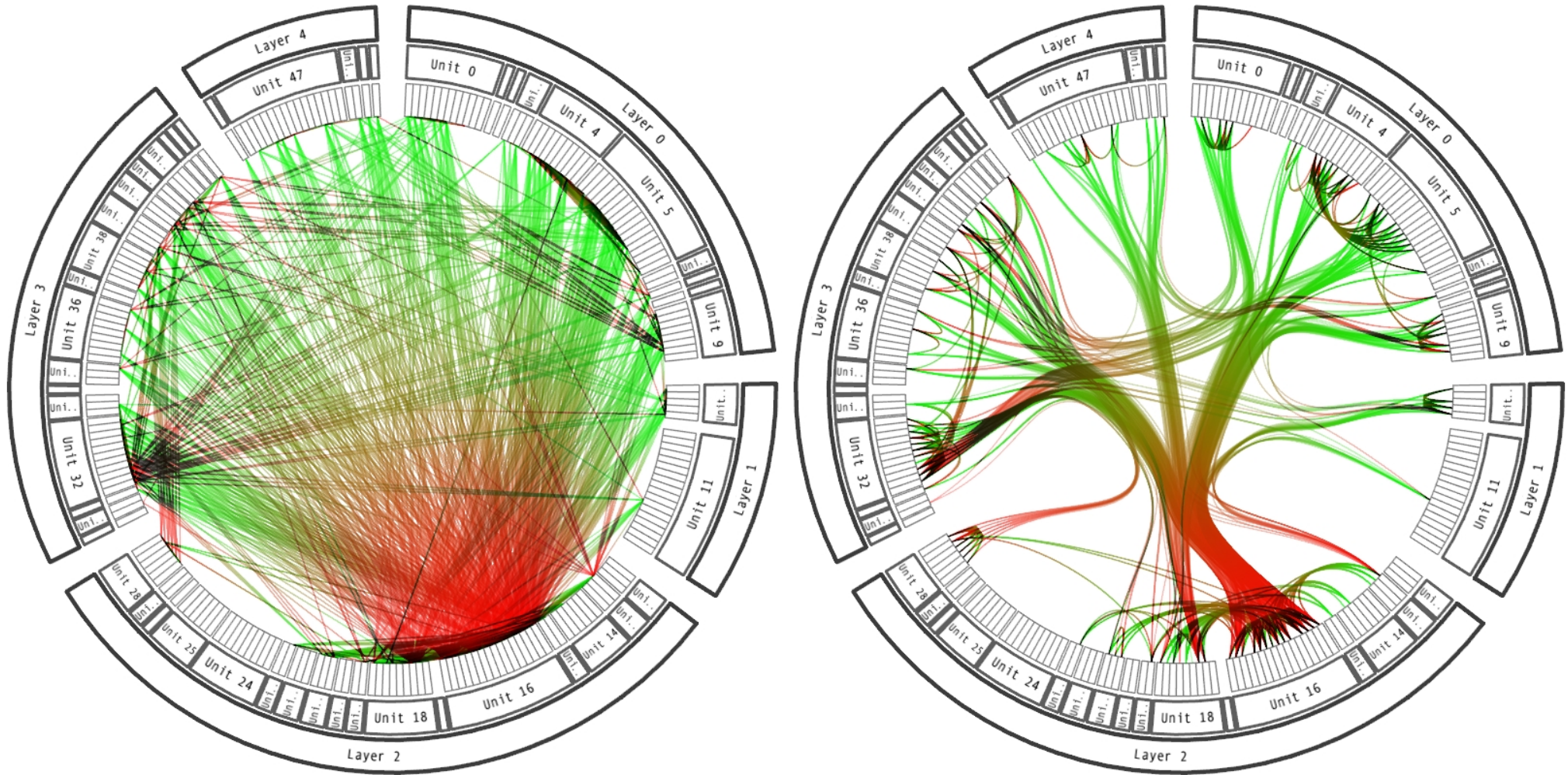
- ◆ Интенсивность разработок в области визуализации программного обеспечения параллельных вычислений снизилась.
- ◆ Понижилась активность в разработке новых визуальных средств параллельного программирования.
- ◆ Меньше новых разработок универсальных средств визуальной отладки правильности и эффективности.
- ◆ Появились средства отладки для распределенных сред и для гибридных вычислителей.



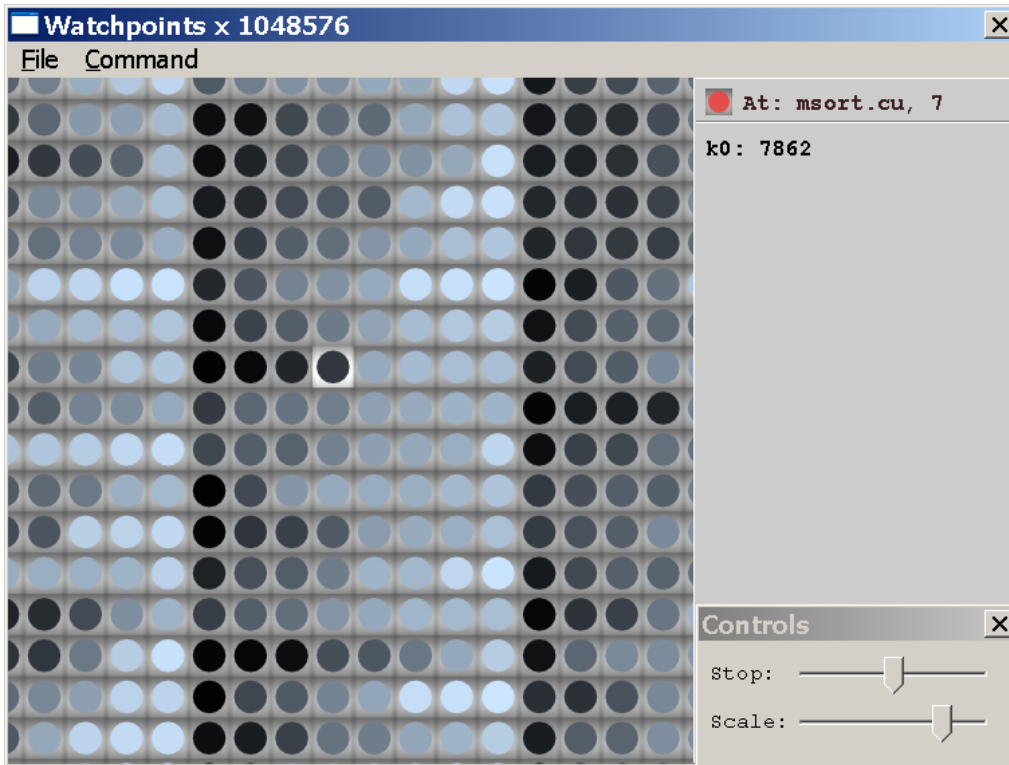
Logical graph for four processors in VisualGOP



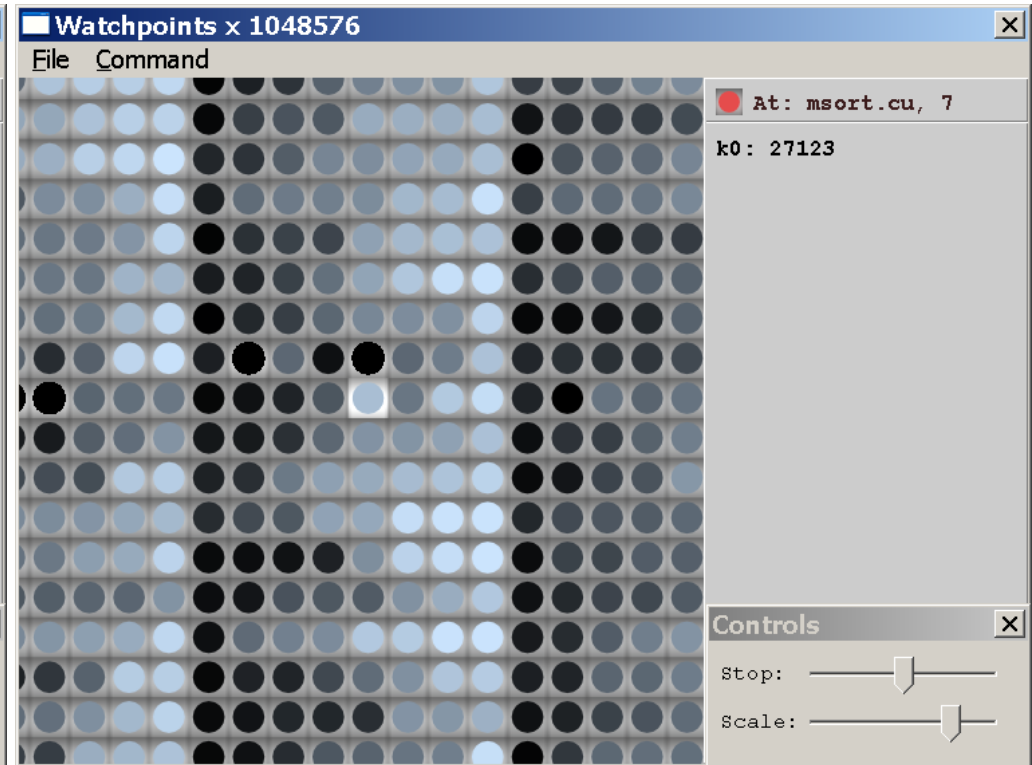
File View



Call relations within a program shown using linear edges (left) and using hierarchical edge bundles (right).



(a) Correct result



(b) Erroneous result

Методы решения

- ◆ семантический зуминг, позволяющий “сворачивать” и “разворачивать” визуальные блоки, отображающие отдельные части программы.
- ◆ “бесконечный экран”
- ◆ “полет” над визуальным пространством.
- ◆ методики визуализации на базе виртуальной и расширенной реальности
- ◆ инструменты поддержки создания параллельных программ

КОД ПРОГРАММЫ

```
or (j = 0; j < n; j++)
if (a[j][i] == 0)
{
k = j;
while ((a[k+1][j] == 0) && (k < n))
k++;
if (a[k+1][j] != 0)
for (i = 0; i < m; i++)
{
t = a[j][i];
a[j][i] = a[k+1][i];
a[k+1][i] = t;
}
}
```

БЛОК СХЕМА ПРОГРАММЫ

Вставить в блок Последовательно ▾

КОД ПРОГРАММЫ

```
or (j = 0; j < n; j++)
if (a[j][i] == 0)
{
k = j;
while ((a[k+1][j] == 0) && (k < n))
k++;
if (a[k+1][j] != 0)
for (i = 0; i < m; i++)
{
t = a[j][i];
a[j][i] = a[k+1][i];
a[k+1][i] = t;
}
}
```

БЛОК СХЕМА ПРОГРАММЫ

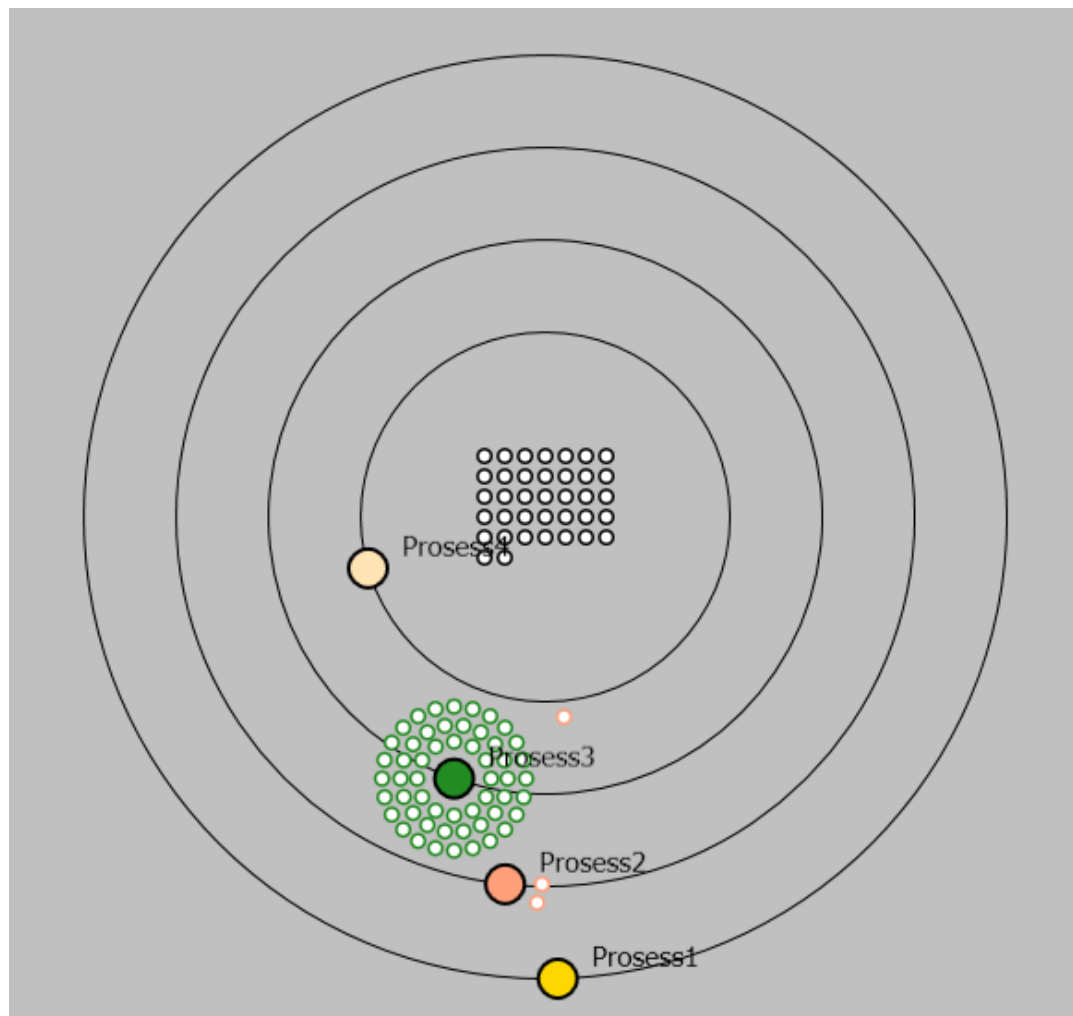
Вставить в блок Последовательно ▾

Инструменты поддержки процесса распараллеливания и создания параллельных программ из последовательных

Комплексное решение:

Разработка средств визуализации программного обеспечения в рамках системы программирования, как ее составная часть.

Первый прототип визуализатора **RideVis**



Первый прототип визуализатора **RideVis**

Визуализируются *хранилище, данные, процессы, количество процессоров в системе*. В качестве событий, на которых основывается визуализация, выступают *добавление/удаление* данных в хранилище, *запуск/завершение* процессов, *чтение/запись* данных в хранилище.

В визуализаторе предусмотрено несколько динамических видов отображения работы системы, которые могут использоваться для иллюстративной визуализации.

Планируется разработка инструмента для анализа корректности созданного программистом потока данных средств для статистического анализа эффективности работы программ для системы **RiDE** базового инструментария для визуального программирования.